EAI/Springer Innovations in Communication and Computing

Tanupriya Choudhury
Bhupesh Kumar Dewangan
Ravi Tomar · Bhupesh Kumar Singh
Teoh Teik Toe · Nguyen Gia Nhu *Editors*

Autonomic Computing in Cloud Resource Management in Industry 4.0

Springer

EAI/Springer Innovations in Communication and Computing

Series editor
Imrich Chlamtac, European Alliance for Innovation, Ghent, Belgium

Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but are not limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

Indexing: This series is indexed in Scopus, Ei Compendex, and zbMATH.

About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI reaches out to hundreds of thousands of individual subscribers on all continents and collaborates with an institutional member base including Fortune 500 companies, government organizations, and educational institutions, provide a free research and innovation platform.

Through its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and recognition of excellence by community.

More information about this series at http://www.springer.com/series/15427

Tanupriya Choudhury • Bhupesh Kumar Dewangan
Ravi Tomar • Bhupesh Kumar Singh
Teoh Teik Toe • Nguyen Gia Nhu
Editors

Autonomic Computing in Cloud Resource Management in Industry 4.0

Springer

EAI
RESEARCH MEETS INNOVATION

Editors
Tanupriya Choudhury
Department of Informatics
School of Computer Science
University of Petroleum and
Energy Studies (UPES)
Dehradun, Uttarakhand, India

Ravi Tomar
Department of Informatics
School of Computer Science
University of Petroleum and
Energy Studies (UPES)
Dehradun, Uttarakhand, India

Teoh Teik Toe
Nanyang Technological University (NTU)
Singapore, Singapore

Bhupesh Kumar Dewangan
Department of Informatics
School of Computer Science
University of Petroleum and
Energy Studies (UPES)
Dehradun, Uttarakhand, India

Bhupesh Kumar Singh
Computing and Software Engineering
Arba Minch University
Arba Minch, Ethiopia

Nguyen Gia Nhu
Duy Tan University
Da Nang, Vietnam

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-030-71755-1 ISBN 978-3-030-71756-8 (eBook)
https://doi.org/10.1007/978-3-030-71756-8

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Dr. Tanupriya Choudhury would like to dedicate this book to all Corona Warriors and Indian ARMY, for their dedication, sacrifice, and excellence towards our motherland INDIA, and also he would love to dedicate this book to his parents and in-laws: Sri Mrigendra Choudhury, Smt. Minakshi Choudhury, Sri Hemabrata Bhowmick, Smt. Debjani Bhowmick and his beloved wife Rituparna Choudhury and beloved Son Rajrup Choudhury (Ritam) for their immense support and love throughout this work. And also he would like to dedicate this book to his research guides Prof. (Dr.) Vivek Kumar, Prof. (Dr.) V Cyril Raj, Prof. Sumathy Eswaran, and Dr. Darshika Nigam, who have always mentored him during his Master's and Doctoral research. He would like to thank his uncle Late Girindra Mohan Choudhury for his all-time love, blessings, and support and wants to dedicate the book to his Uncle. He would also like to thank his uncle(s) Dr. Tapobrata Chowdhury (MBBS), Mr. Bhaskar Das, and Mr. Hitabrata Chowdhury and brothers Mr. Supriya Choudhury, and Mr. Debopriya Choudhury,

who supported whole-heartedly to complete this work. He would like to thank all colleagues of UPES for their all-time love, blessings, and support and also wants to dedicate the book to the UPES research fraternity.

Dr. Bhupesh Kumar Dewangan would like to dedicate this book to his father Shri Santram Dewangan, a strong person who always encouraged and supported to believe in himself; his mother Smt. Janki Devi, for being his first teacher; his beautiful wife Sanjana Dewangan, a lady who always behind him whenever he needs; and his children Shaurya and Sanvi Dewangan, who are the motivation of life. And also he would like to dedicate this book to his research guides Dr. Amit Aggarwal, Dr. Ashutosh Pasricha, and Dr. Tanupriya Choudhury, who have always mentored him. He would like to thank all colleagues of UPES for their all-time love, blessings, and support and also wants to dedicate the book to the UPES research fraternity.

Dr. Ravi Tomar would like to dedicate this book to all Corona Warriors, for their dedication, sacrifice, and excellence towards our motherland INDIA, and also he would love to dedicate this book to his father Sri Raj Kumar Tomar, and mother Smt. Sudesh Tomar, his beautiful wife Manu Tomar, and beloved daughters Kritika and Vedika for their immense support and love throughout this work. And also he would like to dedicate this book to his research guides Prof. (Dr.) Manish Prateek, and Dr. Hanumat Sastry G, who have always mentored him. He would like to thank all colleagues of UPES for their

all-time love, blessings, and support and also wants to dedicate the book to the UPES research fraternity.

Dr. Bhupesh Kumar Singh, is extremely grateful to his wife Kamana and his daughter Mahika for their love, prayers, caring, and understanding his research work. Also, he wants to express his thanks to his parents and family for their support and valuable prayers. His wishes to convey his special thanks to his mentor Prof. D.P. Sharma, Arba Minch University, for the keen interest shown to complete this book successfully. He would like to dedicate this book to his parents for all their love and support.

Dr. Teoh Teik Toe would like to dedicate this book to his family.

Dr. Nguyen Gia Nhu would like to dedicate this book to his wife Vu Thi Van Nhung and children, Nguyen Vu Minh Thu, and Nguyen Gia Bao

Joint Foreword

Autonomic computing could be defined as the methodology using which computing systems can manage themselves. The main objective of autonomic computing is to develop systems that can self-manage the management complexities arising out of rapidly growing technologies. Autonomic systems consist of autonomic elements that automatically employ policies. Many researchers have focused their research on this topic with their aim being improved application performance, improved platform efficiency, and optimizing resource allocation. The performance of any system depends on the effective management of resources. This is particularly significant in cloud computing systems which involve management of large number of virtual machines and physical machines. The editors have done a wonderful job in collating a variety of chapters from autonomic computing in the perspective of Industry 4.0. The editors are successful in presenting a comprehensive view of CRM and its integration with Industry 4.0 concepts. The book facilitates its reader in having a valuable understanding of various application areas pertaining to cloud and autonomic computing. Moreover, different issues and challenges in cloud resource management (CRM) techniques with proper propped solution for IT organizations has shown here. The recent research in CRM is the evidence of better performance through autonomic computing. We truly believe that the book will fit as a good read for those looking forward to exploring areas of autonomic computing.

HoD and Assistant Professor,
Department of CSE, Faculty of Engineering,
Comilla University, Comilla, Bangladesh — Partha Chakraborty

Professor, Department of CSE, Arba Minch University,
Arba Minch, Ethiopia — Bhupesh Kumar Singh

Preface

Our next generation of an industry—Industry 4.0—holds the promise of increased flexibility in manufacturing, along with automation, better quality, and improved productivity. It thus enables companies to cope with the challenges of producing increasingly individualized products with a short lead time to market and higher quality. Intelligent cloud services and resource sharing play an important role in Industry 4.0 anticipated Fourth Industrial Revolution.

Autonomic Computing is an operating environment that is on demand and responds to problems, threats, and failures automatically. It provides a computing environment that can manage itself and dynamically adapt to change. In a self-managing computing environment, system components such as hardware (desktop computers, storage units, and servers) and software (business applications, middleware, and operating system) include embedded control loop functionality.

The book will serve the different issues and challenges in cloud resource management (CRM) techniques with proper propped solution for IT organizations. The recent research in CRM is the evidence of better performance through autonomic computing. The book has 21 chapters based on the characteristics of autonomic computing with its applicability in CRM. Each chapter presents the techniques and analysis of each mechanism to make better resource management in the cloud.

Chapter 1 presents an Introduction to Cloud Resource Management. The chapter also discusses a few prominent use cases for CRM. Chapter 2 talks through Emerging Paradigms and Practices in Cloud Resource Management. Chapters 3 and 4 deal with explaining the role of Autonomic Computing in Cloud and Models

and Applications. It discusses the benefits and challenges of Autonomic computing also. Chapters 5, 6, and 7 present a detailed illustration of Issues and Challenges in Autonomic Computing and Resource Management. Chapter 8 is about the Classification of Various Scheduling Approaches for Resource Management System in Cloud Computing. Chapter 9 talks through the role of Optimization in Autonomic Computing and Resource Management. Chapter 10 discusses the Framework for Autonomic Resource Management in Cloud Computing Environment. Chapter 11 illustrates various reviews on Autonomic Computing on Cloud Computing using Architecture Adoption Models. Chapter 12 discusses self-protection approach for cloud computing. Chapter 13 deals with the concept of elastic security for Autonomic Computing using Intelligent Algorithm. Chapter 14 explores the concept of the architecture of Autonomic Cloud Resource Management. Chapter 15 talks through Industry 4.0 through Cloud Resource Management. Chapter 16 presents a Walkthrough in Live Migration Strategies for Energy-Aware Resource Management in the Cloud. Chapter 17 discusses Virtual Machine Scaling in autonomic Cloud Resource Management. Chapter 18 explores autonomic Resource Management in a Cloud-Based Infrastructure Environment. Chapter 19 talks through the digital dimensions of Industry 4.0: Opportunities for Autonomic Computing. Chapter 20 explores the area of security concept in the form of a Study of Resource Management and Security-Based Techniques in Autonomic Cloud Computing.

We hope that our efforts are appreciated and the reader benefits from this book.

Dehradun, India — Tanupriya Choudhury

Dehradun, India — Bhupesh Kumar Dewangan

Dehradun, India — Ravi Tomar

Arba Minch, Ethiopia — Bhupesh Kumar Singh

Singapore, Singapore — Teoh Teik Toe

Da Nang, Vietnam — Nguyen Gia Nhu

Acknowledgment

Dr. Tanupriya Choudhury, Dr. Bhupesh Kumar Dewangan, and Dr. Ravi Tomar would like to thank their workplace, University of Petroleum and Energy Studies, Dehradun, India, for giving a positive research environment to start this proposal. They would like to thank all contributors from ten different countries and specially reviewers throughout the globe who helped to review the chapters to maintain the quality of the book and for their valuable suggestions whenever required. They are also thankful to the senior leadership of the University of Petroleum and Energy Studies (UPES) and administration for giving the opportunity to hold ACCRMI 2020 and providing with all possible support. They are thankful to Shri Sharad Mehra, CEO, GUS-Asia, for his "all-time-go-ahead" blessings and freedom of work and Shri Dr. S. J. Chopra, Chancellor UPES, for his blessings and guidance as always. Honorable Vice-Chancellor Dr. Sunil Rai has been a continuous support as a torchbearer to us, big thanks to you Sir for your mentorship. Not to mention the instrumental personality, Prof.(Dr.) Priyadarshan Patra, Dean School of Computer Science, UPES, and Prof. (Dr.) T. P. Singh, HoD-Informatics, UPES, have been a rock solid support behind everything, thank you so very much Sirs. They would also thank their colleagues and friends for all time support, specially Mr. Partha Chakraborty from HoD CSE, Comilla University Bangladesh, Dr. Durga Prasad Sharma and Dr. Bhupesh Kumar Singh from AMU Ethiopia, and Dr. Praveen Kumar from Amity University Tashkent for moral and technical support as always whenever required. They would like to thank everyone enough for their involvement and their willingness to take on the completion of tasks beyond their comfort zones. See you all in the next edition of the book.

Dr. Bhupesh Kumar Singh would like to thank God, the Almighty, for His showers of blessings throughout his research work to complete the book successfully. He is highly grateful to all the authors from all parts of the world for contributing their research work for this book. He would like to say thanks to his friends and research colleagues Prof. D. P. Sharma, Dr. Tanupriya Choudhury, and Mr. Amin Tuni for their constant encouragement. He is extending his thanks to the University for their support during his research work. He would also like to thank all the staff of Research section of Arba Minch University for their kindness. He would like

to thank the management of Arba Minch University for their support to do this work. Finally, he would like to acknowledge all those who contributed directly or indirectly to complete this book.

Dr. Teoh Teik Toe would like to thank his family for all time support.

Dr. Nguyen Gia Nhu wishes to thank various people for their contribution to this project. He would like to deeply thank his wife, Vu Thi Van Nhung.

Contents

Introduction to Cloud Resource Management

G. Sobers Smiles David, K. Ramkumar, P. Shanmugavadivu, and P. S. Eliahim Jeevaraj

1 Introduction

Data have left the building since the emergence of cloud computing. Cloud computing [3, 4] ensures the capacity to facilitate the quality computing services. With the growing of the number of users, cloud computing [5, 6] addresses the needs of the different types of domains and its requirements. Cloud consists of advancement of new business and computing technologies such as Virtualization, Grid Computing, Web Services, Cloud Computing [7, 8], and Utility Computing. Professor Leonard Kleinrock is a renowned professor of computer science at University of California, Los Angeles, USA. As a graduate student in MIT during 1960–1962, he modeled the packet networks as a mathematical model and invented the Internet technology. The development of Internet occurred in his UCLA lab (3420 Boelter Hall). His workstation became the first link of the Internet in September 1969. This host computer sent the first message to be transmitted in Internet on October 29, 1969. An idea for providing "computing as a service" was first proposed by Leonard Kleinrock. He was the chief scientist heading the Advanced Research Project Agency (ARPA) project. Kleinrock anticipated that in future computer networks will emerge as an "Utility" [1].

G. Sobers Smiles David (✉) · P. S. Eliahim Jeevaraj
Bishop Heber College, Tiruchirappalli, India
e-mail: eliahimps.cs@bhc.edu.in

K. Ramkumar
Villa College, Male, Republic of Maldives
e-mail: ramkumar.krish@villacollege.edu.mv

P. Shanmugavadivu
Gandhigram Rural Institute – Deemed to be University, Gandhigram, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_1

Since1969, Information and Communication Technology (ICT) has evolved extensively, and this vision has become a reality. The advancements in networked computing scenarios have made computing into a service model. These services could be commoditized and provided like other conventional utility delivery such as water, electricity, gas, and telephony. Time has moved on, and technology has imbibed the ideas. There are a few notable success stories to be mentioned. In 1999, Salesforce.com began providing applications to customers through a basic website. The software applications were provided to organizations through the Internet and the dream of computing as a utility started its journey [2]. Despite the success of this method, time has to pass for global acceptance.

Web services were started to be provided in 2002 by Amazon and were christened as Amazon web Services. The services included data storage, task computation, and artificial intelligence. But, not until 2006, when Elastic Compute Cloud was introduced, it became a really successful commercial offering. In 2009, Google introduced browser-oriented cloud apps called Google Apps. The year 2009 also saw other tech giants like Microsoft introducing Windows Azure and others like Oracle and HP vying for the market share with their own products [3]. And today, cloud computing [4, 5] has become mainstream. The history of the cloud is shown in Fig. 1.

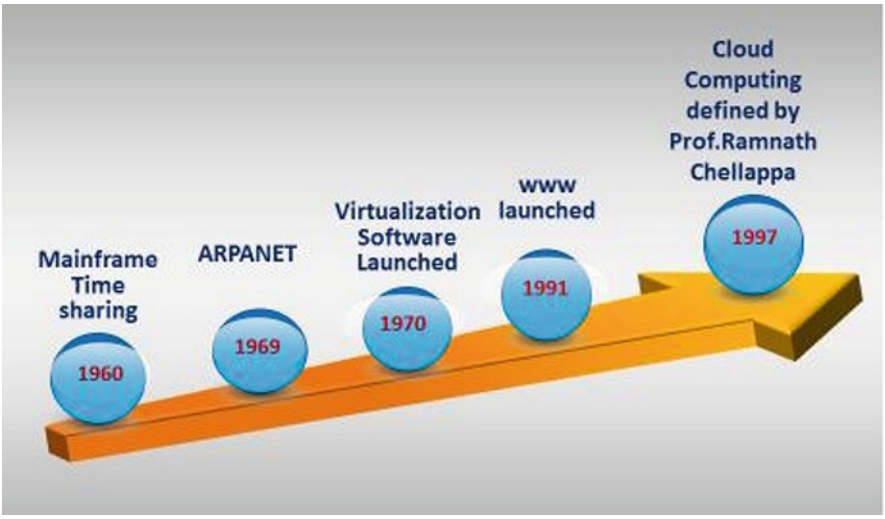

Fig. 1 History of the cloud

2 NIST's Cloud Model

The National Institute of Standards and Technology (NIST) is the standard organization of the U.S government. It defines cloud computing [6] as a computing prototype that allows pervasive, opportune, and need-based network access to a distributed collection of customizable resources such as networks, servers, storage devices, applications, and services. These resources can be rapidly acquired and freed with minimal work from administration or interaction with the service provider [7]. The NIST's Cloud model is given in Fig. 2.

NIST's cloud model consists of 5 important features. They are (1) on-demand self-service, (2) broad network access, (3) resource pooling, (4) rapid elasticity (5), and measured service [8].

The important features are elaborated follows.

- In need-based self-management of service, options are available to the user to acquire cloud computing resources on a self-service web portal.
- In broad network access, cloud computing resources can be accessed by heterogeneous devices.
- In resource pooling, the physical resources can be shared by multiple customers. This is done by segmenting the resources on a logical level.
- In rapid elasticity, on-demand resource provisioning and access is available to the user depending on the varying demands.
- In measured service, customers are billed on a pay method based on the amount of service utilized.

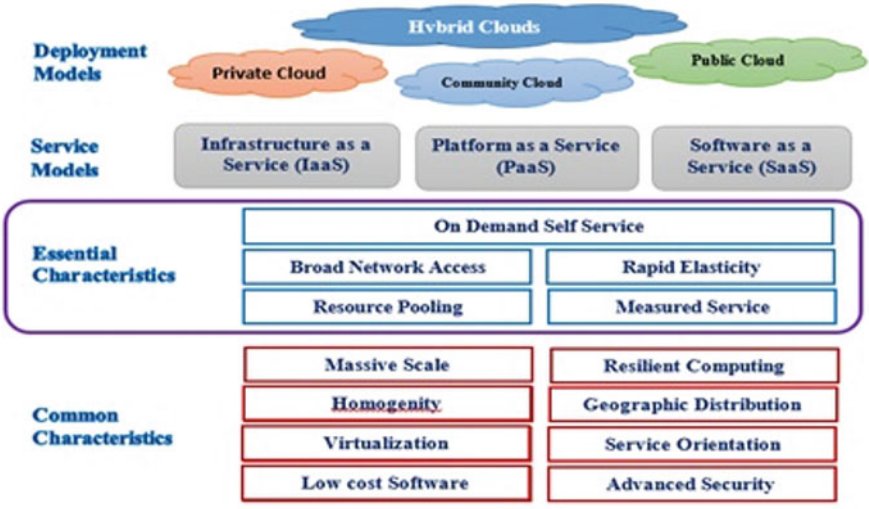

Fig. 2 NIST's cloud model

3 Benefits of the Cloud Model

Cloud computing presents enormous advantages to consumers in the form of charge or fee, faster service, and productivity. For starting a project, users need not have to invest money on resources. The need-based access to a distributed collection of resources in a self-service is offered by cloud. This service could be scaled dynamically and charged on a metered basis. There are compelling advantages using cloud computing in charges, faster delivery of services, and productivity [9]. Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. Cloud computing is comparable to grid computing. Grid is a type of computing where idle processing cycles of all computers in a network are coupled to solve problems that are otherwise difficult to solve for an individual machine.

In cloud computing, the word cloud is a metaphor for "the Internet". The term cloud computing refers to "Internet-based computing." Computing services like execution environment and devices, repository, and programs are delivered to an organization through the Internet. Cloud computing provides very efficient and faster computations that benefit applications to deliver personalized information, Also, Cloud provides efficient data storage and high computing power to the online applications.

To achieve this, cloud computing uses networks of enormous collection of servers. These servers run on cheaper technology such as PC. This, by employing specific connections, allows to share data handling duties. A large pool of systems that are linked together is what makes this shared IT infrastructure [10]. Containerization and virtualization techniques help maximize the computing power of the cloud. At present, there are no fully defined standards (1) for connecting the systems and (2) the software needed to make cloud computing work. Because of this scenario, many organizations define their own cloud computing technologies.

Cloud computing systems such as IBM's "Blue Cloud" technologies are based on open-source standards and software and deliver Web 2.0 capabilities [11]. Precisely, cloud computing provides Cybernation, automatic capacity adjustment, more productive development routine, better-quality demonstrability, handling the traffic fluctuations, recovery from failure, and business stability. Cloud computing could be termed as a new architype for delivering vigorous establishment of futuristic data centers by combining services of networked cybernetic machines. The true potentiality of cloud computing could be realized if the leading cloud vendors deploy cloud centers across the corners of the world. This would effectively counter failure. Data centers are slowly becoming the support for companies to maintain productivity and efficiency in their business processes [11].

4 Cloud Computing Architecture

The IT Architecture has progressed over a period time. The progress got the momentum during 1960s and the 1970s. This period witnessed costly, very big, overworking, and inflexible servers, where resources were collected, and virtualization was used expansively [12]. During 1980s and 1990s, client server technology emerged as the alternative, and the cost of computing structure and networking went down. During the 2000s, data centers emerged and were implemented in large numbers. Thus, commodity grid computing had to give way for the return of virtualization [12]. Cloud computing has taken a step further by giving need-based self-service, metered usage, fully computerized need-based resource, and amount of work management [13]. Cloud computing architecture that is responsible for distribution of cloud services contains multiple cloud constituents interacting with each other over a loosely tied mechanism such as a communique queue (Fig. 3).

Flexible allocation indicates astuteness in the use of firmly or loosely joined as applied to mechanisms such as these and others [14]. Janel Ryan et al. proposed that "In cloud computing, it is imperative to find the appropriate architecture for an application. It is the duty of organizations to recognize their application requests and their analogous cloud architecture. This would ensure the consistency and performance requests of their applications." In addition, an ideal architecture's characteristics include the following:

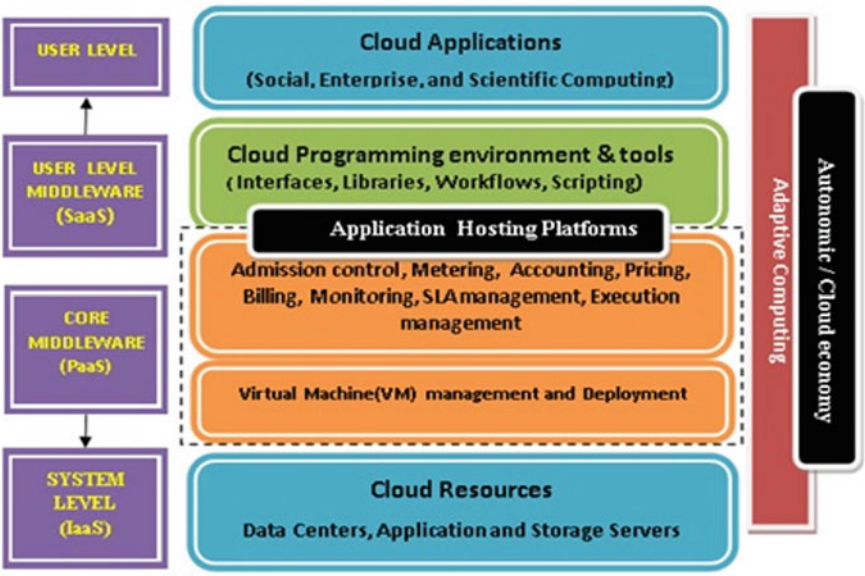

Fig. 3 Layered cloud computing architecture

- Must promote many standards within the same infrastructure with regard to resource management.
- Should aid multiple standards in the same cloud architecture and allow the customer to migrate to a newer one if they wish, with the possibility of retaining everything in the CSP network.
- Should promote use cases, multiple deployment models, and service categories.
- Must provide early detection, diagnosis, and fixing of service-related problems.
- Should help with the management of resources allotted to a user and provide reports on SLA compliance.
- Should make only the services as visible and abstract the resource allotment details.
- Must ensure security up to the Intranet level on the network.
- Should allow mobility of virtual machines within a data center.
- Must allow the users to access their resources by their actual names irrespective of the migration of resources.
- Must provide automated deployment of cloud—services to support scalable resource provisioning and configuration.

5 Cloud Resource Management

In cloud computing, resource administration includes provisioning the resource, allocating the resource, and monitoring the resource. Cloud resources include the web servers, memory, storage, network, CPU, application servers, and cybernetic machines called the virtual machines. Cybernetic machines are the processing units in cloud. The productivity of any system depends on the successful administration of resources. This is particularly significant in cloud computing systems that involve administration of enormous number of virtual and physical machines. In particular, the performance is innately reliant on the efficient management of resources [15]. Significantly, degradation in performance is caused by resource dispute by numerous applications. The varied nature of hardware resources in the cloud makes it even more challenging. Any resource administration system in cloud computing must cater to the three types of services, such as IaaS, PaaS, and SaaS. For IaaS, an adaptive resource management system (RMS) is needed to cater to the ever-changing virtual resource requests and resource constraints [16]. In short, resource management in cloud is aimed at discovering available resources, selecting the appropriate resources, and reserving the network and end system resources. This reservation mechanism ensures guaranteed service to the user's requirement. The cloud resource management phases are given in Fig. 4.

Also, the RMS should be adept at controlling complex numerous resources to performance association without any supposition of any system prototype. The RMS system must possess the ability to highly scale for supporting larger applications. For PaaS, the RMS should offer enhanced performance of the platforms on offer. It should also offer moderation for virtual resource dispute and job intervention.

Fig. 4 The cloud resource management phases

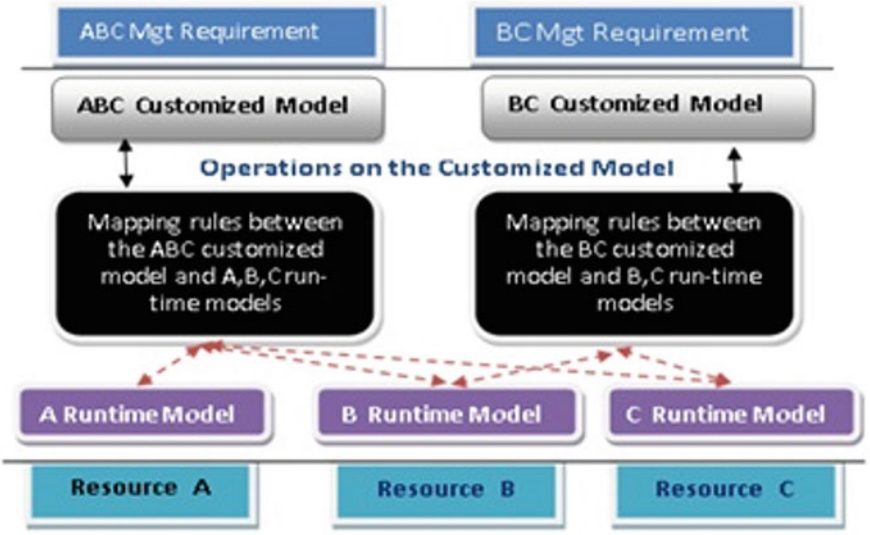

Fig. 5 The architectural model for managing cloud resources

For SaaS, the RMS must provide elasticity for varying application requests for resources and be highly flexible and offer operations for online configuration [17]. Resource allocation is the method of obtaining resources and then administering these resources by assigning them to the needed applications. The architectural model for managing resources in cloud is given in Fig. 5.

The huge bottleneck in administering cloud resources originates from the dissimilar nature of cloud applications and their resources. One of the key bottlenecks in administering resources is the administrator's familiarity with the interfaces (APIs) and their proficiency in writing programs on them [18]. In this model shown in Fig. 6, the execution time model of the resource is built, and then a synchronism model is built to avoid uncertainty. Representing rules are needed to insist on administration requests and to guarantee makeover of models [19].

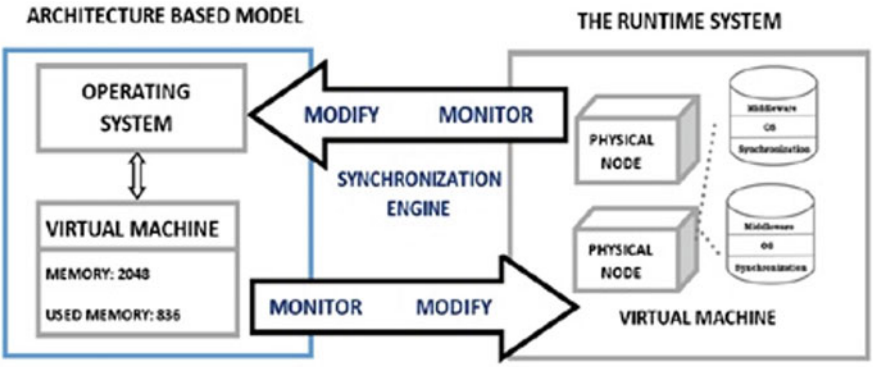

Fig. 6 The synchronization of models

6 Industry 4.0

Industry 4.0 is a strategic initiative by the German Government to transform industrial manufacturing through digitalization and latest technologies. Industry 4.0 symbolizes the fourth industrial revolution in the manufacturing field. The first revolution involved mechanization through water and vapor power. The second industrial revolution involved using current for large-scale manufacture and assembly lines. The third revolution involved computers and automations systems. The fourth industrial revolution involved enhancing the computers and automations systems with smart and autonomous systems managed by data and machine learning.

In Industry 3.0, while computers were introduced, it was the disruptive technology. It changed the entire scenario in industries involving mass production. With the emergence of Industry 4.0, the systems have become more connected and have the ability to communicate with one another. Also, the systems now have the capacity to make decisions without any human intervention. As Industry 4.0 unfolds, a rare combination of Internet of systems, mechanized systems, and Internet of things (IoT) have made the concept of Smart Factory a reality. Industry 4.0 is making the smart systems more smarter, and the production has become more efficient resulting in less wastage. The true power of Industry 4.0 lies in the network of smart machine digitally connected that are capable of acquiring new information and sharing them [20] (Fig. 7).

6.1 Industry 4.0 Applications

- *Providing insights to the manufacturer*: Smart machines collect large volumes of data through the sensors for maintenance and performance issues and analyze the data to identify any potential patterns and to acquire insights. This is not

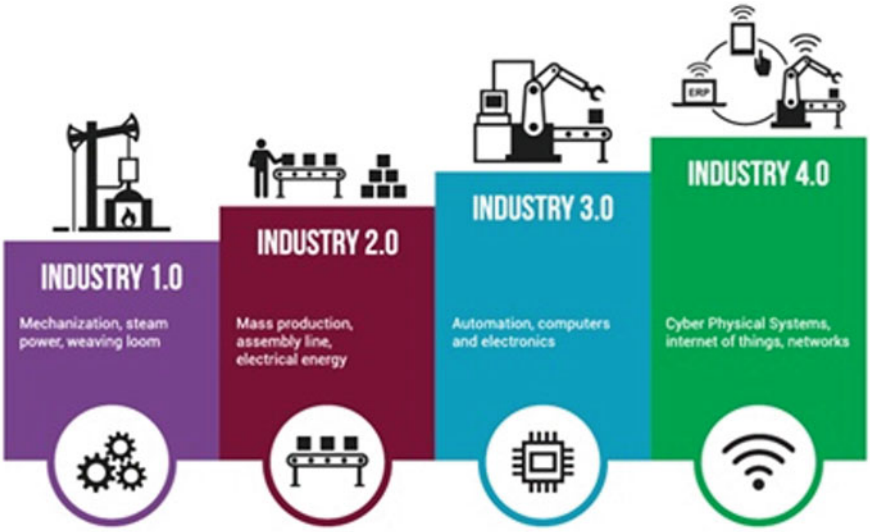

Fig. 7 Evolution of industry 4.0

possible with human workforce. Smart machines can draw the attention of the manufacturer on the issues that need immediate attention and help to optimize the operations. In the recent times, an African Gold mine employed smart machines, and the sensors identified the low level of oxygen during the leaching process. This insight helped the gold mining company to fix the problem and increase the yield by 3.7% which in turn resulted in cost saving of 20 million $ annually.

- *Connected supply chain*: Smart machines can help a connected supply chain in case of a weather delay. In such a case, smart machines can proactively adjust the manufacturing priority based on the weather scenario.
- *Autonomous vehicles*: Smart machines can manage autonomous vehicles involved in shipping yards by leveraging trucks and autonomous cranes and thereby can streamline shipping operations.
- *Robotics*: Robotics is emerging to be affordable for medium- and small-scale industries. Robots are employed in packing, loading, and readying to ship by the manufacturers. Robots bring down the cost and allow optimized use of floor space for retailers. Amazon employs robots in its warehouses and enjoys significant cost reduction in operations.
- *3D printing*: 3D printing technology has evolved from prototyping to actual production. And 3D printing technology is still evolving. With the possibility of using metal additive manufacturing, 3D printing has opened a floodgate of opportunities for production.
- *Internet of things (IoT)*: The emergence of Internet of Things has brought in a sea of changes in the way internal operations are done. Equipment and operations could be optimized through cloud environment. This also opens up the possibility

of sharing the insights with organizations that use the same equipment and gives smaller companies a chance to access the technology insight [21].

7 Significance of Cloud Computing in Industry 4.0

The role played by cloud computing for continuing the development of the Fourth Industrial Revolution, Industry 4.0, is very crucial. As with its characteristic, Cloud technology assists to collect resources and integrate information for businesses. It also offers an opportunity for open-source group effort that helps to expedite and refine research. The first phase of the fourth industrial revolution is happening in Automotive and manufacturing industries. Distributed cloud is one of the key technologies to harness the growth in Industry 4.0. Cloud computing has allowed businesses to voluntarily change with the evolving times without dropping data, made possible with the integration of artificial intelligence and automation into industry, cloud computing provides unprecedented network, storage, and computing abilities. Compute services enable the platforms capable of merging Internet of Things, robotics, and automation, which contribute to innovation [22].

The IT enterprise is about to be reshaped due to a seismic shift. Due to the introduction of key trends like Artificial Intelligence and Internet of Things, applications and data are more and more being extended across multiple data centers—some on onsite and some in multiple clouds—and edge sites hitherto encapsulated in their supporting infrastructure. A recently held Gartner study reveals that, by 2022, half of the present enterprise-generated data will move to a single centralized cloud. The study has also forecasted that by 2025 this number will rise to 75–90% [23]. This scenario will lead to two distinct and separate phenomena: (1) the clouding of the edge and (2) the rise in proper multicloud [22]. And these developments point to the rise in the distributed cloud as shown in Fig. 8.

Distributed cloud is an application of Cloud Computing used to interconnect data and applications that can be provided from various geographical locations. In distributed computing, the data and applications are shared among multiple systems located in different geographical locations. This enables cloud to speed up communication for global services and more responsive communication for particular region [23].

7.1 Types of Distributed Cloud

There are two types. (1) Public resource computing (2) volunteer cloud

- Public-resource computing:
  This is a subclass of traditional cloud computing and more related to distributed computing. This is also referred as Global Computing and peer-to-peer

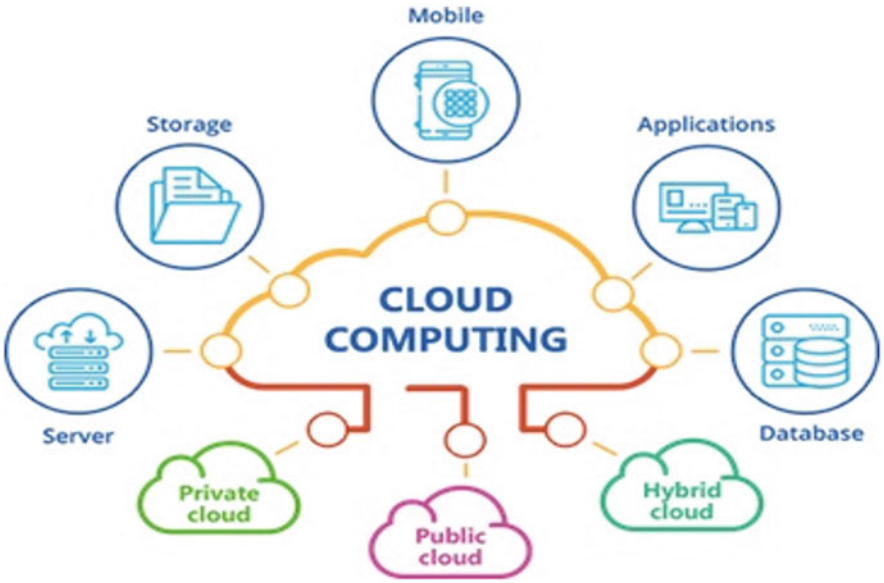

Fig. 8 Distributed cloud computing scenario

computing. This type of computing is employed in scientific supercomputing. The number of systems with Internet connectivity is growing almost every second. The combined computing power of these systems is enormous, almost equivalent to many data centers [24].

- Volunteer cloud:

  This is a combination of cloud computing and public-resource computing. Here, the cloud computing structure is constructed with willing resources. This type of cloud is also referred as Ad hoc clouds.

7.2 Benefits of Distributed Computing

- Helps to reduce costs by offering repetition, dependability, and geo-replication.
- Easy to respond in case of failures by employing remote replicas that can instantly reset.
- Reduced wide-area traffic by using the distributed cloud resources.
- Allows parallel processing of tasks by breaking a complex problem and data into segments and employing multiple systems to work on it.

7.3 Challenges With Distributed Computing

- Very high deployment cost involved with processing overhead due to additional computation and exchange.
- Requires more maintenance and security [25, 26] as users also have to deal with replicated data provided from multiple locations.
- Difficult to maintain, deploy, and troubleshoot hardware and software due to the volume of data and size of the applications associated.

8 Use Cases for Cloud Computing in Industry 4.0

The cloud [1] is a fantastic, federated tool that is altering the approach how data are stored and managed. Companies are now more open to amalgamating and partaking information rather than secreting it from competitors. Cloud [2] has enabled to open up channels of information exchange which will be of value to the whole industry. This also has resulted in faster and more refined results.

The mobile networks, 4G which is in use today, and the emerging 5G technology, are designed to support higher frequency range and lower response time communication. This is available on the broadcasting interface for both downlink (DL) and uplink (UL) data. This makes it an ideal scenario for Industry 4.0. Distributed cloud [3] makes use of these features, enabling a widespread runtime environment for applications. This ensures communication with short latency, high reliability, performance, and data locality [27].

The complexity of the infrastructure is hidden by the distributed cloud. Simultaneously, the elasticity of cloud computing [4] is also maintained. Also, the application components are kept in an optimal location. This enables to avail key characteristics of distributed cloud. Many manufacturing industries and the automotive sector already have use cases, and they are most likely the first ones to adopt distributed cloud technology.

8.1 The Next-Generation Services for Automotive Sector

The next-generation services for automotive sector aims to transform driving, ease the movement of vehicles, manage energy usage competently, and emit low. All this could be possible only when the automobiles support mobile communication in their vehicles. With the addition of mobile communications, automotive service could include intelligent and automated driving, use of sophisticated maps that includes real-time data, and innovative driving support using cloud [4–6]. The driving assistance using cloud could make use of cloud-based analytics of uplink video streams. These processes involve large chunks of data to be stored in cloud

and transmitted between vehicles when the vehicle is active and on the move with actual time features within a stipulated time.

The real-time use case for safety in automotive industry is vehicle to everything/co-operative Intelligent Transport System (V2X/C-ITS), where short latency is one of the main requirements. The new class of cloud-based services by automotive industry is possible only when large volumes of real-time data with real-time characteristics are able to be transmitted between mobile vehicles. This puts a huge demand on the network capacity. Also, actual time data require to be transmitted not beyond a stipulated time frame of 30 min/day. These data with changing geographical collection of vehicles employ different rules and network technologies. By the year 2025, the global estimation for the number of connected vehicles is 700 million. And the data volume estimation is 100 petabytes/month. But, at Ericcson, the estimation is comparatively low at 10 Pb/month. At the same time, Gartner expects the volume to be 1 TB/month /vehicle [28]. The Automotive Edge Computing Consortium (AECC) white paper states that the effective handling of large volumes of data is not supported by the present computing and network architectures. The white paper has also suggested remedial measures. Of these, three recommendations stand out: (1) filtering the data while it is allowed in the cloud; (2) employing topology-aware computing and storage services otherwise called as Global Automotive Distributed Edge Cloud; and (3) upgrading the mobile network capacity, availability, and coverage to cater to the growing demand.

8.2 Localized Network With Distributed Computing

In order to streamline traffic and data-processing problems in the existing ubiquitous systems, Ericcson developed the concept of localized network with distributed computing services. Here, the connectivity of vehicles is taken care by the respective localized networks with network coverage. All the processing of data associated with a vehicle is done locally, thereby reducing the amount of data traffic. This enables faster communication for the vehicles.

There are three key components in this concept: (1) a localized network; (2) edge computing; (3) and data exposure. A localized network is a restricted network that consists of minimal connected vehicles in a particular region. In this network, the level of movement between the means of transportation and the clouds is significantly reduced. Edge computing represents distributed computational resources in the range of a localized network. This results in less computation and less processing time for data exchange to connected vehicles. Data exposure combines the restricted network and the distributed processing. It also secures integration of the locally produced data. Data can be rapidly processed by contracting related data down to a particular area, thereby enabling the network to integrate the information. Further, the connected vehicles could be notified in real time. Extreme care is needed to keep the size of the data as minimum during transmission of data [29].

8.3 5G Technology and Augmented Reality

Recent research indicates that 5G technology will empower automotive industry with wide deployment of interactive media applications. Also, Machine learning and augmented reality will be the two main technologies that will be extensively used in the digitization of industries. It is expected that the future workers will use more of eye-tracking smart glasses and tactual gloves than any physical equipment. In such a case, human-to-machine interaction will require heavy compute resources, high network bandwidth, and low latency network. The limitations involved with light waves used in optic fibers rule out the possibility of running the complete application in large central databases [30].

The components of the application can be executed in three possible ways. (1) On the device itself, (2) in the edge server, and (3) in the central cloud. Here, the device could be offloaded with minimal latency by setting up application constituents at the network edge. In order to determine the 3D position of objects, more synchronization is needed for numerous actual time camera feeds. This mechanism is optimized by edge compute. Further, more services are also provided on the edge site in the form of advanced cloud software as a service.

8.4 Distributed Cloud Solution by Ericcson

Developers and researchers at Ericcson have designed a disseminated cloud solution that could provide all the necessary competencies supporting the usage scenarios of Industry 4.0, including localized networks and private networks. This distributed cloud solution satisfies all the security requirements needed for a smart factory. Further, this cloud solution provides edge computing and makes sure all the end-to-end network requirements are met. In addition, this cloud solution also provides orchestration, management, and exposure for the cloud and network resources together [31].

The distributed cloud is defined as a distributed execution environment spread across multiple sites situated at different geographical locations and managed as a single entity.

- The cloud infrastructure resources are abstracted, and the complexity involved in allocating the resources is hidden to the application and the user.
- The cloud solution is based on the technologies, such as 3GPP edge computing technology, network functions virtualization, and software-defined networking.
- The use of the abovementioned three technologies enables open access for applications.
- This solution also supports automated deployment in heterogeneous clouds.

9 Conclusion

Global standards and Common Architecture for Industry 4.0 are still in its nascent stages. An ideal solution would be coming together of industries and vendors creating an ecosystem and formulating the requirements, use cases, working methods, common reference implementation mechanisms, and standards to be followed. The world is looking forward to ecosystems such as Automotive Edge Computing Consortium (AECC), The 5G Alliance Connected Industries and Automation (5G-ACIA), Industrial Internet of Things (IIoT), and Industry 4.0 to join and formulate the mechanisms. Industry 4.0 has truly become the game changer. With the advancements in artificial intelligence and machine learning, the scenario can only get better and better. The emergence of 5G technology has opened the doors for automated transport and vehicles. Days are not far when everything around the human beings will move toward automation.

References

1. Alex, Y. A. G. (2017). Comparison of resource optimization algorithms in cloud computing. *International Journal of Pure and Applied Mathematics*, 847–854.
2. An architectural blueprint for autonomic computing. (2005). *Autonomic computing white paper*. Third Edition: IBM Press.
3. Vashistha, A., Kumar, S., Verma, P., Porwal, R. (2018). A self-adaptive view on resource management in cloud data center. IEEE Computer Society.
4. Singh, B. K., Alemu, D. P. S. M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
5. Tomar, R., Khanna, A., Bansal, A., Fore, V. (2018). An architectural view towards autonomic cloud computing. Data engineering and intelligent computing (pp. 573–582).
6. Yadav, A. K., Tomar, R., Kumar, D., Gupta, H. (2012). Security and privacy concerns in cloud computing. *Computer Science and Software Engineering*.
7. Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing, 60*(2), 268–280.
8. Biography of Leonard Kleinrock. *IEEE Computer Society* (2019).
9. Sheshasaayee, A., & Megala, R. (2017). A study on resource provisioning approaches in autonomic cloud computing. In *International conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC 2017)* (pp. 141–147). IEEE Publications.
10. Mell, P., Grance, T. (2011). The NIST definition of cloud computing. *Computer security*. NIST Special Publication 800–145.
11. Sukhpal Singh Gill, Peter Garraghan, Vlado Stankovski, Giuliano Casale, Ruppa K. Thulasiram , Soumya K. Ghosh, Ramamohanarao, K., Buyya, R. (2019). Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *The Journal of Systems and Software, Elsevier Publications*. 102–127.
12. Dehraj, P., & Sharma, A. (2021). A review on architecture and models for autonomic software systems. *The Journal of Supercomputing, Springer Nature, 77*, 388–417.
13. Pompili, D., Hajisami, A., & Tran, T. X. (2016). Elastic resource utilization framework for high capacity and energy efficiency in cloud RAN. *IEEE Communications Magazine, 54*(1), 26–32.

14. Felter, W., Ferreira, A., Rajamony, R., Rubio, J. (2015). An updated performance comparison of virtual machines and linux container. 2015 IEEE International Symposium on in performance analysis of systems and software, (ISPASS) (pp. 171–172).
15. Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems, 24*(7), 1366–1379.
16. Coutinho, E. F., Gomes, D. G., & Neuman de Souza, J. (2015). An autonomic computing-based architecture for cloud computing elasticity. In network operations and management symposium (LANOMS) (pp. 111–112).
17. Tesfatsion, S. K., Wadbro, E., & Tordsson, J. (2018). PerfGreen: Performance and energy aware resource provisioning for heterogeneous cloud. IEEE international conference on autonomic computing, *IEEE Computer Society*.
18. Vieira, K., Koch, F. L., Sobral, J. B. M., Westphall, C. B., de Souza Leão, J. L. (2019). Autonomic intrusion detection and response using big data. *IEEE Systems*. https://doi.org/10.1109/JSYST.2019.2945555.
19. Vuksanović, D., Ugarak, J., Korčok, D. (2016). Industry 4.0: The future concepts and new visions of factory of the future development. International scientific conference on ICT and E-business related research. *Advanced Engineering Systems* (pp. 293–298).
20. Rojko, A. (2017). Industry 4.0 concept: Background and overview. *International Journal of Interactive Mobile Technologies (iJIM)*.
21. Alcácer, V., & Cruz-Machado, V. (2019). Scanning the industry 4.0: A literature review on Technologies for Manufacturing Systems. *Engineering Science and Technology, 22*, 899–919.
22. Zhou, K., Liu, T., Zhou, L. (2015). Industry 4.0: Towards future industrial opportunities and challenges. 12th international conference on fuzzy systems and knowledge discovery (FSKD) 2015 (pp. 2147–2152).
23. Boberg, C., Svensson, M., Kovács, B. (2018). Distributed cloud – A key enabler of automotive and industry 4.0 use cases. *Charting the Future of Innovation, Ericcson Technology Review, No.11–2018.*
24. Khan, A., Turowski, K. (2016). A perspective on industry 4.0: From challenges to opportunities in production systems. In proceedings of the international conference on internet of things and big data (IoTBD 2016), scitepress (pp. 441–448).
25. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*(9), 583–585. https://doi.org/10.35940/ijitee.i7637.079920.
26. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*(5), 1144–1148. https://doi.org/10.35940/ijeat.E9941.069520.
27. Sharma, M., Kumar, R., & Jain, A. (2019). Implementation of various load-balancing approaches for cloud computing using CloudSim. *Journal of Computational and Theoretical Nanoscience, 16*(9), 3974–3980.
28. Jain, A., & Kumar, R. (2014). A taxonomy of cloud computing. *International Journal of Scientific and Research Publications, 4*(7), 1–5.
29. Compastié, M., Badonnel, R., Festor, O., He, R., & Kassi-Lahlou, M. (2016). A software-defined security strategy for supporting autonomic security enforcement in distributed cloud. IEEE 8th international conference on cloud computing technology and science, IEEE Computer Society.
30. John, W., Sargor, C., Szabo, R., Awan, A. J., Padala, C., Drake, E., Julien, M., Opsenica, M. (2020). The future of cloud computing - highly distributed with heterogeneous hardware. *Ericcson Technology Review*.
31. Eriksson, A. C., Forsman, M., Ronkainen, H., Willars, P., Östberg, C. (2020). 5G new radio RAN & transport – Choices that minimize CTO. *Ericcson Technology Review*.

Emerging Paradigms and Practices in Cloud Resource Management

Durga Prasad Sharma, Bhupesh Kumar Singh, Amin Tuni Gure, and Tanupriya Choudhury

1 Introduction

Originally cloud computing is a computational model that facilitates on-demand resources of computing systems and services, especially virtual machines, cloud storage, computing, and communication software, with the least involvement of user's inactive modes. These computing resources are provided on-demand as metered services like a public utility. In the primary phase, the main architecture of cloud computing was designed based on characteristics of utility computing, and later the cloud resources were pooled at distributed places in centralized modes and characterized as data canters [1].

To realize the Journey of cloud computing technology, we need to go back in history. In the 1950s, scientist Herb Grosch (a well-known author of Grosch's law) hypothesized that the entire world would operate on dumb terminals powered by about 15 large data centers, i.e., perceived as modern cloud data canters [2]. Later, the term "Cloud" was used as a symbolic representation or metaphor for the Internet and an abstraction of the underlying network infrastructure. Cloud computing [3, 4] was introduced by John McCarthy in 1960 with a concept of the illusion of an infinite supply of resources. The actual term "Cloud" was borrowed from telephony in that telecom companies, who until the 1990s offered primarily dedicated point-to-point data circuits, began offering Virtual Private Network (VPN) services with comparable quality of service but at a much lower cost [5].

D. P. Sharma · B. K. Singh (✉) · A. T. Gure
Computing and Software Engineering, Arba Minch University, Arba Minch, Ethiopia
e-mail: sharma.dp@amu.edu.et; dr.bhupeshkumarsingh@amu.edu.et; amin.tuni@amu.edu.et

T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_2

Next to the dot-com bubble, Amazon played a key role in the rapid development of cloud computing [6, 7] by modernizing their own data centers. It was an initial major step toward the computing paradigm and revolutionizing the cloud technology. As an innovator, Amazon opened the door for access to cloud computing resources to external consumers. In 2006, Amazon launched Amazon Web Service (AWS) on a utility computing platform for world consumers and became the pioneer of cloud computing in real sense.

In early 2008, Eucalyptus also entered into the cloud market and became the first open-source cloud service provider, AWS API-compatible platform for deploying and facilitating cloud computing resources privately [8]. Also, at the same time in early 2008, Open Nebula, was declared as the first and the foremost open-source software for deploying the private, hybrid, and other federated clouds [9]. In mid-2008, Gartner observed a scope for cloud computing (i.e., to shape the relationship among consumers of information technology services and information technology service providers). Later, this was revolutionized as "switching from company-owned hardware and software assets to per-use service-based models" [10].

Cloud computing is a general term for anything that involves the delivery of hosted services over the network, i.e., the Internet. Cloud computing can be viewed as access to resources from a set of pooled computing resources needed to perform functions with dynamically changing needs. Cloud can be perceived as a techno-business paradigm in which hosted resources are delivered over the Internet to perform certain tasks with dynamically changing needs of resources. In fact, the cloud is a convergence model for enabling convenient, on-demand access to a shared pool of computing resources such as computing servers, storage grids, networks, application software, computing tools, and services in a convenient and ubiquitous environment.

These infrastructure products and services are nothing but "resources" that can be provided as service with quick provision or reprovision at anytime, anywhere over any device, and released with nominal admin efforts or user intervention. The cloud models presented in Fig. 1 are composed of five essential characteristics (e.g., on-demand self-service, broad network access, resource pooling, location independence, rapid elasticity, and measured service), three service models (IaaS, PaaS, and SaaS), and four deployment models (Public, Private, Hybrid, and Community) [11–14].

The use of information technology (ITs) by multidimensional applications has been changing with respect to time dynamics. These dynamic changes create a new horizon of the vibrant echo system in computing, communication, and collaboration environment. The individual users of the big enterprises need on-demand computing, communication, and collaboration resources. This scenario reflects the fact that contemporary IT needs have been dynamically evolving, and motivation is shifting from owned infrastructure, i.e., capital expenditure to operational expenditure. This implies that the popularity of rent-based infrastructure is rapidly increasing than own infrastructure. The acceptance of cloud computing can be visualized by incremental growth and investment migration over the cloud rather than the purchase of new infrastructure. The enhanced business efficiency

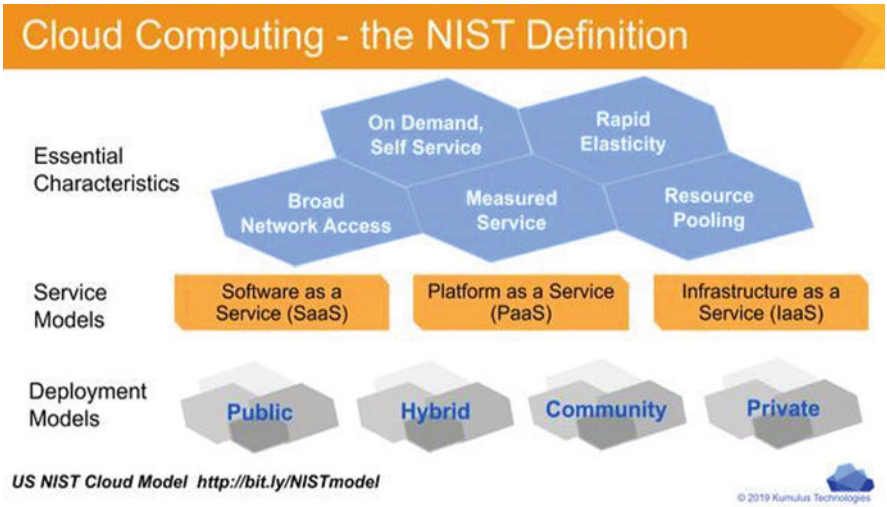

Fig. 1 The cloud models

through the growing usage of IT services offered by cloud computing will further boost the growth in migration of IT services toward cloud, especially in small- and medium-scale enterprises (SMEs) [15].

In general, cloud computing [16–18] enables users to migrate their data storage and computational needs to a remotely available infrastructure with minimal user intervention and impact on computing system performance. Usually, this offers a variety of benefits that could not be experienced when computing over traditional infrastructure like, for instance, on-demand high scalability, both vertical and horizontal; freedom of provision and reprovision of computing resources with a variety of options; and high uptime, i.e., 99.99%. All such variety of IT tools, products, and services could be accessed in an easy and user-centric manner over cloud. These cloud resources can be anything like hardware or software (network, storage, applications, developing tools, high-performance system, etc.).

In SMEs and Big Enterprises, workflows have emerged as a technique to formalize and structure data analytics, perform computations over distributed cloud resources, gather the output of the processed data, and then repeat the analysis if required for desired results. As a matter of fact, the SMEs cannot afford rapid changing high-end modern IT needs for exploring the full potential of structured or unstructured data analytics collected from the salient distributed sources to manage and alleviate the competitive needs [19].

Also, the scientific workflows in scientific collaborations enable the sharing of data analytics results, and therefore the scientific workflows have been viewed as an emerging paradigm, where engineers and data scientists can handle complex scientific processes easily and conveniently to share worldwide for rapid result disseminations in scientific discoveries and research [20].

This cloud computing, business, and scientific workflows convergence is enabled by resource management, which includes resource provisioning and reprovisioning, scheduling and rescheduling, and allocation and reallocation [21, 22].

In a Cloud computing [23, 24] environment, smart and portable systems such as Mobiles, Tablets, and Fablets and services are highly anticipated, as provisioning of inefficient resources may result in the failure of timeliness of task processing [25] [26]. To avoid such issues and challenges, provisioning the most feasible computing resource, most fit storage space, and suitable application can significantly reduce the unpredictable monetary losses. Such critical cost savings with no substantial impact on application performance can be considered as a good sign toward efficient management of cloud resources.

2 Literature Review

2.1 Cloud Resource Management

In the cloud resource management, the significant challenges are efficient allocation of resources to the workload based on the specifications, energy efficiency, uptime, on-demand horizontal and vertical scalability, consumer satisfaction, trust, transparency, and QoS. Resources are hardware or software entities used for computing and communications [27]. Resource management is the process or method of allocating appropriate computing, communication, storage, and other resources to run the applications as per the needs of cloud consumers. The cloud SLA specifications are kept in the center while allocating the resources. Cloud resource management is a dynamic process that deals with locating and releasing resources in an environment, where the dynamics of the needs and specifications frequently change. The efficient and effective utilization of the resources in any computing model like the cloud is highly anticipated. Today greenness or energy efficiency of the resources has been declared as one of the most important QoS. The other issues in resource management are violations of SLA and efficient load balancing with high service availability, i.e., uptime 99% [28].

It is easy to procure the resources but difficult to deploy, deliver, and manage the customer workloads in cloud environments, where worldwide customers and their resource dynamics fluctuate within a very small quantum of time. The arrival of the CSPs such as Amazon, Microsoft, and Google, which are ranging from scientific applications to the business, commerce, industry, academia, and personal use, creates the need for ultra-advanced resource management solutions with complex systems design and management strategies. The SLAs specify the need specification of the cloud resources, and it is the sole responsibility of the CSPs to fulfill the resource requirement of the customers to maintain the trust and transparency for customer satisfaction. In the cloud environment, heterogeneity of the resources is the major challenge as they need well-designed and well-tested robust solutions

for complex system management. The convergence of performance data analytics and automatic resource management carries new challenges and opportunities. It becomes difficult for the system integration and management designers to transform the theoretical models and conceptions into practically implementable solutions. In order to achieve this, resource management in the cloud requires well-structured and agreed policies and efficient decisions for multi-objective optimization of resources. These policies can be categorized into five classes or processes: (1) admission control, (2) capacity allocation, (3) load balancing, (4) energy optimization, and (5) quality of service guarantees [29, 30]. This chapter covers the general concepts of cloud resource management and investigates the trust, transparency, and QoS in service-level agreements (SLAs) as a case-based experimental analysis.

2.2 Essential Concepts and Definitions

Since the beginning, the cloud models were designed by their built-in techno-business characteristics. Usually, the cloud infrastructures from its foundation are provided as utility computing resources and availed in cost-effective scale to utilize resource offering in a pay-per-use model [31, 32]. Several authors have defined the necessary components and their functionalities. According to [33], resource management comprises nine major processes:

- Resource provisioning: Assigning the desired resources to a workload based on need specifications.
- Resource discovery: Identification or discovery of a list of cloud resources that are available for workload handling or execution.
- Resource modeling: A standard framework that helps in predicting the resource specifications required by a workload based on attributes such as states, transitions, inputs, and outputs within a given cloud environment.
- Resource scheduling: Cloud resource scheduling can be defined as the mapping, allocation, and execution of workloads based on the cloud resources shortlisted (provisioned) in the provisioning phase. It can also be defined as a timetable of activities and resources, with start and end points along with the duration of the workloads. The quality attributes of services such as cost-effectiveness, timeliness, energy efficiency, etc. (i.e., as promised under service-level agreement (SLA)) are also aligned.
- Resource allocation: Balanced distribution of resources among competing workloads with minimum conflicts.
- Resource mapping: Negotiations between resources required by the workload and resources provided by cloud service providers.
- Resource estimation: Prediction of the actual resources required for executing a workload efficiently.

- Resource brokering: Arbitration or negotiation of cloud resources through a mediator entity (agent) to guarantee their availability at the right time to execute or handle the workload.
- Resource adaptation: Ability to dynamically adjust (elasticity) the desired resources to fulfill the dynamic requirements of the workload efficiently.

Complexity, the variety, and the nature of the data are dynamically changing. The scientific and business organizations nowadays rely on the analytics of the complex, varied, and voluminous data sets, and the processing must be done over on-demand scalable and auto-configurable computing resources. The substantial performance enhancement and overhead reduction in virtualization boosted its adoption as a key feature in cloud computing technology [34, 35].

There are salient underlying techniques, technologies, and their configurations that transform computing over the cloud in reality. Among these technologies, the most significant technologies are the virtualization of data center resources for access to enormous processing capabilities and scalable resources to handle complex data with unpredictable computing needs.

3 Functions of Cloud Resource Management

The main essence of resource management is to recognize the suitable resources for a specific workload to handle in the most proficient manner. The quality-of-service specifications are determined by the consumers. This process is known as provisioning of the most suitable computing resources [36].

As mentioned in Fig. 2, the cloud resource management consists of three main functions—provisioning, scheduling, and monitoring.

Fig. 2 Resource management in cloud computing environment [redesign]

3.1 Cloud Recourse Provisioning

In the resource provisioning, the first step is consumer authentication. Afterward, the consumer interacts with the cloud servers via a cloud portal and submits the resource requirements of the workload along with quality specifications. In this process, the Resource Information Centre (RIC) maintains the status of the pooled resources and provides this state information to the customer about the availability of the requested resources for handling or executing the workload. The Resource Provisioning Agent (RPA) is a responsible component that checks the availability of the requested resources by the customer, i.e., what is required and the states of availability. When the resource provisioning is over, the customer workloads are submitted to the next component, i.e., scheduler. Finally, the resource states information is submitted to the Workload Resource Manager (WRM) which forwards it to the Resource Provisioning Agent, and the final results are forwarded to the cloud customer.

3.2 Cloud Resource Scheduling

Cloud resource scheduling consists of the three processes, i.e., mapping, allocation, and execution of workloads, based on the cloud resources shortlisted (provisioned) in the aforementioned provisioning phase. This is usually performed aligning with quality attributes of services such as cost, timeliness, energy efficiency, etc. as promised under service-level agreement (SLA) [37].

The whole process consists of the three activities that are: (1) Mapping—selection of the suitable resources based on the quality-of-service specifications (i.e., mentioned in SLA) of the customer, (2) detection—identification or discovery of the list of available cloud resources, and (3) Selection—choosing the most feasible resource from the list produced by detection based on SLA.

3.3 Cloud Resource Monitoring

The monitoring and surveillance processes of cloud resources are also required to be autonomic. Cloud resource monitoring supports in achieving the desired performance as promised (i.e., SLA specifications). As per the standard agreements of SLA, both the parties (cloud service provider and cloud service consumer) must specify and agree on the possible deviations or violations in service terms and conditions so as to manage promised quality attributes in SLA and avoid the conflicts. Subsequently, this phase also controls the rescheduling of activities in the cloud environment.

This phenomenon state is necessary for the optimization of the trust and transparency in metering and monitoring of cloud resources consumptions. It is

envisioned that the violations or deviation must be less than the defined thresholds for successful execution of a workload in the cloud environment. Logically, resource monitoring is also one of the important quality attributes that should be taken care of seriously when trust and transparency are categorically mentioned as the essential QoS specifications like availability of services, uptime, and performance specifications, and security [7]. In the monitoring process, the existing workload states are compared to the number of required cloud resources. In the case of less, more resources are demanded by the resource scheduler so as to maintain the SLA provisions and promises. If the resources are sufficiently available in the pool, the resources can also be released and made available for allocations.

3.4 Resource Management Techniques/Methods

The cloud computing resource management has a variety of solutions and techniques that are accumulated and classified from the literature survey.

Effective and efficient resource utilization is confined to the optimization and assured by algorithms running in the cloud environment. The researchers [38] classified the cloud resource management into nine classes/categories. In this scheduling [39] solution, cost, time, success rate, scalability, make span, speed, resource utilization, reliability, and availability were considered. Usually, the reliability and availability have lots of similarities, but they were typically ignored; however, time, speed, and make span are sufficiently described as interconnected properties. The 12 properties were defined in the research [40] such as (1) *Time-based:* its deadline based by the blending of deadline and budget, (2) *Cost-based:* It is multi-QoS, application, virtualization, and scalability based; (3) *Compromised Cost: It is time* based either on workflows or workloads; (4) *QoS-based:* Created on several QoS aspects, such as resource utilization and security; (5) *SLA-based: Created on the baseline* SLA types, such as autonomic feature and workload; (6) *Energy-based: It connects the* deadlines and SLAs; (7) *Optimization-based*: It optimizes permutation and combinations of parameters; (8) *Nature and Bio-Inspired*: It includes the genetic algorithms and ant colony approaches; (9) *Dynamic*: It includes the dynamic aspects of resource management with salient permutation and combinations; (10) *Rule-based*: It considers the special cases for failures and hybrid clouds, (11) *Adaptive-based*: Prediction-based and Bin-Packing strategies; and (12) *Bargaining-based*: It is organized in market based, auction, and negotiations.

Another study of [41] classified the resource management solutions in relations to scalability, interoperability, flexibility, heterogeneity, localized autonomy, load balancing, information exposure, past scheduling records, unpredictability management, real-time data, geographical distribution, SLA compatibility, rescheduling, and intercloud compatibility. In this study, several properties are overlapping or correlated, such as rescheduling, scalability, and managing unpredictable phenomenon.

A research study of [42, 43] proposed nine categories to classify their references such as (1) *Best effort: Single* objective optimization by ignoring other factors;

(2) *Deadline constrained*: When the deadline is set, it schedules based on the execution time and monetary cost; (3) *Budget constrained:* finishing within budget, (4) *Multicriteria:* combining many objectives together, (5) *Workflow as a service:* A moment when the resource manager receives many workflow instances to perform; (6) *Robust scheduling*: Capability of handling uncertainties such as performance fluctuation and failure together; (7) *Hybrid environment*: ability of handling hybrid cloud requirements; (8) *Data intensive*: scheduling with data-aware workflows; and (9) *Energy aware*: ability of greenness while optimizing execution.

After rigorous review and analysis of resource management techniques, it is clear that this field still lags behind in terms of trust, transparency, and QoS in resource management and needs vigorous improvements and extensions in the existing techniques and methods. The claim becomes significantly more important when we explore cloud computing applications in a wider spectrum of multi-cloud and industry 4.0. The cyber-physical production system that combines ICTs, cyberspace, and intelligent systems is expanding the pathways of Industry 4.0 in salient dimensions toward multidimensional revolutions like traditional manufacturing to intelligent system-supported manufacturing.

Also, the convergence of the Internet of Things into Industry (i.e., Industrial Internet of Things (IIoTs)) has created new ways of computing, communication, collaboration, and control toward a new era of automation. In order to comprehend these transformations into reality, a wide range of resource management optimization and dynamics of connected resources will need reengineering. The existing process of computing and communication in automated environments such as cloud, fog, and due computing needs serious attention and collaborative research on salient tiers of research such as management and effective monitoring and control over the Service-Level Agreement (SLAs) for efficient and effective communication and interaction among the mobile system components and devices in autonomic manners [44, 45].

3.5 Service-Level Agreements (SLAs) Gaps in Cloud Computing

In general, a service-level agreement (SLA) is the bond for performance arbitration between the CSPs and the customer. Most of the SLAs are standardized. The SLAs are also categorized at different levels: (1) Client-side SLA, (2) Service-level SLA, and (3) Multilevel SLA. Most of these contracts are more along the lines of operating-level agreements (OLAs) and may not be restricted by the court of law. On ample occasions, these SLAs are violated, and therefore they need to have an attorney to review before agreeing to the CSPs. The SLA contracts usually specify some measuring/metering parameters such as availability of the Service outage (uptime), the Response time (latency), QoS (greenness), Service Configuration, Service components reliability, and Warranties. If a CSP

fails to meet the stated/warranted requirements of minimums, then the CSP has to pay the compensation/penalty to the consumer. Microsoft publishes the Service-Level Agreements linked with the Windows Azure Platform components, which is demonstrative of industry practice for cloud service vendors. Each individual component has its own Service-Level Agreements such as Windows Azure SLA and SQL Azure SLA [46].

Effective cloud resource management needs robust implementation techniques to manage the resources of cloud data centers. However, the Service Level-Objective (SLO) is a judicious range in order to achieve optimum performance in business service operations.

The energy efficiency and ineffective resource metering, monitoring, and utilization can build better trust and transparency among CSPs and Cloud Service Consumers (CCC); however, it can lead to an increase in the operational costs. Also, an increase in the cloud resource utilization needs energy efficiency, as quality-of-service (QoS) parameters are nowadays towards green computing initiatives. However, combining cloud resources such as virtual machines can cause a serious violation of SLAs [47, 48].

The cloud service providers (CSPs) are responsible for metering and monitoring of the consumed cloud resources. The cloud resources for computing, communication, storage, and other purposes need efficient and effective frameworks for metering and monitoring mechanisms. The metering and monitoring systems must utilize the trusted and transparent scales so that the scheduling of globalized or localized resource allocation and utilization can be optimized.

It was envisioned that this rapid growth in the technology sectors will warrant a substantial techno democratic environment in terms of trust, transparency, and empowerment of loyal consumers. As a matter of fact, most of the computing and communication system services have been metered and billed in a monopolized method by the service provider companies/enterprises. Most often, the consumers have to believe in the metered service measurements and pay the bills accordingly. What if the service-level agreements are violated in terms of promises between the service provider and the consumer, and metering system reading outputs are manipulated? In most of the service-level agreements, a cross-verification or metered data tally at the client side is still at the embryonic stage toward judicious empowerment of the consumer rights. Also, the weak steps of statutory compliance, settlement, and decree make violations very thoughtful in the consumer market. A system for preserving such trust and transparencies in functional and nonfunctional attributes is highly anticipated in Telecom, ICT, and Clouds service sectors.

Since the cloud resources are deployed on the virtual infrastructure, therefore, consumers lack or have limited privileges of metering and monitoring consumed services and resources. CSPs have domination, and therefore there are ample chances of violations or deviations on dynamical alterations in the prices charged for leasing the infrastructure, while cloud users can alter the costs by changing application parameters and usage levels only. However, the cloud consumers have limited privilege for resource management, being embarrassed to generate workload requests and control when and where the workloads are to be found.

The client-side monitoring and tally system is required to be implemented by regulatory authorities, and both the client and the service providers must abide by the monitoring and tally system regulatory framework and mechanisms. This system can enhance the trust and transparency in metering and billing systems for the betterment of consumer rights.

3.6 The CSPs SLA Monitoring Mechanism (Table 1)

3.7 Client-Centric SLA Framework for Enhancing the Trust, Transparency, and QoS

Figure 3 presents a proposed Client-Centric SLA Framework. The framework labels the metering and monitoring of consumed cloud resource services in an enhanced democratic manner to empower consumer rights. The client-centric SLA systems can help support in cross-verifying the warranties of the quality-of-service (QoS) attributes promised in SLAs such as energy efficiency and standard certifications. The proposed framework is the vigorously unique conceptualization of cloud consumer empowerment through incremental growth in managing the trust and transparency in metering and monitoring of consumed cloud resource services.

3.8 The Client-Centric SLA Framework

The prime aim of the framework is to empower the client rights in a trusted and transparent manner with QoS. This framework is an effort toward the betterment of two-party business relations, i.e., between cloud service consumers (CSCs) and cloud service providers (CSPs). Cloud Service Providers (CSPs): CSPs as mentioned in Fig. 3 are cloud service provider organizations that need to be regulated for consumer rights. The CSPs deliver cloud services to the consumers per their business workload specifications and requirements as guaranteed in SLAs. In general, the CSPs collect the consumer feedbacks and store them in the Service Management Databases (SMDB). This feedback is analyzed for consistent improvements in the cloud service delivery mechanisms.

Cloud Service User (Customers/Consumers): The cloud service consumers or customers are the end users of CSP services and may be an individual or organizational entity that maintains a business relationship with CSPs to consume cloud services. Generally, the existing state-of-the-art metering and monitoring systems at CSPs are single sided, monopolized with the lack of consumer rights, and empowered in service provisioning and monitoring. In today's democratic system governance and management environment, consumerism is an essential stratum, and consumer trust and transparency should be maintained for the vibrant ecosystem

Table 1 Describes the cloud service provider and monitoring of SLA [49]

CSP	Types of service	Step 1 dis-covering the service provider	Step 2 defining SLA	Step 3 making agreement	Step 4 monitoring the violations of SLA	Step 5 terminate SLA	Step 6 penalty for SLA violation
Amazon Ec2	Computing (IaaS)	Discovering manually via website	Predefined SLA terms and QoS specifications	Predefined SLA document of CSPs.	Third-party monitoring system (e.g., cloud watch) can be used	Monitoring by third regulatory party/using program by CSP or manually by both the parties	Additional service crediting by CSP or refunds or legal actions
Amazon S3	Storage (IaaS)	Discovering manually	Predefined SLA terms and QoS specifications	Predefined SLA document of CSPs.	Third-party monitoring system (e.g., cloud status) can be used	Monitoring by third regulatory party/using program by CSP or manually by both the parties	Additional service crediting by CSP or refunds or legal actions
MS azure compute	PaaS	Discovering manually (e.g., website)	Predefined SLA terms and QoS specifications	Predefined SLA document of CSPs.	Third-party monitoring system (e.g., monitis) can be used	Monitoring by third regulatory party/using program by CSP or manually by both the parties	Additional service crediting by CSP or refunds or legal actions
MS azure storage	PaaS	Discovering manually	Pre-defined SLA terms and QoS specifications	Predefined SLA document of CSPs.	Third-party monitoring system (e.g., monitis) can be used	Monitoring by third regulatory party/using program by CSP or manually by both the parties	Additional service crediting by CSP or refunds or legal actions
MS azure software	SaaS	Discovering manually	Pre-defined SLA terms and QoS specifications	Predefined SLA document of CSPs.	Third-party monitoring system (e.g., monitis) can be used	Monitoring by third regulatory party/using program by CSP or manually by both the parties	Additional service crediting by CSP or refunds or legal actions

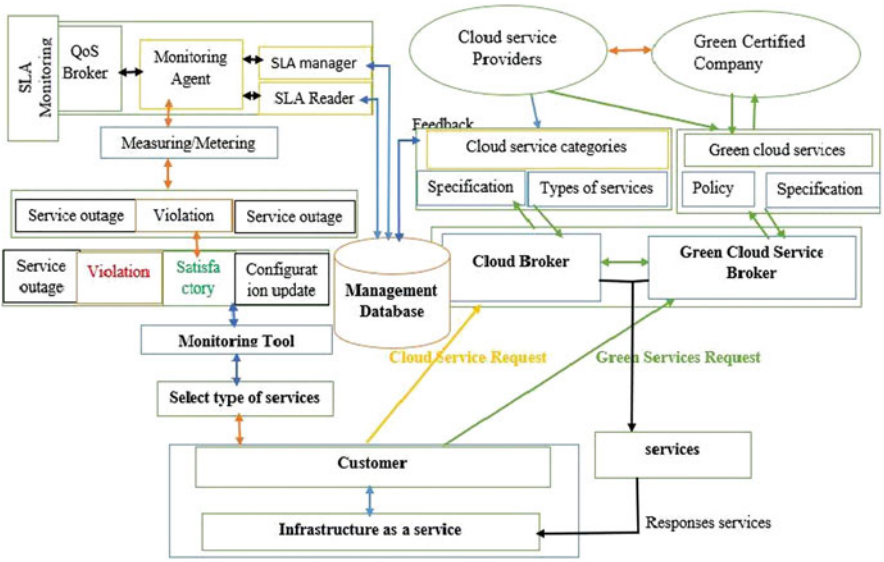

Fig. 3 Client-centric SLA framework

in the CSP industry. It can minimize the conflict between consumerism and professionalism. The proposed framework (i.e., in Fig. 3) proposes a new way for client-side metering and monitoring of consumed cloud services. This framework can be implemented as a metering and monitoring tool on cloud consumer devices. The central cloud or IT regulatory agencies may authorize/certify as a tally or auditing tool for better relationships among consumers and CSPs. This framework "Client-Side SLA" will empower the consumers judiciously.

Cloud Brokers: There are two types of brokers, i.e., (1) Cloud Service Broker and (2) Green Cloud Broker. This component of the framework (i.e., in Fig. 3) proposes to offer the most suitable services with better prices, efficiency, and QoS based on needs or service specifications of the client/consumer. This component has two major responsibilities as a typical cloud service broker with an additional QoS feature, i.e., energy efficiency/ green certification of cloud resources or services. The green cloud service broker verifies the suitability of the consumer's energy efficiency specifications/green certifications and recommends the services based on their preference and specifications included in SLA. Further, the Green Cloud Service Broker (GCSB) verifies greenness of services declared by CSPs with the certification issued by competent authorities and generates a validation certification to filter the false claims [50–52].

SLA Metering (Measuring) and Monitoring Agent: The framework as presented in Fig. 3 consists of three different agents, i.e., (1) SLA Readers, (2) Monitoring Agent, and (3) QoS Broker. The functionality of different agents may vary from one CSPs to another. SLA reader reads the signed SLA from the CSPs and Consumer both which are stored in the database having the exact value of

parameters with full transparency and trust via the Internet. It feeds the signed SLA to the monitoring agent. The QoS Broker is responsible for monitoring the nonfunctional requirements (i.e., greenness of services and others) and collects data from the customer and disseminates the information to the CSPs.

Layers of SLA Monitoring Framework: The functionality of the layers may vary as they work based on the assigned values. As presented in Fig. 3, the following layers are included in the framework:

Application Layer: The logic tier is pulled out from the presentation tier and, as its own layer; controls an application's functionality by performing detailed processing. The application layer receives the results from the lower layer. The Metering and the Monitoring agent provide notifications about SLA state (violated or not?). So, the application layer forwards the results of the monitoring/ agent to the consumers which are displayed by the presentation layer. Actually, presentation is the topmost level of the application. The presentation layer displays the results of browsing merchandise, purchasing, and shopping cart contents.

Service Management Layer: The monitoring layer provides the results to the upper layer (i.e., the presentation layer). The monitoring layer includes different types of components such as SLA Reader, QoS Broker, and Monitoring Agent. The QoS Broker gathers the data from the client and disseminates to the CSPs to set judicious compensation for the violated services.

Database Layer: This layer stores information of the SLA with the exact parameter specified in the SLA. The database contains SLAs that are specified by the CSPs and Cloud Service Consumer. Thus, client-side measuring and monitoring SLA can improve the trust and transparency between the involved parties. The following diagram presented in Fig. 4 presents the metering and monitoring layers of clients-side SLA.

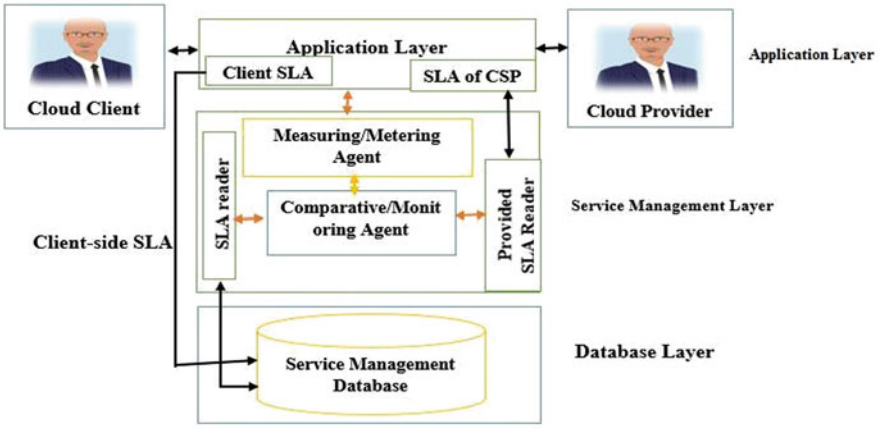

Fig. 4 Metering and monitoring layers of client-side SLA

4 Experimental Analysis and Discussions

For the testing of the conceptual framework, the client-side SLA was implemented over cloud-based AppNeta. The Alternative to AppNeta, the tools that were studied, analyzed, and compared with parametric analysis are (1) Microsoft System Center, (2) Datadog, (3) LogicMonitor, (4) ThousandEyes, (5) NinjaRMM, (6) Zabbix, and (7) Wireshark.

Finally, AppNeta was selected as a cloud tool to monitor and manage applications and network performance. The first experimental setup test results of AppNeta for AWS and Azure Cloud Data Centre are presented in Fig. 5. The test results were recorded for a week from October 25, 2019, to October 31, 2019. In this test, some of the selected services were ordered by a customer to Amazon AWS. In this experimental test, as presented in Fig. 5, the (1) service outage—0.192%, (2) SLA violation—2.154%, (3) satisfactory services—93.531%, and (4) configuration update—4.122% over cloud data centers were recorded. Also, in an ideal state, 99.99% (uptime), i.e., the service availability was promised in cloud data center SLAs, but in actual it was recorded as 95.686%. The experiments confirmed the

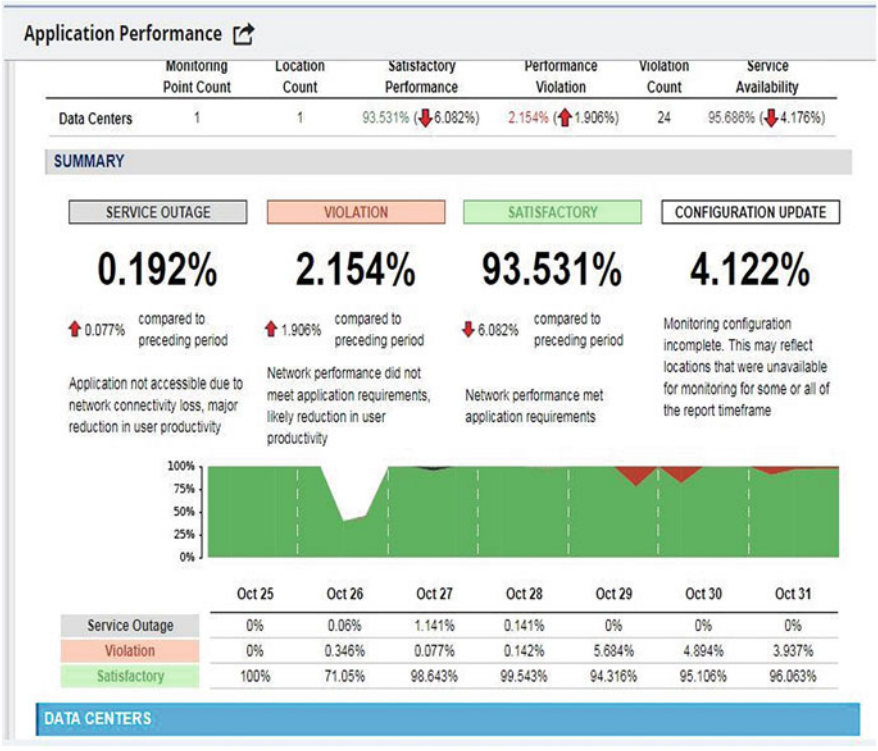

Fig. 5 Experiment 1- AWS data center measured services by AppNeta on the client-side machine

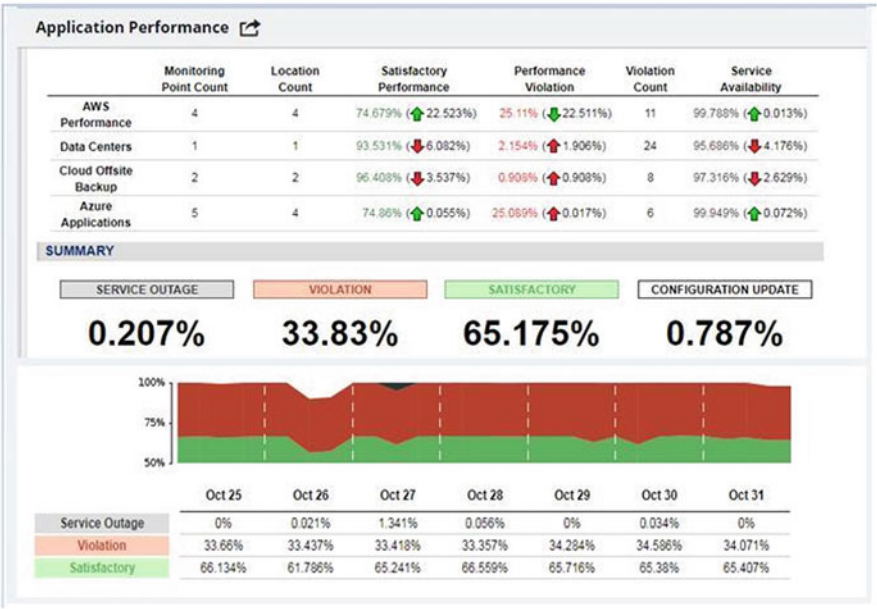

Fig. 6 The applications performance on Amazon AWS and MS Azure run on the client-side machine

SLA violation, and therefore using the measured/metered results on the client-side machine, the customer can ask the compensation for the deviation of the terms and standard promised in SLA. Or otherwise, case customers can terminate the contract agreements.

In the second experimental test (Fig. 6), another CSP, i.e., MS Azure and Amazon AWS, was considered. The test results were recorded for a week from October 25 to 31 October 2019. In this test, some of the selected services were ordered by a customer to Amazon AWS & MS Azure. This CSPs promised 99.99% (uptime) service availability. In this experimental setup, the test results at Azure and AWS cloud data center services by the AppNeta were recorded as: (1) service outage—0.207%, (2) SLA violation—33.83%, (3) satisfactory services—65.175%, and (4) configuration update—0.787%. This clearly implies that if cloud consumers have their own metering and monitoring tools for SLA, they can negotiate with CSPs either in terms of compensation for the service violation or terminate the agreement. The consumers can ask for compliance settlement through regulatory authorities or negotiations and hence the trust and transparency will be enhanced. The performance of applications was also observed and recorded in this experimental setup. As presented in Fig. 6; the fluctuations (ups and downs) and violations can be clearly observed in AWS performance, data center services, cloud off-site backups, and Azure applications.

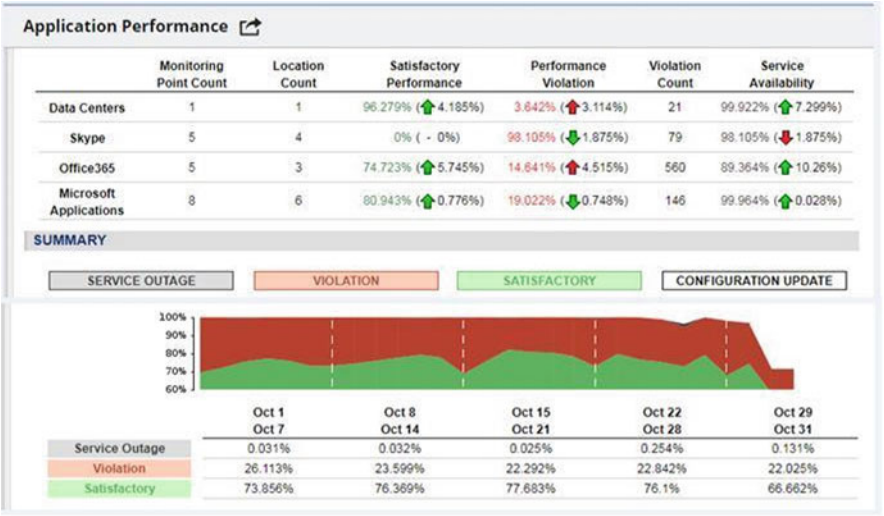

Fig. 7 Amazon application performance metering on the client-side machine (month report)

Fig. 8 Performance of the Amazon AWS and MS Azure cloud service providers

Another experimental setup presents the metered result for the application performance of the services provided by Amazon AWS. The result presented in Fig. 7 clearly indicates that SLA violation is at second position and satisfaction of services is in the first position. Here service outage is negligible. The customer can also view the performance of the cloud services provided by CSPs.

The experimental results in Fig. 8 present the performance of both Amazon AWS and MS Azure. In addition, the customer can also measure and monitor the service performance to check which performs better before they place an order and sign SLA.

The following experimental results in Fig. 9 present the performance of the Amazon AWS measured for one week of the Service Outage, violated and Satisfactory

The experimental result presents the performance of MS Azure products and services. Using the following result, the customer can decide which service provider is the best or the most suitable based on consumer requirement specifications. Based on the one-week reports presented in Fig. 10, the SLA promised 99.99%

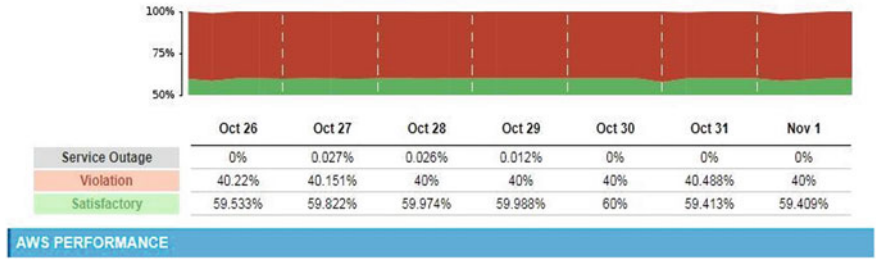

Fig. 9 The Performance testing of the Amazon AWS for one week on the client-side machine

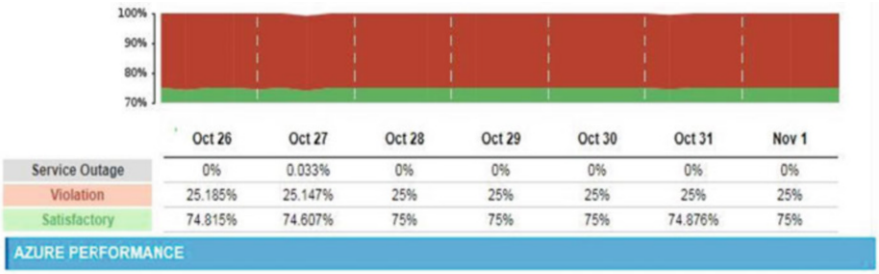

Fig. 10 The Performance measurement of MS Azure for a one-week report on the client-side machine

(uptime) but measured and found 25.1% (uptime) which is a typical violation. The satisfactory result of the client side is 74.899 % out of the promised result of 99.999% which is written in the Service-Level Agreement.

According to the result presented in Fig. 11, the MS Azure almost failed to fulfill the promised performance mentioned in SLA. According to 1-week report, the service outage found was 6.868%, violated result was 55.568%, and the satisfying result was 36.524%. According to the energy efficiency standards, benchmark, and measurements, the power usage Effectiveness (PUE) can be calculated by cloud-based metering greenness/energy efficiency tools such as 42U. The measurement results can be compared to the energy efficiency declared by CSPs of Data Centre resources. The customers have the right to terminate the cloud service agreement or to ask for the compensation credits for the greenness violations or service outage results.

5 Challenges in Data Centre Resource Management

Resource management in a cloud environment is a typical challenge due to the following issues in the modern data centers [27, 53].

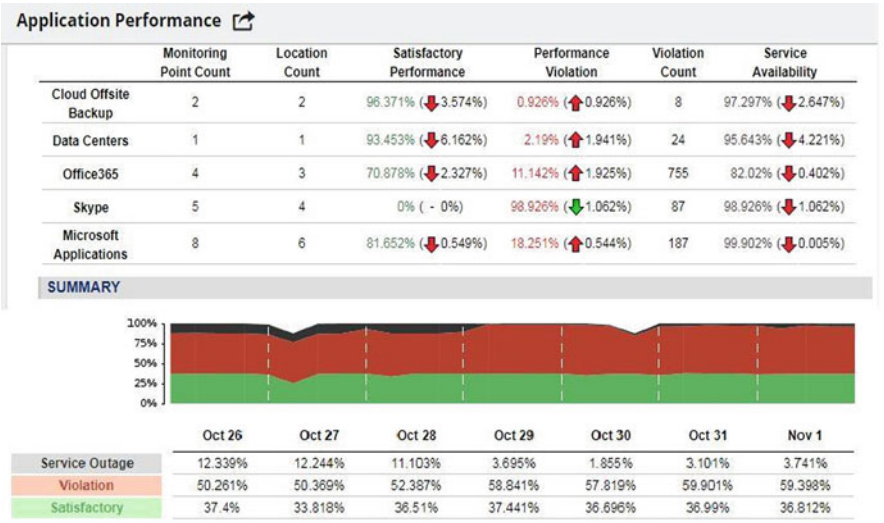

Fig. 11 The application performance on the MS Azure platform for the 1-week report

- The fault tolerance of the resources
- The interoperability and the interdependence of the resources
- The high level of scalability of the resources
- The heterogeneity of the resource in multicloud
- The variability and unpredictability of the resource load
- The salient players and multi-objectivity in a cloud ecosystem
- The data-intensive workflows
- The energy-aware resource scheduling
- The reliability to performance fluctuations
- Communication and transfer costs of resources
- The dominance of the execution time and cost
- The data placement strategies design for decision-making while resource provisioning
- The performance fluctuations in multitenant resource sharing
- The workflow scheduling in the management of workflow execution in cloud environments

6 Conclusion

This chapter provides an exhaustive investigation and analysis of concepts, definitions of cloud resource management, and the review of existing techniques of management, SLA, and violations. The chapter started from evolution of the concepts, defining the terms and references on the subject area, covering the

basics of the foundation published in salient publications from the research and academia. Among the salient common tasks in resource management, each phase of the resource life cycle, such as resource discovery, allocation, scheduling, and monitoring, is also covered. Moreover, the critical objective in all cases is to enable task execution while optimizing infrastructural efficiency. These most important issues related to cloud resource management are also covered. A rigorous review of literature is incorporated for the characterization and selected solutions of the pinpointed issues, challenges, and gaps in resource management. Finally, the chapter concluded that trust, transparency, and QoS (greenness) in cloud resource metering and monitoring are necessary. The experimental analysis of the proposed framework for the client-side metering and monitoring of SLA proved that there are violations in the metering of cloud services along with the QoS claimed and the QoS provided. The serious violations are observed at CSP sides, but neither clients nor CSPs have abundant attention to resolve such issues seriously for the betterment of the trust and transparency between consumers and CSPs. The essential solutions must be designed, developed, and deployed with future research recommendations in high dynamic and scalable environment of cloud resource management.

7 Unanswered Questions as Recommendations for the Future Research Efforts

- How to discourse the particularities of data-intensive workflows and address the particularities of large-scale cloud setups with more complex environments in terms of resource heterogeneity and distribution, such as hybrid and multicloud?
- How to handle the fluctuations in workflow progress due to performance variation and reliability and to maintain reliability based on actual and measurable metrics.

References

1. Marinescu, D. C. (2013, July 8). *Cloud computing: Theory and practice*. Cambridge, MA: Elsevier.
2. Kumbhar, P. (2019, July 6). Summary of computing laws amdahl, dennard, gustafson, little more and more.
3. Dinote, A., Sharma, D. P., Gure, A. T., Singh, B. K., & Choudhury, T. (2020). Medication processes automation using unified green computing and communication model. *Journal of Green Engineering, 10*(9).
4. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
5. Chana, I., & Kaur, T. (2013, May). Delivering IT as a utility – A systematic review. *IJFCST, 3*(3), 11–30.
6. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing – data engineering and intelligent. *Computing*.

7. Yadav, A. K., Tomar, R., Kumar, D., & Gupta, H. (2012). Security and privacy concerns in cloud computing, *Computer Science and Software Engineering, 2*(5).
8. Wikipedia. (2013). Cloud computing. The free encyclopaedia.
9. Ward, J. S., & Barker, A. (2014). Observing the clouds: A survey and taxonomy of cloud monitoring. *Journal of Cloud Computing, 24*(3).
10. unctad. (2019). Digital economy report. United Nation.
11. Douglas, D. D., & Barry, K. (2013). Cloud computing model. *Science Direct*.
12. Mell, P., & Grance, T. (2011). *The NIST definition of cloud computing*. Gaithersburg, MD: National Institute of Standards & Technology.
13. Sharma, D. P., Sharma, R. K., & Ayodele, A. (2008). Convergence of intranetware in project management for effective enterprise management. *Journal of Global Information Technology (JGIT)-USA, 4*(2), 65–85.
14. Amin Tuni, D. P. S. (2019). Assessment of knowledge sharing practices in higher learning institutions: A new exploratory framework–AT-DP KSPF. *The IUP Journal of Knowledge Management, 17*(4), 7–20.
15. Intelligence, D. M.. (2020, June 12). Cloud migration services market, size, share, opportunities and forecast. *DDIC131*.
16. Dewangan, B. K., Agarwal, A., Choudhury, T., & Pasricha, A. (2020). Cloud resource optimization system based on time and cost. *International Journal of Mathematical, Engineering and Management Sciences, 5*(4). https://doi.org/10.33889/IJMEMS.2020.5.4.060
17. Wadhwa, M., Goel, A., Choudhury, T., & Mishra, V. P. (2019). Green cloud computing-A greener approach to IT. 2019 international conference on computational Intelligence and knowledge economy (ICCIKE) (pp. 760–764).
18. Kaur, A., Raj, G., Yadav, S., & Choudhury, T. (2018). Performance evaluation of AWS and IBM cloud platforms for security mechanism. 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS) (pp. 516–520).
19. Deelman, E., Gannon, D. B., Shield, M., & Taylor, I. J. (2014). *Workflows for E-science: Scientific workflows for grids*. London: Springer.
20. Li, Y., Raicu, I., Lu, S., Tian, W., Liu, H., & Zhao, Y. (2015). Enabling scalable scientific workflow management in the cloud. *Future Generation Computer System, 46*, 3–16.
21. Bubendorfer, K., & Arabnejad, V. (2015). Cost effective and deadline constrained scientific workflow scheduling for commercial clouds. In: Network Computing and Applications (NCA). *IEEE, 14th International Symposium On* (Vol. 33, pp. 106–113).
22. Shyam, G. K., & Manvi, S. S. (2014). Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. *Journal of Networking and Computer Application, 141*, 424–440.
23. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2018). Privacy and security of cloud-based internet of things (IoT). In Proceedings – 2017 international conference on computational intelligence and networks, CINE 2017. https://doi.org/10.1109/CINE.2017.28
24. Bansal, S., Gulati, K., Kumar, P., & Choudhury, T. (2018). An analytical review of PaaS-cloud layer for application design. In Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358374
25. Chard, K., Bubendorfer, K., Lacinski, L., Madduri, R., Foster, I., & Chard R. (2015). Cost-aware elastic cloud provisioning for scientific workloads. In IEEE 8th international conference on cloud computing (Vol. 130, pp. 971–974).
26. Yi, S., Andrzejak, A., & Kondo, D. (2012). Monetary cost-aware checkpointing and migration on amazon cloud spot instances. *IEEE Transactions on Services Computing, 15*(4), 512–524.
27. Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and System Management, 23*(3), 567–619.
28. Mustafa, S., Nazir, B., Hayat, A., Khan, A. R., & Madani, S. A. (2015). Hayat resource management in cloud computing: Taxonomy, prospects, and challenges. *Computer and Electrical Engineering, 47*, 186–203.

29. Marinescu, D. C. (2013). *Cloud computing: Theory and practice*. Waltham: Morgan Kauffman.
30. Asres, K., Gure, A. T., & Sharma, D. P. (2019). Automatic surveillance and control system framework-DPS-KA-AT for alleviating disruptions of social Media in Higher Learning Institutions. *Journal of Computer and Communications, 8*(1), 1–15.
31. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010). A view of cloud computing. Communications of ACM. New York.
32. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., et al. (2010). A view of cloud computing and communication. *ACM, 53*(4), 50–58.
33. Manvi, S. S., & Shyam, G. K. (2014). Resource management for infrastructure as a service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Application, 141*, 424–440.
34. Chieu, T. C., Mohindra, A., Karve, A. A., & Segal, A. (2009). Dynamic scaling of web applications in a virtualized cloud computing environment. Washington, DC.
35. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., & Zagorodnov, D.. (2009). The eucalyptus open-source cloud-computing system. Shanghai.
36. Chana, I., & Singh, S. (2014). Quality of service and service level agreements for cloud environments: Issues and challenges. In Challenges, limitations and R&D solutions, Switzerland.
37. Chana, I., & Singh, S. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing, 14*(2), 217–264.
38. Bala, A., & Chana, I. (2011). A survey of various workflow scheduling algorithms in cloud environment In Nagpur.
39. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
40. Singh, S., & Chana, I. (2016). Cloud resource provisioning: Survey, status and future research directions. *Knowledge and Information System, 49*(3), 1005–1069.
41. Sotiriadis, S., Bessis, N., & Antonopoulos, N. (2011). Towards inter-cloud schedulers: A survey of meta-scheduling approaches. In 2011 international conference on, Barcelona.
42. Wu, F., Wu, Q., & Tan, Y. (2015). Workflow scheduling in cloud: A survey. *The Journal of Supercomputing, 71*(9), 3373–3418.
43. Shekhawat, H. S., & Sharma, D. P. (2012). Hybrid cloud computing in E-governance: Related security risks and solutions. *Research Journal of Information Technology, 4*(1), 1–6.
44. Wan, J., Chen, B., Imran, M., Tao, F., Li, D., Liu, C., & Ahmad, S. (2018). Toward dynamic resources management for IoT-based manufacturing. In IEEE Communications Magazine (Vol. 56(2), pp. 52–59). IEEE.
45. Raptis, T. P., Passarella, A., & Conti, M. (2019). Data management in industry 4.0: State of the art and open challenges. In Access (Vol. 7, pp. 97052–97093). IEEE.
46. Ibrahim, A. A. Z. A., Kliazovich, D., & Bouvry, P. (2016). Service level agreement assurance between cloud services providers and cloud customers. Cartagena.
47. Mandal, R., Mondal, M. K., Banerjee, S., & Biswas, U. (2020). An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing. *Journal of Super Computer, 76*, 7374–7393.
48. Daraghmeh, M., Melhem, S. B., Agarwal, A., Goel, N., & Zaman, M. (2018). Linear and logistic regression based monitoring for resource Management in Cloud Networks. Barcelona.
49. Wu, L., & Buyya, R. (2012). *Service level agreement (SLA) in utility cloud systems* (p. 25). IGI Global: Melbourne.
50. Gure, A. T., & Sharma, D. P. (2019). Assessment of knowledge sharing practices in higher learning institutions: A new exploratory framework–AT-DP KSPF. *The IUP Journal of Knowledge Management, 17*(4), 7–20.
51. Muda, J., Tumsa, S., Tuni, A., & Sharma, D. P. (2020). Cloud-enabled E-governance framework for citizen centric services. *Journal of Computer and Communications, 8*(7), 63–78.

52. Chakraborty, P. (2017). Simulation of reducing broadcasting protocol in ad hoc wireless networks. *International Journal of Scientific & Engineering Research, 8*(7), 295–301.
53. Wu, F., Wu, Q., & Tan, Y. (2015). Workflow scheduling in cloud: A survey. *Journal of Super Computer, 71*(9), 3373–3418.

Autonomic Computing in Cloud: Model and Applications

G. Sobers Smiles David, K. Ramkumar, P. Shanmugavadivu, and P. S. Eliahim Jeevaraj

1 Introduction

The latest trending technologies of the twenty-first century are ubiquitous high-speed networks, smart devices, and edge computing. The beginning stages of the twenty-first century saw the emergence of technologies that allowed mobility, an ability to access Internet and information anytime, anywhere, and anyhow. Cloud computing brought in a fresh wave of computing, where a slice of the resource could be utilized whenever needed and released after its use, bringing down the cost associated with resource management. The best part of cloud computing is that it can deliver both software and hardware as a service on a pay-per-use manner. The services are abstracted in three levels as infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). The benchmarks set by cloud in offering these services are guaranteed service availability and quality, irrespective of the cloud dynamics such as workload and resource variations [1].

Cloud software deployment and service involve configuring large number of parameters for deploying an application, and error-free application configuration is vital for successful delivery of cloud service with quality. This task when done manually resulted in 50% service outages due to human mistakes [2]. The parameters are on the rise and increasingly becoming difficult for a human being

G. Sobers Smiles David · P. S. Eliahim Jeevaraj (✉)
Bishop Heber College, Tiruchirappalli, India
e-mail: eliahimps.cs@bhc.edu.in

K. Ramkumar
Villa College, Male, Republic of Maldives
e-mail: ramkumar.krish@villacollege.edu.mv

P. Shanmugavadivu
Gandhigram Rural Institute – Deemed to be University, Gandhigram, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_3

to keep track. Consider, for Apache servers, there are 240 configurable parameters to be set up and for Tomcat servers, there are more than 100 parameters to be set up and monitored. These parameters are very crucial for the successful running of servers, as they are support files, are related to performance, and are required modules [3]. To be able to find the appropriate configuration for a service, the person in charge of configuring must be thorough in the available parameters and their usage. Complexity is also driven by the diverse nature of the cloud-based web applications.

Chung et al. [4] demonstrated how no single universal configuration is good enough for all the workloads in the web. Zheng et al. [5] showed that it is necessary to update the number of application servers whenever the application server tier was updated in a cluster-based service. It is also important that the systems configuration must also be updated to induct the change in the number of servers. Compared to traditional computing, Cloud computing offers varied level of services but also presents new challenges in application configuration. On-demand hardware resource reallocation and service migration are necessary for this new class of services, and virtualization offers scope for providing these services. Maximizing resource utilization and dynamic adjustment of resources allocated to a VM are the desired features to ensure quality of service. If the application configuration is taken care by automatically and adaptive response for dynamic resource adjustment and reallocation is automatically done, the services could be guaranteed with quality [6]. Autonomic computing is the answer to all the issues mentioned.

The National Institute of Standards and Technology's (NIST) definition of cloud computing [7] has earmarked five essential resource management characteristics for cloud computing. The summary of the characteristics and the requirements are shown in Table 1.

In cloud environment [8], resource management is not a quick process and not an easy task to accomplish. However, the system performance is completely dependent

Table 1 Requirements for dynamic resource management in cloud

	Characteristics	Requirements	Objectives
Cloud computing	On-demand self-service	Intelligent and business-related resources management	Cost optimization and quality of service (QoS) guarantee
	Broad network access	End-to-end resource chain management and resource location optimization	
	Resource pooling	Dynamic deployment management	
	Rapid elasticity	Dynamic adaptive resource management	
	Measured service	Monitoring and reporting of resource usage	

on efficient resource management [9]. In order to provide accurate resource to performance mapping, there are challenges to be overcome. The first challenge is the varying demand of multiple resources due to a mix of hosted applications and its varying workloads [10]. For busy applications, the relationship between performance and resource allocation is inherently nonlinear. The second challenge is performance isolation between coresident applications. Although many existing virtualization techniques such as VMware, KVM, and Xen provide many services like security isolation, environment isolation, and fault isolation, performance isolation is not available. In such scenario, hypervisor deprivation, resource contention could possibly result in performance degradation [11]. The third challenge is to overcome the uncertainty associated with the cloud resources [12]. In the front end, the resources appear as a unified pool, but the background scenario is a complicated process. The Cloud resources are multiplexed, and virtualization of heterogeneous hardware resources is done in the background. Due to this, over a period of time, the actual resources available to hosted applications may vary. Adding to the complexity is the fact that the cloud resource management process is not an independent one. It is interlinked with the management of other layers. A coordinated strategy for configuration management for VM applications is needed. The solutions to resource provisioning issues in cloud computing are summarized in Table 2.

Table 2 Metrics for resource provisioning management in cloud

Name of the Scheme	Functionality
Fault-Tolerant Scheduling [1]	It is dynamic and provides high resource utilization and high schedulability
Online Auction Framework [2]	It follows auction mechanism and provides optimized system efficiency
Meta-heuristic Optimization Technique [13]	It follows particle swarm optimization (PSO) technique and provides minimized workflow execution cost
Agent-based automated service composition (A2SC) [14]	It follows A2SC algorithm and provides reduced VM service cost
Fault Aware Resource Provisioning [15]	It follows FARP algorithm and provides service to redirect the user'
On-Demand Provisioning [16]	It follows augmented shuffled frog leaping (ASFLA) algorithm and provides minimized cost for execution and time
High bandwidth Provisioning [17]	It follows polynomial time energy-aware routing algorithm and provides energy-aware paths to allocate servers and switches for accommodating traffic requests
Fault Tolerance, Fault Detection, and Fault Recovery [18]	Provides self-healing, self-detection, preemptive migration, checkpoint, restart, replication, system node recovery, and job migration techniques
Hybrid Approach [5]	Combination of autonomic computing and reinforcement learning (AC & RL) that could predict the future demands for the cloud services

2 Autonomic Computing Models

Advanced Research Projects Agency (ARPA) is a research body of the Department of Defense, US Government. They research on innovative ways for communication and develop communication systems for the US military. The research done at the ARPA lab is the pioneer research in the modern communication network area. ARPA developed the OSI Model and TCP/IP Model in the late 1950s of the nineteenth century. The whole world of communication follows OSI Model and TCP/IP Model. Similarly, in 1997, ARPA was assigned a project to develop a system with the capability of situational awareness. The system was called "Situational Awareness System (SAS)." As with any other ARPA project, the aim of SAS project was to create devices that aid personal communication and location services for their military personnel. These devices were designed to be used by soldiers on the battlefield. The devices include Unmanned Aerial Vehicles (UAV), Sensors, and Mobile communication systems. The devices could communicate with each other. The data could be collected by all the three devices and shared among them. The latency goal was set below 200 ms [19].

The most important challenge to overcome is that the communication has to take place in enemy's land. This brings two challenges: (1) overcome jamming by the enemy and communicate and (2) minimize interception by the enemy. To overcome the jamming, it was proposed to widen the communication frequency to 20–2500 MHz and bandwidth to 10–4Mbps. In order to overcome interception, it was proposed to employ multi-hop ad hoc routing. In this, every device could send to its nearest neighbor and so on till all the devices are reached. Since the latency was set below 200 ms, the self-management of the ad hoc routing emerged as a huge challenge. The on-field communication process could involve 10,000 soldiers [13].

In the year 2000, ARPA began working on another self-management project [14] named "Dynamic Assembly for System Adaptability, Dependability, and Assurance (DASADA)." This research brought architecture-driven approach in Self-management of Systems. The aim of this project was to research and develop mission-critical systems technology.

Autonomic computing is a term coined by IBM in 2001 to describe computing systems that can manage themselves. Autonomic refers to the automatic reflexes in our body. It is a biological term. In order to overcome the complexities involved with human beings in managing the cloud systems, systems that can manage the functioning of the cloud on their own have emerged. Such systems are referred to as autonomic systems. Autonomic computing includes capabilities such as self-configuration, self-healing, self-optimization, self-protection, and self-awareness to name a few. Autonomic computing approach leads to the development of autonomic software systems [15]. These systems have self-adaptability and self-decision-making support systems to manage themselves in running a huge pool of systems and resources like cloud. While proposing this idea, IBM had laid out specific policies for autonomic computing. They are:

- The system must be aware of its system activities.
- It must provide intelligent response.
- It must optimize resource allocation.
- It must be compatible enough to adjust and reconfigure to varying standards and changes.
- It must protect itself from external threats.

In 2004, ARPA started work on a project titled "Self-Regenerative Systems (SRS)," and its aim was to research and develop military computing systems that could provide services despite damage caused by an attack. In order to achieve this, they proposed four recommendations:

- software with same functionality and different versions
- binary code modification
- Intrusion-tolerant architecture with scalability
- systems to identify and monitor potential attackers within the force.

In 2005, NASA began work on a project titled "Autonomous Nano Technology Swarm (ANTS)." The aim of this project was to develop miniature devices such as "pico-class" spacecraft to do deep space exploration. As the swarm of miniature devices enters the extra-terrestrial boundary, up to 70% of the devices are expected to be lost in the process. With the remaining 30% devices, the exploration has to take place (Fig. 1).

Hence, the devices are organized as colony, where a ruler will give instructions to the remaining devices about the course of action. In addition to the ruler device, a messenger device is planned to take care of the communication between the exploration team and the ground control, such as the round trip delay between the mission control on Earth and the probe device in deep space. NASA planned to

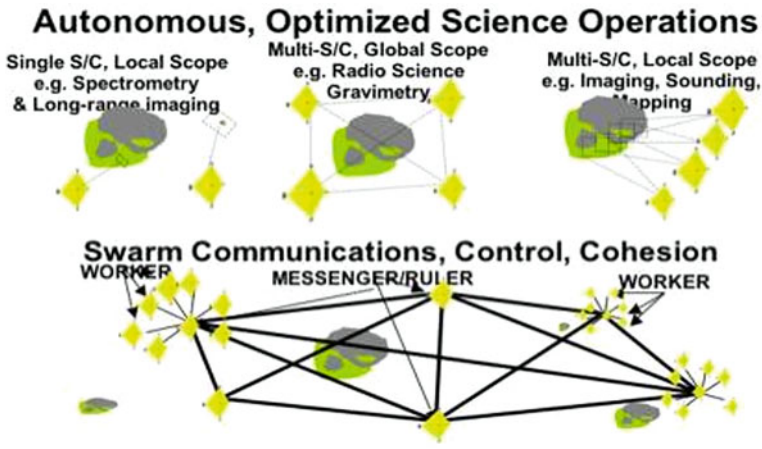

Fig. 1 Autonomous nano swarm

Table 3 Timeline of autonomic computing systems development

Project name	Years	Organization	Features
Situational awareness systems (SA)	1997	ARPA	Self-adaptive network of mobile devices that adjust to varying topology and adjust frequency and bandwidth to varying conditions on the ground.
Dynamic assembly for system adaptability, dependability, and assurance	2000	ARPA	Software systems with ability to probe and gauge for monitoring the system. Adaptation engine to counteract to optimize performance
Autonomic computing (AC)	2001	IBM	Based on human nervous system that employs reflex action. Has four self-management properties: (1) Self-configuring, (2) Self-optimizing, (3) Self-healing, and (4) Self-protecting
Self-regenerative systems (SRS)	2003	ARPA	Self-healing military computing systems that react to attacks or unconditional errors
Autonomous Nano Technology swarm (ANTS)	2007	NASA	The architecture is based on insect colonies. It consists of miniaturized, autonomous, and reconfigurable components for deep space probes

make the devise decide and make decision on its own during critical moments in the probe. Space exploration process strongly needs autonomous systems in order to avert mishaps. These missions are called model-driven autonomic systems [16].

The timeline of the development of autonomic systems is provided in Table 3 [17].

The evolution of autonomic computing is illustrated in Fig. 2 [18]. The evolution started with the function oriented, then object oriented, and component based.

Agent-Based and Autonomic Systems: Automaticity is one of the properties of artificial intelligence. The whole world is witnessing a leap in development and usage of artificial intelligence, and hence, new technologies are emerging in this field [20].

3 IBM's Model for Autonomic Computing

When IBM proposed the concept of autonomic computing in 2001, it had zeroed in on four main self-management properties: (1) self-configuring, (2) self-optimizing, (3) self-healing, and (4) self-protecting. These Self-X properties are inspired by the work of Wooldridge and Jennings (1995) in properties of software agents. They had laid down the following properties for the Self-X management of software systems through software agents [21].

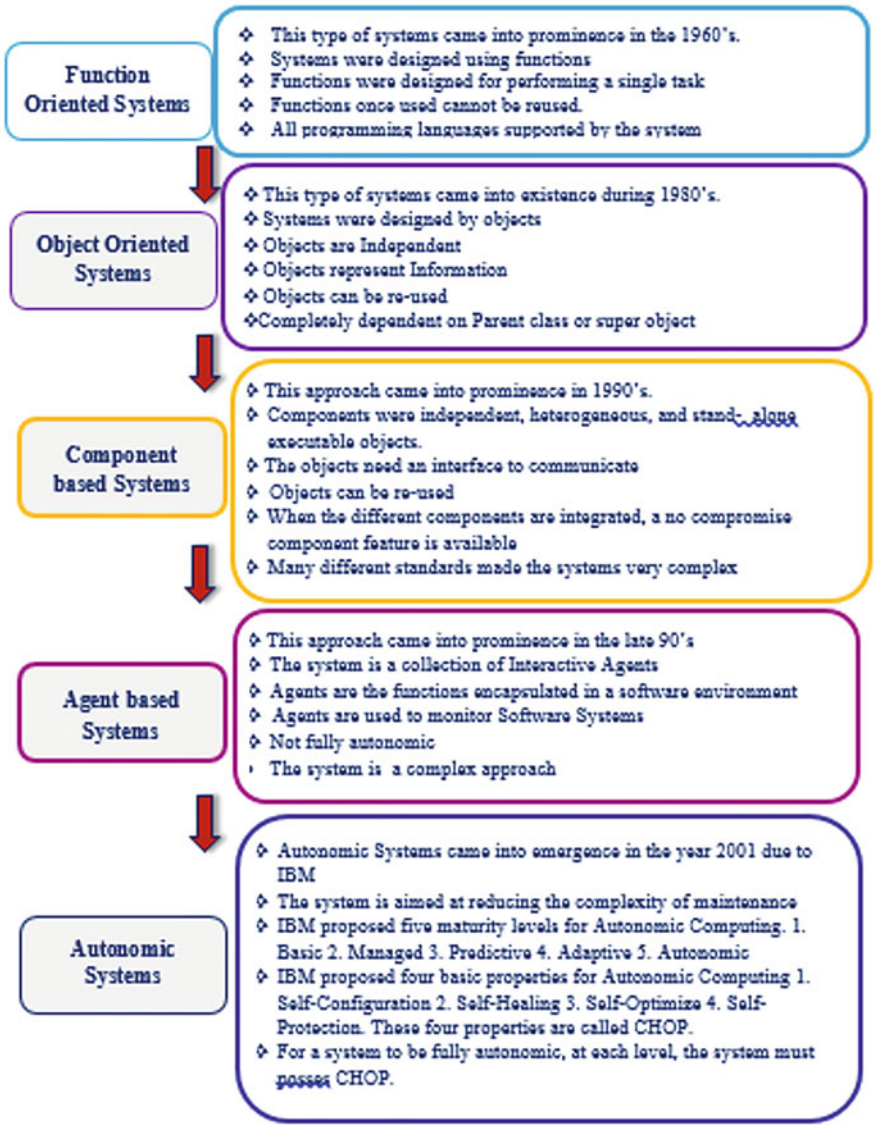

Fig. 2 Evolution of autonomic computing

- Autonomy:

The software agents can work independently without human intervention and have control over what they do and the internal state.

- Social Ability.

  The software agents can communicate with each other and in some cases with humans through a separate language for agents.

- Reactivity.

  The software agents are reactive in nature and can respond to changes immediately whenever they occur.

- Pro-activeness.

  In addition to being reactive, the software agents can also display proactive behavior before any change occurs.

The concept of autonomic computing was described by Horn [6]. The essence of Autonomic Systems was given by Kephart and Chess [22]. It was a novel attempt to correlate the human nervous system which is autonomic in nature to the select attributes in a computer. Further, the authors proposed autonomic elements (AE) and architecture for autonomic systems (AS). Each autonomic element was proposed to have an autonomic manager with capabilities such as Monitor, Analyze, Plan, and Execute (MAPE) and Knowledge database (K), collectively called as MAPE-K Architecture [22].

IBM's self-management properties are based on the Self-X properties discussed earlier. The self-management properties are dealt in detail by Kephart and Chess (2003) and Bantz et al. (2003) [23].

The IBM's self-management properties are

- Self-configuration.

  Ability to self-configure includes ability to install software according to the needs of the user and the goals.

- Self-optimization.

  Ability to self-optimize includes proactive measures to improve QoS and performance by incorporating changes in the system for optimizing resource management.

- Self-healing.

  Ability to self-heal includes ability to identify faults and problems and take corrective measures to fix the error.

- Self-Protection:

  This ability includes preventing attacks from external as well as internal threats. The external threats can be malicious attacks, and internal threats can be erroneous work by a worker.

The advantages of autonomic computing over traditional computing [23] are summarized in Table 4.

Table 4 Advantages of autonomic computing vs. traditional computing

Autonomic feature	Traditional computing	Autonomic computing
Self-configuration	Installing, configuring, and integrating systems is error-prone and time-consuming	Automated configuration of components and systems follows high-level policies. Rest of the system adjusts automatically and seamlessly
Self-healing	Systems have hundreds of manually set nonlinear tuning parameters and their number increases with each release	Components and systems continuously seek opportunities to improve their own performance and efficiency
Self-optimization	Problem determination in large, complex systems can take a team of programmers weeks	System automatically detects, diagnoses, and repairs localized software and hardware problems
Self-protection	Detection and recovery from attacks and cascading failures is manual	System automatically defends against malicious attacks; it uses early warning and prevents system side failures

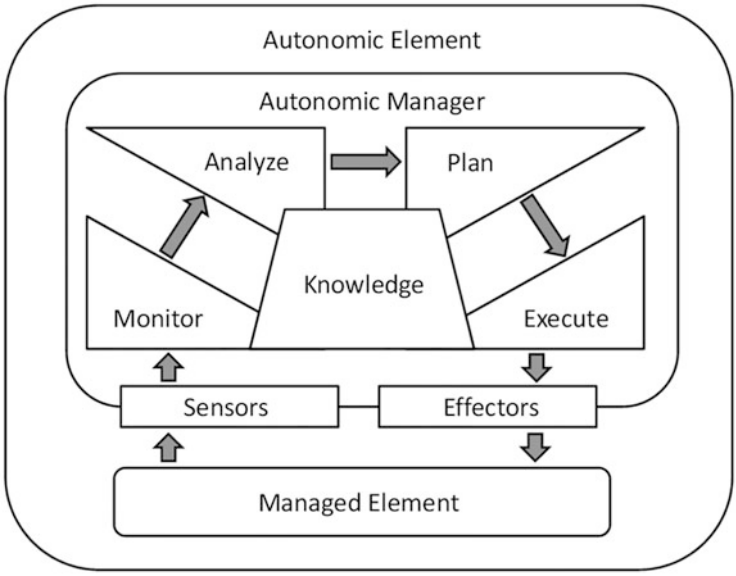

Fig. 3 IBM's MAPE-K architecture reference model

The IBM's self-management properties are incorporated in the IBM's MAPE-K Architecture Reference Model as shown in Fig. 3 [24]. The MAPE-K reference architecture derives its inspiration from the work "Generic Agent Model by Russel and Norvig [2003]." In this model, they had proposed an intelligent software agent that collects data through the sensors. From the data, it infers knowledge and use the knowledge to determine the actions, whichever is necessary.

The key components of the IBM's MAPE-K architecture reference model are [24].

- Autonomic manager
- Managed element
- Sensors
- Effectors
- The managed element

3.1 The Autonomic Manager

It is a software component that collects data through the sensors and can perform to monitor the managed element and, whenever necessary, to execute changes through the effectors. The actions of the autonomic manager are driven by the goals already configured by the administrator. The goals are set in form of event–condition–action (ECA) policy. For example, the goal could be of the form, "When a particular event occurs with a specific condition, then execute a particular action." In such a scenario, conflicts may arise between policies. The autonomic manager applies the knowledge inferred from the internal rules to achieve the goals. Utility functions are also handy when it comes to attain a desired state during a conflict [24]. Further, Nhane et al. proposed the incorporation of an innovative idea like swarm technology in Autonomic Computing. In this technique, the Autonomic Manager is referred to as Bees Autonomic Manager (BAM). Its role is to follow Bee's Algorithm and identify and assign different roles and manage the resource allocation. Further, an exclusive language for autonomic system was proposed by the authors.

3.2 Managed Element

It is a resource that can be a software or hardware that can perform autonomic functions. This autonomic behavior is presented whenever the resource is coupled with autonomic manager.

3.3 Sensors

It is a device that senses the managed element and collects data about it. Sensors or otherwise called as Gauges or Probes. Sensors are used to monitor the resources.

3.4 Effectors

Effectors carry out changes to resources or in the resource configuration in response to a situation in computing. There are two types of changes effected, namely, coarse-grained effect where resources are added or removed and thin-grained effect where changes are made to the resource configuration.

4 Challenges in Autonomic Computing

The evolution of autonomic computing has brought some challenges as well. The development of fully autonomic systems is far from over. Such a development needs the challenges to be identified and remedial measures are found out. The challenges include both the coarse-level challenges and fine grain–level challenges [25].

One of the important issues to be addressed is security in autonomic systems. Smith et al. [4] proposed anomaly detection framework to improve the security. Wu et al. [6] proposed an intrusion detection model to improve security in autonomic systems. Nazir et al. [26] proposed an architecture to improve security in Supervisory Control and Data Acquisition (SCADA). Golchay et al. [27] proposed a gateway mechanism between the Cloud and the IoT to improve security. The challenges pertaining to Autonomic Computing comprises of challenges in architecture, challenges in concepts, challenges in the Middle-wares used, and challenges in implementation. IBM has come up with two factors for evaluating the level of autonomicity of fully autonomic systems. They are functionality and recovery-oriented measurement (ROM). Functionality is used to measure the level of dependence on external factors such as human involvement. The recovery-oriented measurement is used to measure the level of availability, scalability, and maintenance. These factors are yet to be applied and measured in any of the autonomic computing systems. At present, factors such as functionality and recovery-oriented measurement are like hypothesis. For an ideal scenario, a fully autonomic system can fulfill the conditions laid out by the two factors (Fig. 4).

5 Applications of Autonomic Systems

Autonomic computing systems find applications in many fields such as smart industry, where large-scale manufacturing is automated, transportation systems where autonomous vehicles are designed and run, healthcare management, Internet of things (IoT), robotics, and 3D Printing [28].

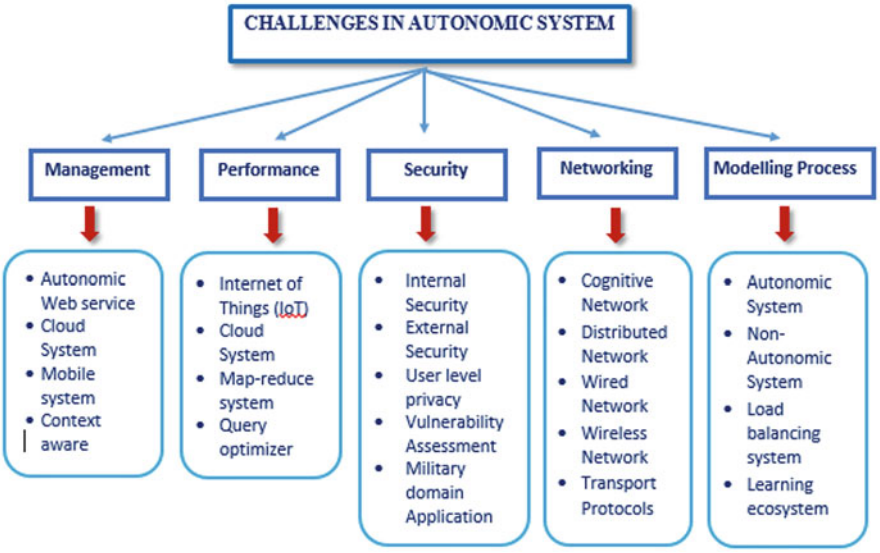

Fig. 4 Challenges in autonomic computing

5.1 Manufacturing Industry

Industry 4.0 offers the scope for emerging technologies to be utilized in production systems. This results in maintenance of quality in the products produced and low cost incurred in producing the product. Also, the process of manufacturing is streamlined to result in flawless production and integration of better engineering practices [6]. Manuel Sanchez et al. [23] proposed an integration framework for autonomic integration of various components in Industry 1.0 as shown in Fig. 5.

The framework is designed to monitor the production process and ensure quality. Internet of Services (IoS) and Internet of Everything (IoE) are incorporated in the framework. IoS enhances the communication between various components such as people, things, data, and services involved in the production.

It is proposed that the business process is the managed resource. And hence business process is the service offered (BPaaS). It is proposed to employ everything mining, and hence, people mining, things mining, data mining, and services mining are included. And it is also proposed to include everything mining in autonomic cycles.

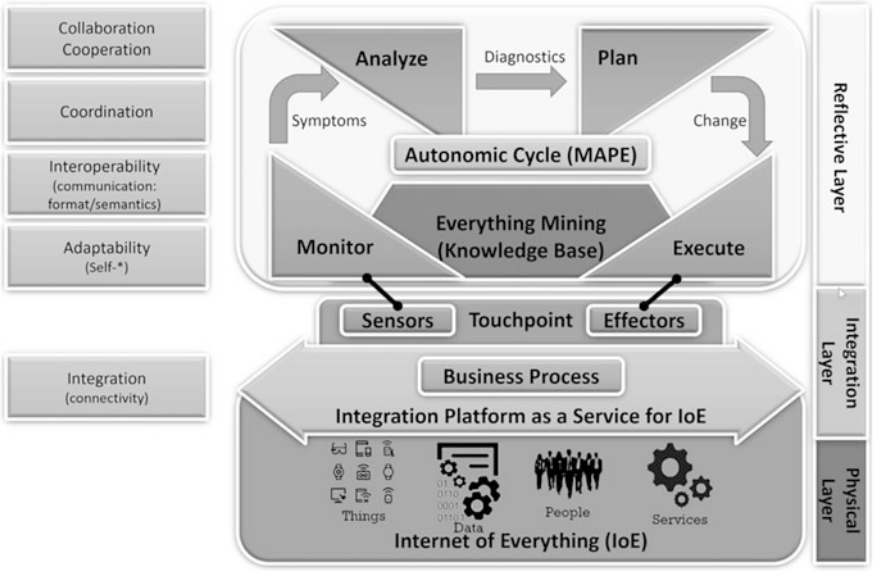

Fig. 5 Integrated framework for autonomic components in industry 1.0

5.2 *Automotive Industry*

One of the main beneficiaries of autonomic computing is the automotive industry. Autonomous vehicles have taken a giant step implementing fully autonomic systems. Every aspect of autonomous vehicle is managed by software [29]. To do so, autonomous vehicles employ machine learning systems, radar sensors, complex algorithms, and latest processors. There are six levels in autonomous vehicles: Fully manual—level 0, single assistance—level 1, partial assistance—level 2, conditional assistance—level 3, high assistance—level 4, and full assistance—level 5. In levels 0, 1, and 2, the human being is responsible for monitoring the driving environment. In levels 3, 4, and 5, the automation system is responsible for monitoring the driving environment. Generally, the vehicles of this class are designed to take orders from the users and hence referred to as automated vehicles rather than autonomous vehicles (Fig. 6).

5.3 *Healthcare Management*

Cloud computing finds application in almost all domains where a large chunk of data are collected and provided using a large pool of resources. Particularly, in remote villages where basic health care is still not available in third-world countries,

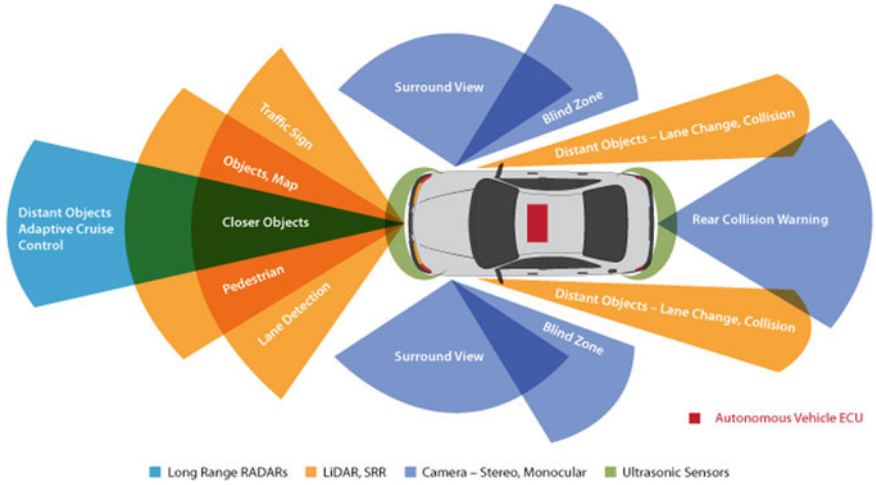

Fig. 6 Automated vehicle

Cloud Computing has provided the opportunity to reach out to the communities and provide better healthcare services [30].

Autonomic systems find application in health care industry in managing complex issues that are otherwise difficult and time-consuming for human beings to operate manually. The services could include pervasive healthcare services. The architecture for autonomic healthcare management is proposed by Ahmet et al. [31]. In this work, the authors have combined the cloud computing and IoT technologies to come up with an autonomic healthcare management architecture.

The efficient healthcare services such as cost-effective and timely critical care could be a reality by incorporating autonomic systems in healthcare management. One bigger bottleneck for developing countries is the population explosion and lack of substantial healthcare professional like Doctors and Technicians. In particular, emergency and trauma care could get the maximum benefits out of autonomic systems in health care (Fig. 7).

5.4 Robotics

With the recent advancements in artificial intelligence and machine learning, robotics field has seen a sea of changes. Robots are now employed in factories performing large-scale operations. Robots are employed in two ways: (1) controlled and (2) autonomous. The controlled robots can perform functions that are instructed or programmed them to do by humans. The successful functioning of a controlled robot includes partial involvement of humans. The autonomous robots can observe the situation and decide to act themselves. No human involvement is need for

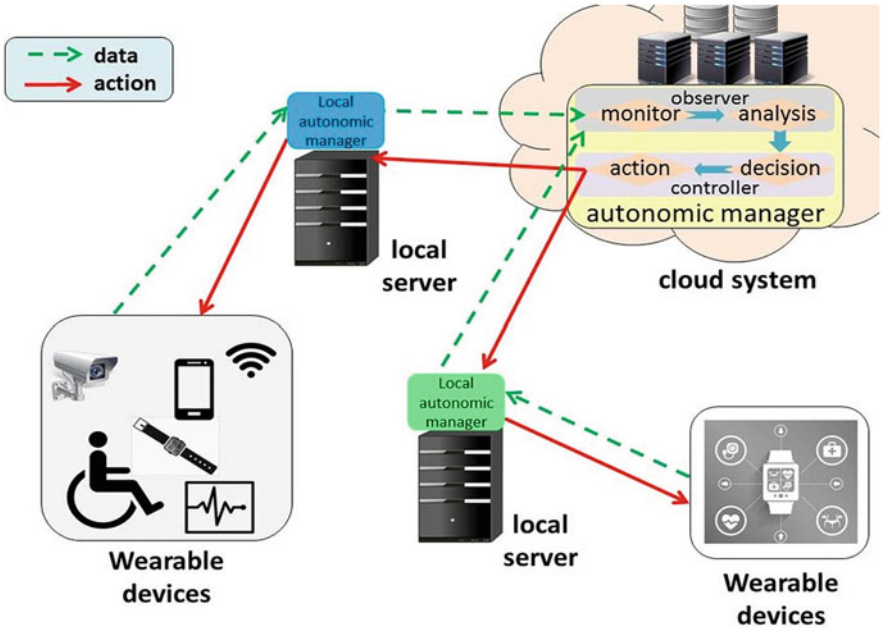

Fig. 7 Autonomic healthcare management architecture

controlled robots [32]. They are intelligent machines. They improve the efficiency and bring down the error. The particular advantage is that they can operate in conditions where human beings cannot like extra-terrestrial explorations [33] (Fig. 8).

6 Conclusion

Autonomic computing has resulted in a new level in IT industry when it comes to automation. Automation has brought down the cost involved and time. Also, the error rate has gone down. These are some of the advantages in the present scenario. In future, as autonomic computing still evolves, it is possible to achieve end-to-end management of services. Further, communication can be made more robust by embedding autonomic capabilities to all the components. The components include Network, Middleware, Storage, Servers, etc. Autonomic systems may be equipped to oversee electricity, transport, traffic control, and in services where the users are more in the future.

Fig. 8 Autonomous robots in space exploration

References

1. Zhu, X., Wang, J., Guo, H., et al. (2016). Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. *IEEE Transactions on Parallel and Distributed Systems, 27*(12), 3501–3517.
2. Shi, W., et al. (2016). An online auction framework for dynamic resource provisioning in cloud computing. *IEEE/ACM Trans Network, 24*(4), 2060–2073.
3. Abeywickrama, D. B., & Ovaska, E. (2016). *A survey of autonomic computing methods in digital service eco-systems*. Amsterdam: Springer.
4. Huebscher, M. C., & Mccann, J. A. (2008). A survey of autonomic computing degrees, models and applications. *ACM Computing Surveys.*
5. Ghobaei-Arani, M., Jabbehdari, S., & Pourmina, M. A. (2017). "An autonomic resource provisioning approach for service-based cloud applications: A hybrid approach. *Future Generation Computing Systems*.
6. Wang, W., Jiang, Y., & Wu, W. (2017). Multiagent-based resource allocation for energy minimization in cloud computing systems. *IEEE Trans Syst Man Cybern Syst, 47*, 205–220.
7. Nzanywayingoma, F. (2018). Efficient resource management techniques in cloud computing environment: A review and discussion. *International Journal of Computers and Applications*.
8. Singh, B. K., Alemu, D. P. S. M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
9. Zheng, Z., Zhou, T. C., Lyu, M. R., King, I., & Cloud, F. T. (2010). A component ranking framework for fault-tolerant cloud applications. Proceedings of the 2010 IEEE 21st international symposium on software reliability engineering (pp. 398–407).
10. Singh, S., Chana, I., & Singh, M. (2017). The journey of QoS-aware autonomic cloud computing. *IT Prof, 19*(2), 42–49.
11. Sobers Smiles David, G., & Anbuselvi, R. (2015). An architecture for cloud computing in higher education. International conference on soft-computing and network security, ICSNS.
12. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. *Data Engineering and Intelligent Computing*.

13. Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *Cloud Computing., 2*(2), 222–235.
14. Singh, A., Juneja, D., & Malhotra, M. (2017). A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. *The Journal of King Saud University Computer and Information Sciences, 29*(1), 19–28.
15. Javadi B, Abawajy J, Buyya R (2012): "Failure-aware resource provisioning for hybrid cloud infrastructure", J Parallel Distributed Computing, 72(10), 1318–1331, (2012).
16. Kaur, P., & Mehta, S. (2017). Resource provisioning and work flow scheduling in clouds using augmented shuffled frog leaping algorithm. *J Parallel Distributed Computing, 101*, 41–50.
17. Yang, S., et al. (2017). Energy-aware provisioning in optical cloud networks. *Computer Networks, 118*, 78–95.
18. Cheraghlou, M. N., Khadem-Zadeh, A., & Haghparast, M. (2016). A survey of fault tolerance architecture in cloud computing. *The Journal of Network and Computer Applications, 61*, 81–92.
19. Dewangan, B. K., Agarwal, A., & Venkatadri, M. (2018). Autonomic cloud resource management. 5th IEEE international conference on parallel, distributed and grid computing (PDGC-2018), *IEEE Digital World.*
20. Kumar, M., & Sharma, A. (2017). An integrated framework for software vulnerability detection, analysis and mitigation: An autonomic system. *Sādhanā, 42*(9), 1481–1493.
21. IBM Corporation. (2005). An architectural blueprint for autonomic computing (3rd ed.).
22. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *IEEE Computing, 36*(1), 41–50.
23. Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration, Elsevier Publications.*
24. Doran, M., Sterritt, R., & Wilkie, G. (2018). Autonomic management for mobile robot battery degradation. *International Journal of Computer and Information Engineering, 12*(5), 273–279. ISNI:0000000091950263.
25. Vieira, K., Koch, F. L., Sobral, J. B. M., Westphall, C. B., & de Souza Leão JL (2019). Autonomic intrusion detection and response using big data. *IEEE Systems*.
26. Nazir, S., & Patel, S. (2017). Pat (2017): Autonomic computing meets SCADA security. 16th international conference on cognitive informatics and cognitive computing, ICCI* CC 2017// (pp. 498–502).
27. Golchay, R., Mouël, F. L., Frénot, S., & Ponge, J. (2011). Towards bridging IOT and cloud services: Proposing smartphones as mobile and autonomic service gateways. *arXiv preprint.*
28. Tahir, M., Ashraf, Q. M., & Dabbagh, M. (2019). IEEE international conference on dependable, autonomic and secure computing. *IEEE Digital Library (2019)* (pp. 646–651).
29. Reduce Automotive Failures with Static Analysis. (2019). www.grammatech.com, Accessed May 20, 2020.
30. Mezghani, E., Expósito, E., & Drira, K. (2017). A model-driven methodology for the design of autonomic and cognitive IOT-based systems: Application to healthcare. *IEEE Transactions on Emerging Topics in Computational Intelligence, Institute of Electrical and Electronics Engineers, 1*(3), 224–234.
31. Ozdemir, A. T., Tunc, C., & Hariri, S. (2017). Autonomic fall detection system. IEEE 2nd international workshops on foundations and applications of self* systems (FAS*W).
32. Puri, G. S., Tiwary, R., & Shukla, S. (2019). A review on cloud computing. *IEEE Computer Society*.
33. Bekey, G. A. (2018). *Autonomous robots from biological inspiration to implementation and control*. Cambridge: The MIT Press.

Autonomic Computing: Models, Applications, and Brokerage

Durga Prasad Sharma, Bhupesh Kumar Singh, Amin Tuni Gure, and Tanupriya Choudhury

1 Introduction

Autonomic computing has emerged as a backbone of major computing and communication paradigms and their application-oriented convergences. These technology paradigms in computing and communication fields share some common features and artifacts. First, in the modern computing environments, pervasive and ubiquitous system models have been getting rapid popularity; nevertheless, they need to be highly reliable, high available, interoperable, and enormously scalable. Second, these systems are required to be autonomic in operations to support the self-controlled and self-managed dynamic discovery of resources worldwide [1].

1.1 Evolution of Autonomic Computing

The early-stage autonomic computing work and autonomic research was the Internet. This was one of the most noteworthy early self-managing initiative by the Defense Advanced Research Projects Agency (DARPA) for the military division application in 1997 and popularly known as the Situational Awareness System (SAS) [2, 55]. In a similar direction, another initiative related to self-management was the Dynamic Assembly for Systems Adaptability, Dependability, and Assurance

D. P. Sharma · B. K. Singh (✉) · A. T. Gure
Computing and Software Engineering, Arba Minch University, Arba Minch, Ethiopia
e-mail: sharma.dp@amu.edu.et; dr.bhupeshkumarsingh@amu.edu.et; amin.tuni@amu.edu.et

T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_4

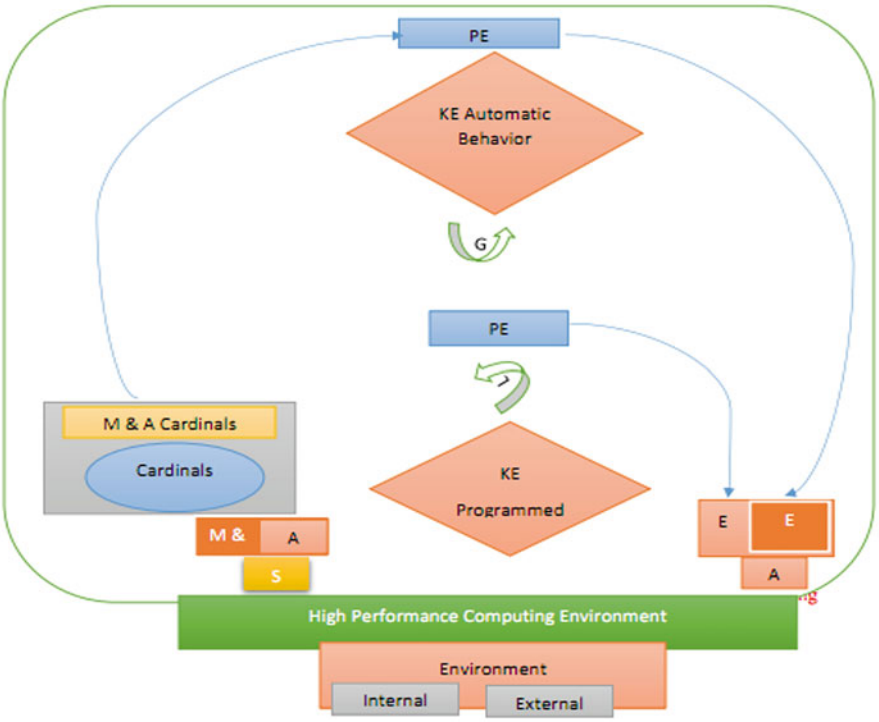

Fig. 1 The conceptual view of autonomic computing

(DASADA), which commenced in 2000. This DASADA program was aimed to design and develop such technology that could enable mission-critical systems to meet high assurance of services but different from IBM's autonomic modernized autonomic computing initiative. Another project initiated by the American agency (NASA) was the Autonomous Nano-Technology Swarm (ANTS) [3].

The broad conceptual view of autonomic computing and communication is depicted in Fig. 1.

Initially, autonomic computing initiative (ACI) as depicted in Fig. 2 was started by International Business Machine Corporations (IBM), USA. The prime goal of the ACI was to encourage technologies that can minimize the man in the middle of the machine and process. Also, it was supposed to be a supporting system to determine and promote the possibilities of automation through well-defined input rules and minimize the human intervention in computing systems. In general, autonomic computing is the ability of a computing system to manage processes themselves automatically through adaptive technologies. The initiative was aimed at reduction of the cost and removal of complexity hurdles in computing technology [4].

Originally, autonomic computing was devised to simulate human body's nervous system. The nervous system of the human body automatically acts and responds

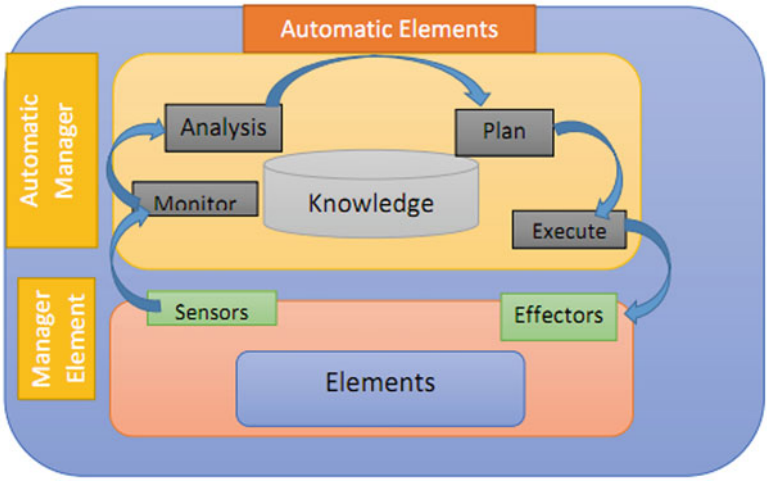

Fig. 2 The IBM's vision of autonomic computing

without the controlling functions (e.g., pulse rate, breath rate, body temperature, and hormone secretion) [5]. Autonomic computing is the anticipated future model of pervasive computing and commutations, where militarized invisible computers will be in the environment and communicating through increasingly interconnected networks leading to the concept of "The Internet of Everything or the Intranet of Everything."

2 Self-Management Properties of the Autonomic Computing System

Autonomic computing is a critical challenge to be alleviated in the cloud [6, 7] or distributed system environments. It needs hardcore improvements in systems modeling, optimization, software architecture, software engineering, human–computer interfaces, and design policy. The prime goal of the autonomic computing and communication system design is to converge these technologies for modeling the appropriate system architecture to achieve self-management capabilities [55].

According to the International Business Machine Corporation (IBM), the four major features of automatic computing are defined as follows [1].

Self-Configuration: This is the ability of the system that how the autonomic computing and communication system configures itself according to the high-level goals of the system and get ready to operate without human intervention. The high-level goals specify what is desired but not necessary to specify how to get it done? This specifies the ability of the system to adapt the system changes, such as self-configuration, self-installation, self-updation, etc.

Self-Healing: This ability of the autonomic computing and communication system specifies how to automatically detect and correct the errors encountered in the system. When the system detects the erroneous process, it takes the corrective action instantly based on self-healing rules. These errors can be low-level bit-errors because of hardware faults or high-level errors in software. High fault tolerance capability is included in the design of the autonomic computing and communication system for a distributed or cloud computing [8, 9] environment. Sometimes high fault tolerance is achieved by the replica and redundancy of the system components which is a vital characteristic of self-healing.

Self-optimization: This feature enables the autonomic computing and communication system to ensure the optimum utilization of the resources. Does this feature also specify how to control the resources automatically for optimal performance/running? The ability to automatic performance optimization is per the need of the customer workloads.

Self-protection: This is the defense capability of the autonomic system which enables the system to protect itself from malicious attacks and also from customers or end-users who accidentally make software changes such as deleting the files. The system can safeguard the customer data against any malicious intruders and unknown threats. This feature enables the system to autonomously self-adjust for maintaining the promised security and privacy. The autonomous system is designed in such a way that it could anticipate the security loopholes and patch them proactively before occurrence.

The other additional features or the properties of the autonomic systems are (1) Self-X—stimulated by the properties of software or hardware agents, (2) Autonomy—Autonomic agents work without human intervention, (3) Social ability—Agents interact and cooperate with other agents like humans using specialized agent-communication language, (4) Reactivity—Agents observe and respond changes or alterations in a timely fashion, and (5) Pro-activeness—The agent does not simply act in response to their environment.

The summarized conceptual differences between the current computing and the autonomic computing and communication environment are depicted in Table 1.

2.1 The Defined Conditions for Autonomic System

The autonomic must be able to-

- Automatically configure and reconfigure itself based on the changes in computing and communication environment.
- Know itself in terms of what resources it has access to, what its capabilities and limitations are, and how and why it is connected to other systems.
- Automatically optimize its performance to ensure the most efficient computing process for the desired workloads.

Table 1 Principal aspects of self-management in autonomic computing environment (Current vs. Autonomic) [1, 10]

Capability	Current computing	Automatic computing
Self-configuration	In the current systems installing, configuring, and integrating systems components is a time-consuming task with an error possibility at data centers	The automated configuration of the system components follows high-level policies. Rest of issues in the system are adjusted automatically and faultlessly
Self-optimization	Current systems have hundreds of manual settings, nonlinear tuning parameters of the system components, and their number increases with each release	In automated system components and systems constantly seek opportunities to improve their own performance and efficiency
Self-healing	Current systems have the problem of self-healing determination in large, complex systems and done by a team of programmers.	The autonomic system automatically detects, diagnoses, and corrects / patch-up the localized software and hardware faults and errors
Self-protection	In the current system's detection and recovery from malicious attacks and cascading failures is manual	The autonomic system automatically defends against malicious attacks or cascading failures. It uses early warning to anticipate and prevent system-wide failures.

- Workaround encountered problems by either self-repairing or routing functions away from the trouble.
- Detect, identify, and protect itself against various types of undesirable attacks to maintain overall security and integrity of the system.
- The capability to adjust and adapt to its environment as its changes interact with neighboring systems and establish communication protocols for better coordination among collaborative subsystems.
- Trust and rely on open standards and cannot exist in a proprietary environment.
- Advanced anticipation of the demand on its resources while keeping transparent to users.

2.2 Fundamental Benefits of Autonomic Computing

The core benefits of autonomic computing are the reduced TCO (Total Cost of Ownership)**.** The breakdowns will be less frequent, thereby drastically reducing maintenance costs. Fewer personnel will be required to manage the systems as most of the functions are managed by the autonomic manager (a system utility) as mentioned in Fig. 3. The current state-of-the-art computing practices have been changing with fast speed. Small- and medium-scale companies and enterprises are unable to afford the fast-obsoleting, high-end, and up-to-date computing and

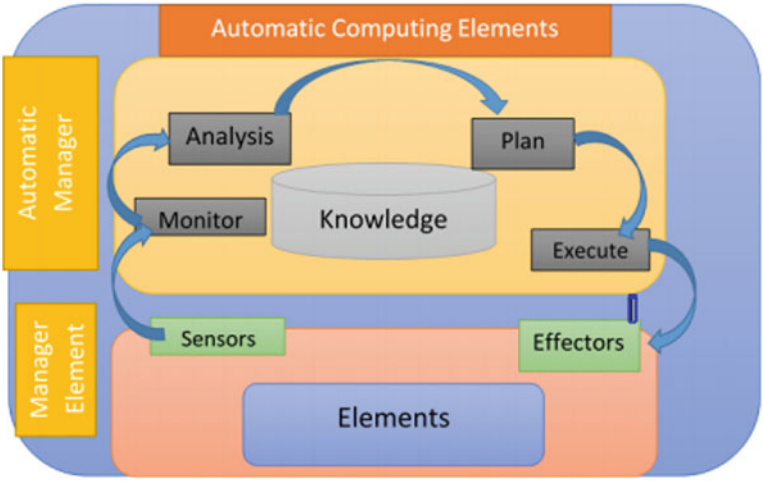

Fig. 3 Functions of autonomic managers and managed resources

communication infrastructure. These companies and enterprises are looking for alternative computing and communication models and applications that can provide such infrastructure services in on-demand manners with real-time scalability, uptime, security, and flexibility. The experts have revealed that the most immediate benefit of autonomic computing will be reduced deployment and maintenance cost, time, and increased stability of IT systems through automation. Table 2 presents a summary of important self-management initiatives in the world chronologically.

Future computing system's philosophy and architectures will be based on the principles "focus on core business and outsource computing and communication needs." The advanced benefits of autonomic computing will include allowing enterprises to better outsource and manage their IT business needs that are easily adaptable based on existing business frameworks and policies with quick provisions, upgrading based on the changing environments and business needs [13]. Another benefit of autonomic computing technology is that it provides easy server consolidation to maximize system services availability and minimize the cost and human interventions to control and coordinate the large server farms for better management and governance. Although the future cloud computing [14, 15] systems will fully exploit the capabilities of autonomic computing and communication systems to create the next generation cloud with ultramodern advancement while computing and communicating with heterogeneous multi-cloud environments. There is a big list of feature-based differences between cloud computing and autonomic computing, but some of the most important differences are described in Table 3.

Table 2 Summary of the important self-management initiatives in the world [11, 12]

1997	DARPA	SAS (situational awareness system)	Decentralized self-adaptive (ad hoc) wireless network Mobile nodes that adapt routine to the changing topology of nodes and adapt communication frequency and bandwidth to the environmental and node topology conditions
2000	DARPA	DSADAA (dynamic assembly for system adaptability, dependability, and assurance	Introduction of gauges and probes in the architecture of software system for monitoring the system. An adaption engine then uses this monitored data to plan and trigger changes in the system. For example, in order to optimize performance or counteract failures of a component
2001	IBM	AC (automatic computing)	Compares self-management to the human autonomic system which autonomously performs unconscious biological tasks. Introduction of the four central self-management properties (self-configuring, self-optimizing, self-healing and, self-protecting)
2003	DARPA	SPS (self-regenerative system)	Self-healing (military) computing systems that react to unintentional errors or attacks
2005	NASA	ANTS (autonomous non-technology swarm)	Architecture consisting of miniaturized, autonomous, reconfigurable components that form the structure for deep space and planetary exploration. Inspired by insect colonies

2.3 Future of Autonomic Computing

Every autonomic computing systems need to have automation, adaptive, and awareness features to provide better services than the existing and ensure long-term survival of the competitive computing systems. Autonomic computing promis**es** to simplify the management of computing systems, especially in distributed, cloud, or other computing environments**.** This capability is the basis for the betterment and effective computing environment over the modern infrastructure such as cloud, fog, or distributed. In the distributed or cloud computing environment, the server

Table 3 Autonomic computing vs. cloud computing [12, 16]

Factors	Cloud computing	Automatic computing
Features	• provides dynamic computing interface • provides a minimal or self-managed platform • provide pay per use bills • provide on-demand access to resources	• provides self-configuration capability • provides self-healing capability • provides self-optimization capability • provides self-protection capability
Geographical location	Computers do not have to be in the same physical location i.e. they can be geographically distributed	Computers do not have to be in the same physical location i.e. they can be geographically distributed
Control & Operating entity	The memory, storage components, and network communications channels are controlled and managed by the operating system of the physical cloud	All computers are controlled by a separate operating system, and error correction and detection are also done by them
Dependency of cloud node	Every node is an autonomic unit and work as an independent entity	Every node is an autonomic unit and work as an independent entity
Network type	Clouds networks MAN, WAN, PAN	Mainly distributed over WAN
Processing capability	Cloud allows numerous applications of different sizes and capacities to run concurrently. Also, use scheduling algorithms for task scheduling	Autonomic system is designed in a manner that their problem is solved according to the high-speed microprocessor by scheduling algorithms
Area of computing	1. Banking & Insurance 2. Weather forecasting & space exploration 3. High-performance computing 4. Service models SaaS, PaaS, IaaS, and XaaS	1. Combined resources tied to business decision-making 2. Combined resources decision-making like cluster services 3. Resources elements managing themselves

load balancing, process allocation, monitoring greenness, automatic updating of applications, software, and system drivers, error detection & correction in memory, automatic backups, advanced—warning about system failure, and the recovery after disasters or failures, etc., are made available. This process of managing the complexity of the systems using several autonomous components can be applied across distributed or cloud applications, networked systems, and services [17].

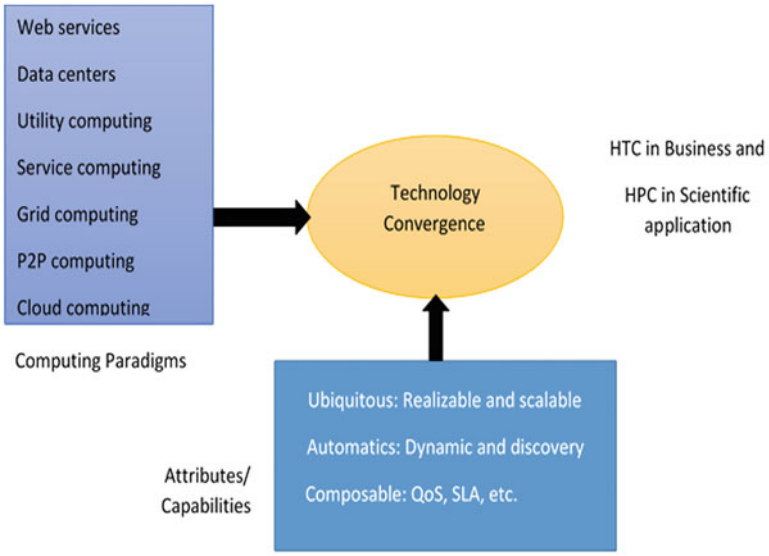

Fig. 4 The computer utilities in modern distributed systems

2.4 Trust, Transparency, and QoS Assurances in Service-Level Agreements

This changing landscape of computing and communication paradigms needs to be aligned with the quality-of-service (QoS) attributes such as energy efficiency/greenness, etc. specifically demanded by the cloud customer and covered in the service-level agreements (SLAs). These energy-efficiency specifications are required to be framed on the basis of trust and transparency at both sides, i.e., client and servers to monitor and measure [18, 19].

In the autonomic computing models, resources are served in two modes—freemium (free for user's usage with limited capacity and features) and premium (paid based on pay-per-use) model. Over the cloud platforms, consumers get the services and resources in self-provision and autonomic discovery an allocation, i.e., with the minimum intervention of human entity. Figure 4 categorizes main computing and communication paradigms in autonomic computing over the cloud and associated technologies with applications.

The convergence can be perceived as a master dynamic enabler of the new technology frontiers and their evolutions. In this essence, cloud computing [20–22] technology is the real convergence of the salient technologies with the two perspectives, business computing and utility computing. The major fields that transform cloud computing into reality are (1) autonomic computing, (2) data center virtualization, (3) utility computing, (4) web 2.0 onward, (5) grid computing, and (6) service-oriented architectures. In conclusion, it can be perceived that autonomic

computing over automated data centers not only boosts the cloud computing in salient dimensions but also creates a new horizon of the technology boom and emerging paradigms.

3 Building Block (Architecture) of Autonomic Computing

Autonomic computing was initiated by IBM and launched in 2001 to construct large-scale distributed computing systems, where human intervention could be minimized or nullified [4]. The prime aim was to alleviate the complexities in computing system configuration, control, and management and tend one step toward self-management system modeling and designs. It was aimed to design such systems that could automatically manage themselves with the help of their self-capabilities. The design specifications and policies are designed in such a way that system's performance can be tested and optimized based on dynamic management. This innovative paradigm of computing and communication outlines a new characterization of unified integration and management of data rather than basic computing. In this technology, customers will be able to access the data, information, and services rapidly. This is done via distributed, cloud, or via other computing resources from geographically dispersed data centers.

3.1 The Basic Architecture of Autonomic Computing

The architecture is the building block or a model that reinforces the plan of work. The autonomic computing architecture describes the autonomic execution of the processes in a heterogeneous computing environment. The basic architecture of autonomic computing consists of two basic components: (1) Autonomic Element and (2) Autonomic Manager. It has a control loop that manages the flow of work between the following subcomponents:

- Autonomic control loop.
- Autonomic manager's role.
- Managed resource and manageability interface.
- The architecture of autonomic element.

To ensure smooth functioning, the autonomic computing system tries to achieve four event-driven tasks: (1) *Collecting* the requirement specifications of the applications from the sensors of the computing environment, making decisions, and, finally, perform the essential adjustments and alterations. It describes two primary components of the systems, autonomic manager and all the managed resources. (2) *Analyzing* the process by applying high-level AI after collecting all the essential

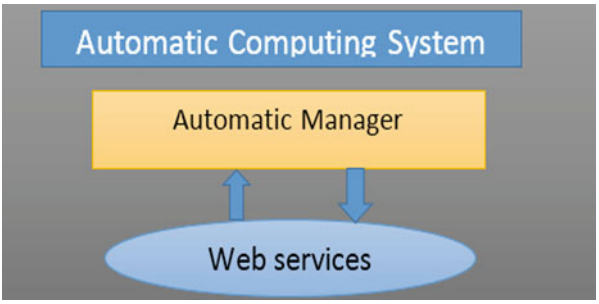

Fig. 5 Autonomic manager

texts. (3) *Deciding* the techniques for the actions that need to be taken to achieve the goals and objectives, (4) *Actions* to execute the plan by making strategies and by managed elements.

3.2 Autonomic Manager

In the autonomic cloud environment, the Autonomic Manager is the key component that is responsible for the self-optimization and self-tuning of the data, information, and knowledge platforms. The autonomic manager as depicted in Fig. 5 is a software element and usually configured manually by the system admins. In the configuration process, the high-level design decision goals are used along with the scrutinized data collected via sensors and the internal knowledge of the system to plan and execute the processes to achieve the goals.

Their prime goal of the autonomic manager is the preservation of appropriate software architecture by making the four parts work together and enhance the functionalities of the autonomic loop. Autonomic manager is also responsible for implementing the control loop. They absorb and create knowledge that is all about the characteristics of the managed resources and is shared continuously among the four parts. Managed resources are the controlled components of the system, which are the core elements of the autonomic computing architecture. The database server, pool of servers, clusters, applications, and routers are the single managed resources. Smart and smooth communication is being established by the autonomic manager to the managed resources. The manageability interface can be separated into two parts (1) sensors and (2) effector operations. The autonomic manager's work is supported by manageability interfaces through transmitting events. The effector manages the modifications or alterations in the data. The four types of interface mechanisms that are executed by sensor and effector operations are (1) sensor healing state, (2) sensor receive notification, (3) effector perform operation, and (4) effector call-out-request. There is a vital need to achieve the several feature-based goals of the autonomic elements used for computing and communication with minimum or zero intervention of the human entity.

4 Autonomic Computing Models

4.1 The MAPE-K Autonomic Loop Model

In 2003, the MAPE-K autonomic loop model was launched by IBM to achieve autonomic computing and communication capabilities**.** The reference model for autonomic control loops is presented in Fig. 6.

Somehow, IBM's MAPE-K autonomic loop model was based on the generic agent model proposed in 2003 by Russel and Norvig [23]. In this model, an intelligent agent observes or monitors its communication environment using sensorsand analyze and apply this observation-based knowledge to determine and plan the actions to execute (implement) in the environment. Actually, the MAPE-K model was designed for smoothening the autonomic communication issues and to provide the architectural design supports to the developments of the communication components in future generation systems [12].

In this model, the managed resources can be any type of autonomic software or autonomic hardware resources. This autonomic characterization in the hardware or software is provided or injected by coupling function with the autonomic manager as mentioned in Fig. 6. In this manner, the managed hardware or software resources can be anything, such as database server, the database itself, web server, web services, containers, query optimizer of the database, operating system, a stack of cloud drives, networks, processors (CPUs), scanners, printers, plotters, and clusters of machines, etc. These system resources are aimed to provide autonomic functions in computing and communication environments over distributed, cloud

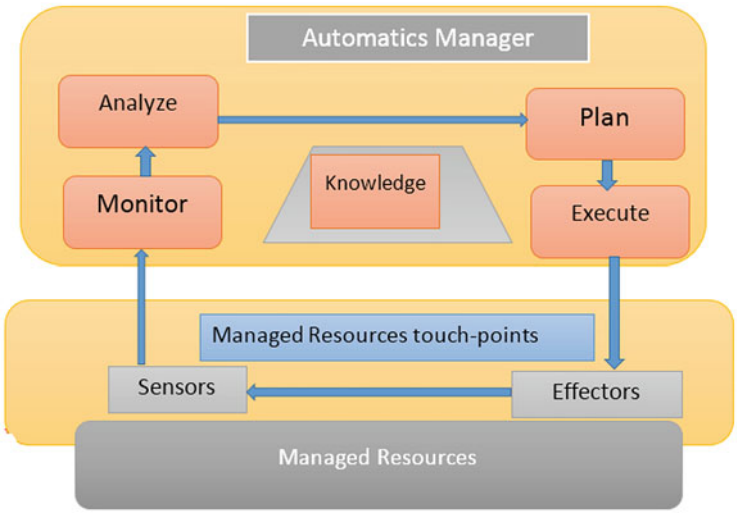

Fig. 6 MAPE-K model of IBM for autonomic looping controls

[24], or other systems. Effectors carry out changes to the managed element. The change in the autonomic environment can be mimicking, for instance, including or excluding (retiring) the server machines to a web server cluster, or in otherwise case thin-grained, for instance altering configuration factors in a web server. Here, the coarse-grained models are computational models that imitate the actions of a complex system by breaking it down into simpler subcomponents. The extent to which the system is broken down reveals the degree of granularity of the models, and the thin grained is just opposite to it.

4.2 The Role of Autonomic Manager and Cloud-TM

The system goals are generally described through event–condition–action (ECA) policies and guidelines [25]. The ECA policies and guidelines consider "when an event occurs and condition holds, then execute an action, for instance, 95% of the web servers' response time in autonomic computing environment exceeds 2 s and if the resources are available, the active web servers are increased." A Cloud-TM is an advanced data-centric middleware technology platform that is responsible for helping in minimizing the operational costs of cloud-based applications, following optimal efficacy using smart autonomic resource provisioning. In the autonomic cloud computing environment, self-optimization is perceived as a ubiquitous or calm characteristic. The Cloud-TM Platform [26–30] leverages on several harmonized and self-tuned tools and techniques that are responsible for automatic optimization based on customer's QoS specifications under the price restrictions toward better servicing. The major factors and functionalities are as follows:

(a) The variety, quality, and quantity of the systems used for the deployment of the data and information platforms in the autonomic cloud environment. This factor is sometimes also referred to as the scale of the platform.
(b) The number of replicas of each data set stored in the platform. This is nothing but the redundancy or replication of data.
(c) The protocols responsible for maintaining the transactional data consistency;
(d) The strategic regulation and the policy frameworks for the data posting and delivery.

The prime goal of this functionality is to exploit the data access locality of Cloud-TM applications. The brief conceptual sketch of the autonomic manager is depicted in Fig. 7 to understand the functionalities.

4.3 MAPE-K Loop Deployment Using Autonomic Toolkit

The International Business Machine Corporation (IBM) has designed and developed a toolkit prototype for the deployment of the MAPE-K loop model. This Toolkit

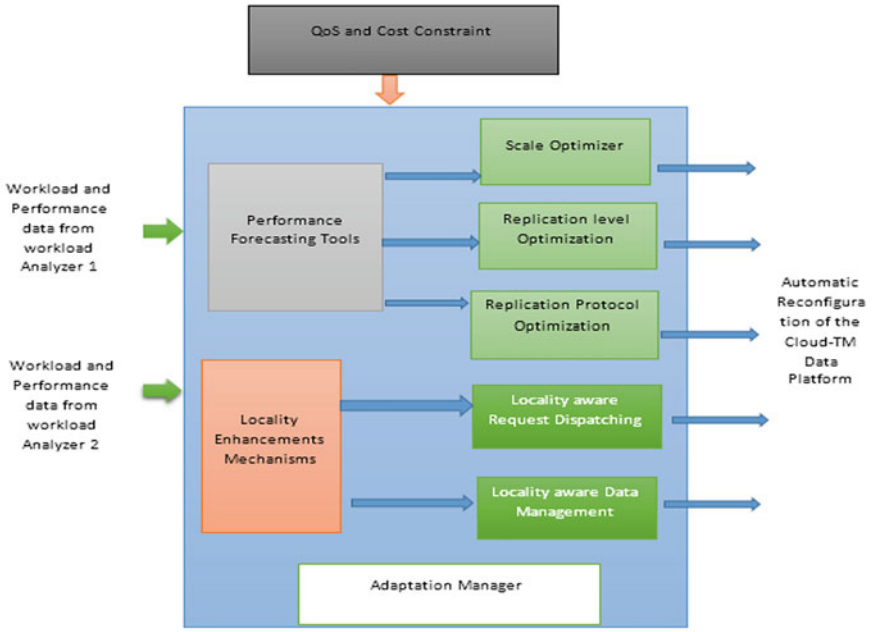

Fig. 7 Conceptual sketch of the autonomic manager's functionalities

is popularly known as the "Autonomic Management Engine (AME)." For the software designers and developers, this Toolkit can be considered as a practical framework and reference deployment for integrating the autonomic capabilities into the distributed or cloud software systems [31]. The toolkit is implemented in Java; however, it can easily communicate with other cross-platform applications using XML message-based communications. This new architecture or model can be applied in the fields, where the autonomic managers can be deployed or implemented at the software application level. For the autonomic functionalities, other familiar toolkits available are ABLE toolkit designed by IBM which is based on multi-agent architecture [32]. The other contributors in the field were Valetto and Kaiser who worked on their own implementation in Java of the complete autonomic loop, called Kinesthetics eXtreme (KX) [33]. This contribution was motivated by the problem of how to include autonomic properties into legacy systems. These legacy systems when designed and developed didn't consider the autonomic artifacts. The efforts are continued to design autonomic middleware models and frameworks that could offer self-management capabilities in distributed and cloud-based applications and still going on.

4.4 Monitoring in MAPE-K Loop

Monitoring involves watchful observations of the computing and communication environment by capturing the salient properties of the physical or the virtual environment over networked computing and communication systems. For instance, the hardware or software components involved in computing and communication monitoring are usually called sensors. For example, the properties that can be monitored are (1) the database indexing and query optimization that affects the response time of a database management system (DBMS), (2) the network latency, and (3) the bandwidth that measures the performance of web servers. As a matter of fact, the MAPE-K (monitor–analyze–plan–execute over a shared Knowledge) feedback loops are the most powerful reference control model for autonomic and self-adaptive computing and communication systems [34].

In the monitoring function, the Autonomic manager collects the monitored data to diagnose the component faults, errors, and the causes of the failure. These data are also used for the performance and monitoring and optimization of the autonomic components. The monitoring activity in autonomic computing and communication systems are of two types-

1. *Passive monitoring*: It can be easily performed using the Linux operating system. For instance, the top command returns information about CPU utilization by each process in the loop. The "vmstat" command returns the memory and CPU utilization statistics. Other operating systems such as Windows version 2000/XP also have analogous passive monitoring tools.
2. *Active monitoring*: It involves the design and development of the software. For instance, for the capturing of the function calls, invocation, or the system calls; modifying and adding code to the implementation of the application or the operating system is popularly ***known*** as active monitoring and usually performed in autonomic modes.

4.5 Planning in MAPE-K Loop

The planning in autonomic computing and communications loops take the monitoring data from the sensors for generating a sequence of modifications that can affect the managed element of autonomic environment. For example, the event-condition-action (ECA) rules are predefined that directly generate the adaptation plans from specific event combinations [35, 36]. The planning of the MAPE-K loop is further divided into the following subactivities:

1. *Policy-based adaptation planning:* The planning and writing adaptation policies and the rules are typical tasks of the system admins, but writing such policy and rules for complex systems is a challenging task. In this activity, the event-condition-action (ECA) rules, which are sometimes also known as strategic

policies, regulate and control the actions to be taken while an event happens and definite conditions have been achieved. Such policy specifications or the rules are planned and articulated by the autonomic system admins [37].

2. *Architectural models:* This planning model emphasizes the Shareability function in autonomic computing and communication environments. In this model, the components represent the integral units of concurrent autonomic computing or communication tasks. However, the connectors symbolize communication among the autonomic system components. In this context, the components could be an integral component of a web application or a complete web-server or a web application on a web-server. But the major drawback of the model is that it does not specify anything about system component configuration, and the connectors. Another drawback of this architectural model is the enforcement of the multidimensional restrictions and properties on the system component and connectors that are determined when the managed elements violate the model rules but require adaptation. This model-based planning in the MAPE-K loop can be explained by some of the most important architectural description languages (ADLs) such as:

   (a) *Darwin:* It is the first and the foremost architectural description languages (ADLs) [38]. This architectural model is a directed graph. This model's approach ensures that every autonomic system component should maintain an identical replica (copy) of the architectural model. In this model, autonomic system component instances are represented by the nodes, and the arcs represent the bindings between the components, i.e., the component requests the service and the component provides the service.
   (b) *Acme/ABLE*: This is actually an architecture but a designer named as a framework. This is a software architecture that is used for monitoring the need detection or assessment for suitable adaptation in a system in which the system components and the system connectors can be annotated with a specific property list along with restrictions for assessing the requirement for adaptation. The Armani, a first-order predicate language, is used in Acme for analyzing the architectural model and detect violations. Afterward, the imperative language is applied to determine policy-based healing strategies [39].
   (c) *C2/xADL:* Another major architectural model was designed by Oreizy et al. and named as C2/xADL [40]. The architectural models' approach is based on a predesigned policy and rules. This model is a converged version of an old architectural model and a new model using the recent monitoring data, and afterward it computed the difference between them to design a patch-up or repair strategic plan. The patch-up or repair plan is then evaluated and analyzed to ensure that the alterations are valid. Now the patch-up plan is executed on the running systems without rebooting or resuming them.

3. *Process-coordination model:* Moreover, the adaptation strategic plan can be achieved by defining the coordination of salient processes executed in the managed elements of autonomic computing and communication systems. For

instance, let us take an example of Little-JIL which is a process coordination language for strategically planning the tasks and coordinating the components that will execute specific subtasks of the original plan. In this manner, Little-JIL can be used for designing or modeling the tasks that are performed by the managed system elements and the numerous other components that can take care of every subtask, resulting in a tree-based pictorial representation of the tasks and subtasks. In this manner, it can be applied for modeling the adaptation strategic plans for the managed autonomic environment [41].

4.6 Knowledge in MAPE-K Loop

The separation between strategic planning and the knowledge that affects the adaptation process in the autonomic computing and communication MAPE-K loop is relatively fuzzy. The diversity of the accumulated knowledge in the autonomic computing and communication systems is just because of the diversity in the input sources like the human expert acts in static policy-based systems. The major methods that are used to represent knowledge in autonomic computing and communication systems are (1) Concept of Utility, (2) Reinforcement learning, and (3) Bayesian Techniques [42–44].

5 Applications of Autonomic Computing

First time International Business Machine Corporation (IBM) introduced the concept of autonomic computing. The prime aim was to the efficient and effective management of the complexity in the globally distributed IT systems. Autonomic computing emerged popular in a very short span, as it alleviated several complexity issues in heterogeneous systems in a cloud computing environment. The concept of autonomic computing revolutionized the computing and communication systems and their design artifacts towards self-manageable systems. In distributed or cloud computing environments, autonomic computing systems are capable of providing the highly scalable, on-demand available, high speed, low cost, and low maintenance service with minimum or without human intervention. There are numerous applications where autonomic computing can be explored extensively. Some of the applications of autonomic computing are as follows:

- Self-healing computing systems and communication service management.
- Autonomic computing in traffic and transportation system management.
- Autonomic computing in virtualized environment management.
- Autonomic computing in self-driving vehicle and aircrafts.
- Autonomic computing in business and governance applications management.

5.1 Self-Healing Computing Systems and Communication Service Management

The concept of autonomic computing and its applications is borrowed from the human body's nervous system. The prime goal and key application of autonomic computing models are is to provide the ability of self-healing and management systems. Self-healing capability of a system enhances the reliability and stability of the system services through constant monitoring and testing of system components. This capability of the cloud or distributed computing system environment focuses on the detection and correction of erroneous faults and reconfigures the system functions without human intervention. The robustness of the system can be ensured by troubleshooting the faults and errors astronomically to complete the tasks without abrupt failure even when the faults persist for a long time. In the cloud computing SLAs, the CSPs ensure high availability of the systems, and this promise is fulfilled by the three mechanisms: (1) replication of the system components, (2) high fault-tolerant design, and (3) self-healing and self-management of the systems so that the customer satisfaction and QoS can be ensured as per the specifications of the SLAs. When the faults or errors are diagnosed by the system, the next task is to plan a corrective action to repair the faults or the erroneous situations for troubleshooting, healing, and resuming the process cycle [18].

5.2 Autonomic Computing in Traffic and Transportation System Management

The load and veracity of the transport systems are increasingly unpredictable with complexities. The transport system needs technology applications to support and enhance the system control and managed efficiently and effectively. The prime convention for applications of autonomic system models in the transport system is to control and manage the high and semi-risky complex operations using self-configurable optimized systems. In general, several autonomic applications have already been introduced in smart cities for parking, taxation, traffic control, accident prediction, crime monitoring, explosive detection, etc. Using the autonomic systems, the traffic movements can be optimized in urban and semi-urban areas to solve the road traffic control and management problems through automatic parking, automatic taxation, self-driven traffic control, auto smart accident prediction, auto crime monitoring through surveillance systems, automatic explosive detection, etc.

The elementary configuration of the transport control system models contains the salient sensors to implement the monitoring and control system solutions using system components and devices. Today such systems can be facilitated by fog-based autonomic computing systems models in smart cities and smart campuses. The procedure focuses on manual traffic managers (traffic operator police) who control and manage the traffic according to their manual understanding. These

system processes are aimed to be included in the autonomic process optimization. The autonomic process optimization can also include the various types of other additional information such as temperature, weather conditions, and the traffic routes using Geographical Positioning (GPS) systems that cannot be estimated by the drivers accurately and smartly.

The transport behavior and the traffic control system follow the concept of bilevel formalism. These systems integrate the operations of the transportation systems. It can manage the automatic traffic lightning system with time complex firmware for the signals. It can delineate the maximal and minimal values separately from the additional traffic characteristics. The network capacity and the nominal levels in congestion conditions can be handled by such a system efficiently and effectively with minimum or without much human intervention in the network. The autonomic computing systems models are capable of minimizing the event of overload and roll back the network links.

5.3 Autonomic Computing in Self-Driving Vehicle and Aircrafts

Around 52 companies have taken official permission to test autonomous vehicles on the roads of the American State of California, and similar initiatives have started in Japan and China [45]. These self-driving vehicles supported by autonomic systems signify a rapid development of autonomic vehicles that will be a fashionable paradigm in the near future. The automobile companies are trying to compete for hardcore dominance in the field of emerging transportation technologies. The arrival of driverless vehicles is usually portrayed as both labor-saving and accident-reducing. The society is changing at fast pace and speed. More and more people are becoming techno-savvy. This societal change will accelerate the transport robots which will undoubtedly be more all-embracing than a simple transformation of the journey between the immediate origin and the destination. These vehicles will be the largest application of autonomic software and hardware systems and their utilities.

Almost all the fighter jets and aircrafts have built-in autonomic components. The civil aviation traffic control systems, airplane autopilot modes, pressure and vacuum measurement, radar functions, GPS positioning, and component failure notifications are the few most important feature examples of autonomic computing and communication applications in the civil aviation industry. The autonomics can be attained by applying multilevel optimization techniques for the control process and functionalities. The autonomic system components in traffic control systems can be demonstrated in a real or simulated environment. The heavy risk industry is today migrating its systems from manual modes to autonomic robotic modes. This increase in the space of the optimal solution in the traffic system corresponds to the requirement for the autonomic behavior of the control system.

5.4 Autonomic Computing in Virtualized Environment Management

System complexity, especially in distributed, fog, or cloud computing environments, is becoming increasingly complicated. The information technology (IT) system needs better algorithms and mechanisms to handle these complexities smoothly. A paradigm shift from physical system access to virtualized system access has revolutionized the computing landscapes in the global IT industry. In autonomic computing, virtualization is the technique where resources are pooled locally or remotely and accessed through an autonomic manageable system with self-provision, control, and management. With the use of virtual autonomic computing models, the data can be easily and safely migrated from one virtual machine location to another in the same or remotely located far away from each other without human intervention.

This process can be done astronomically while both the virtual machines (VMs) are either OFF or ON, i.e., running VMs can also be migrated. The autonomic system models are designed in such a manner that they carry out migration activities while the operating systems run without any interruption and thus minimize the service downtimes to fulfill the SLA specifications [46].

5.5 Autonomic Computing in Business Applications Management

The business drive technology and technology-driven business models have already gained high popularity index and consistently trying to re-engineer the existing business pattern practices and models. The autonomic computing models are differently required to be redesigned or remodeled to suit the current business structure and the variability in the high level of competitive advantages. The modern automated enterprise resource planning (ERP) and customer relation management (CRM) are now being remodeled using autonomic management process concepts. These complex automated models are capable of self-optimize and self-manage business operations. These autonomic processes are managed by well-designed robust algorithms. The algorithms control and manage the system faults, errors, and failure astronomically such as budgetary decisions on allocation, re-allocation of resources, metering, monitoring, pricing policy, schemes, billing, and the performance maintenance from the predicted behaviors. In these system models, the data are collected through several modules, and the automated inventories are done automatically like salary, taxation, and penalty using tax, stock, employee parameters are taken as inputs automatically and the automation process models perform their tasks accordingly. Modern business applications are rapidly transforming from traditional systems to technology-enabled systems. Autonomic computing offers numerous prospects for redesigning the independent intelligent self-managing components to serve

and support the most feasible E-business and E-commerce service models. The emerging E-commerce, E-banking, and E-business models and framework can be explored to upgrade the existing system models and frameworks to generate next-generation business systems by integrating the autonomic features. Autonomic computing for enterprise resource planning (ERP) systems today embrace various system modules that take care of planning, tracking, executing, and controlling the many resources and activities in an enterprise. The autonomic systems have become so complex to manage and therefore the expert and costly engineers are required to establish the ERP or CRM systems, run and maintain. Generally, the ERP systems are as complex, as they are integrated collection of several modules t need hardcore technical knowledge. These ERP system modules could be purchasing, production planning, inventory, and finance, human resource, or CRM [47].

5.6 Autonomic Computing in E-Governance Applications

Today, ubiquitous systems are being introduced in smart cities, smart governance, and smart business and trade systems [30, 48]. Autonomic requests and responses can be designed for citizen-centric services like taxation dues, reminders, loan sanctions, passport issuance, applications, compliance, reminders, and employment application. The autonomic features can be integrated with existing systems and services to create a pervasive environment, where customers or citizens need not search for the services. In such autonomic computing and communication pervasive environment for governance services, the system needs to be intelligent, i.e., the content will be intelligent and will be pushed or pulled to the client or citizen based on his or her access behavior or the smart privileges or rights. The design artifacts of such next-generation smart systems for smart city, smart campus, and smart governance are required to be equipped with autonomic features. Such systems will be able to self-manageable, self-healing, and self-protected from sudden disasters or security breaches.

6 Autonomic Model for Green Cloud Service Broker

Environmental issues are getting abundant attention from computing and communication to IT business and governance around the world. The emission of greenhouse gases, climate change, and sustainability will continue to grow; and the importance of energy-efficient computing and communication in salient domains and IT business will remain argumentative for improving the energy efficiency of their autonomic computing and communication systems and the environmental adversities. The energy-efficient or green computing initiatives are now moving from traditional computing to cloud computing environments. The cost-effectiveness,

high scalability, high uptime, and greenness of autonomic systems and services will be major the focus of futuristic computing and communication systems [49, 50].

Abundant research studies on autonomic computing have investigated and proved that cloud computing is not inherently and always provided energy-efficient system solutions and services. Hence, the autonomic computing and communication products and services provided over clouds are also required to be assessed for energy efficiency. These astronomically provisioned products and services over the cloud need to be declared and certified the level of energy efficiency to minimize the emission of greenhouse gases (GHGs) and adverse impact on the environment [51].

Numerous institutions, organizations, and countries recognized the benefits and efficiency of using cloud computing. Many of them have been adopting energy-efficient cloud computing services to ensure efficient use of power and making the ICT sectors environmentally sustainable. The study [52] highlights the importance of the role played in reducing carbon emissions by developing countries. A crucial part of any global strategy is the role of developing countries like Ethiopia. Research [53] by a leading technology, consulting, and outsourcing company dedicated to environmental sustainability issues shows that moving business applications to the cloud can reduce the carbon footprint of organizations. The study found that, for large deployments, cloud solutions can reduce carbon emissions by more than 30%. The benefits are even more for small businesses; energy use and carbon footprint can be reduced by more than 90%. The large-scale virtualized data centers are now encouraged to establish for meeting this requirement. A study [51] presented energy efficiency challenges while meeting SLAs for autonomic cloud resources. In addition, research of [54] proposed a client-oriented Green Cloud Middleware to assist managers in better management and configuring their overall access to cloud services in the most energy-efficient way. This case study analyzed the existing state of energy efficiency of ICTs usage in higher educational institutions. The primary data were collected using a survey questionnaire to investigate and understand the veracity of the energy efficiency and verified with the facts collected through technical observation. Finally, a model for green cloud broker to support the automatic decision-making processes in selecting or adapting the green cloud products and services in educational institutions. The energy-saving technique called dynamic voltage and frequency scaling technique (DVFS) is also implemented for reducing the energy consumption levels of the data centers using CloudSim. The final contribution is an "Energy-Efficient Autonomic Cloud Service Broker Model (DPS- EEACSB Model)" that was developed for autonomic decision support in selecting the energy-efficient cloud services.

6.1 Simulated Experiment and Analysis Using CloudSim

The power consumption and power-saving overcloud computing environment is considered in the cloud deployment model. By performing experiments in a

controlled environment, the organization can identify performance bottlenecks, pretest expected outcome of implementation using different scenarios, and develop the most viable implementation technique for green clouds. CloudSim toolkit is selected for the simulation tool for this study but there was another option that was considered for this research was Green Cloud which sits on top of NS2 and uses C++ for programming would have been the perfect tool for this research as it measures the details of consumed energy in the components of a cloud data center.

6.2 The Experimental Setup for Power-Aware Technique (DVFS Enable)

In this experimental setup (depicted in Table 4), initially, ten tests were conducted with different scenarios of DVFS technique and then ten tests without power-aware technique (depicted in Table 5). The implementation outputs of different simulated cases were recorded of a simulated cloud data center using the CloudSim toolkit.

Table 4 Experimental values of power-aware technique (DVFS enabled)

	Run 1	Run 2	Run 3	Run 4	Mean value
Experiment 1	0.252	0.251	0.244	0.252	0.25
Experiment 2	0.504	0.493	0.502	0.502	0.5
Experiment 3	1.013	1.023	1.014	0.987	1.01
Experiment 4	2.122	2.102	2.092	2.082	2.09
Experiment 5	4.204	4.184	4.177	4.212	4.19
Experiment 6	8.444	8.414	8.435	8.394	8.42
Experiment 7	16.943	16.903	16.932	16.941	16.92
Experiment 8	33.861	33.833	33.911	33.921	33.88
Experiment 9	67.682	67.675	67.683	67.682	67.68
Experiment 10	101.461	101.458	101.467	101.467	101.46

Table 5 Experimental values of non-power-aware technique

	Run 1	Run 2	Run 3	Run 4	Mean value
Experiment 1	0.871	0.87	0.869	0.869	0.86
Experiment 2	1.77	1.779	1.78	1.781	1.77
Experiment 3	3.58	3.579	3.577	3.58	3.57
Experiment 4	7.281	7.294	7.262	7.299	7.284
Experiment 5	14.582	14.499	14.582	14.521	14.54
Experiment 6	29.711	29.721	29.699	29.71	29.71
Experiment 7	59.692	59.69	59.661	59.643	59.67
Experiment 8	119.435	119.621	119.435	119.421	119.47
Experiment 9	242.291	242.298	242.299	242.308	242.29
Experiment 10	364.192	364.193	364.19	364.189	364.19

The experiments were done for recording the variations in energy consumption metrics for different scenarios like energy-saving techniques versus without energy-saving techniques. The experiments were executed based on the value settings of VMs, Hosts, and Cloudlets. An initial value of 10 Hosts, 20 VMs, and 20 Cloudlets were considered for the experiment. The randomness of the toolkit was set implicitly for each test case which runs four times to produce the mean value of the energy consumption presented metrics using an excel sheet.

6.3 Results Analysis

6.3.1 Experimental Results Using Power-Aware Technique

The prime goal of this experiment was to evaluate how an increase in the number of resources and services in a cloud data center reflects energy consumption. It was also analyzed how the deployment of a DVFS technique can contribute toward reducing energy consumption compared to a data center where a DVFS technique is not installed. The values were set to the initial default value as stated earlier and doubled, recorded, and then analyzed for variations in energy consumption. The experiment based on energy consumption value with the DVFS technique is shown in Fig. 8 The minimum and the default value for this experimental setup were set to 10 Hosts, 20 VMs, and 20 Cloudlets.

Further, an increase in the number of hosts, cloudless, and virtual machines clearly indicates the increase in the power consumption in the cloud data centers. From experiment 1 up to experiment 5, it was recorded that doubling the resources results in doubled energy consumption. Further, after experiment 6, the energy consumption increased (i.e., more than double). This implied that normal and proportional increment in energy consumption deviates from the normal increase pattern to the abnormal increase pattern. This situation concludes that incremental growth in resources and services can increase energy consumption and leads to higher Co2 emission.

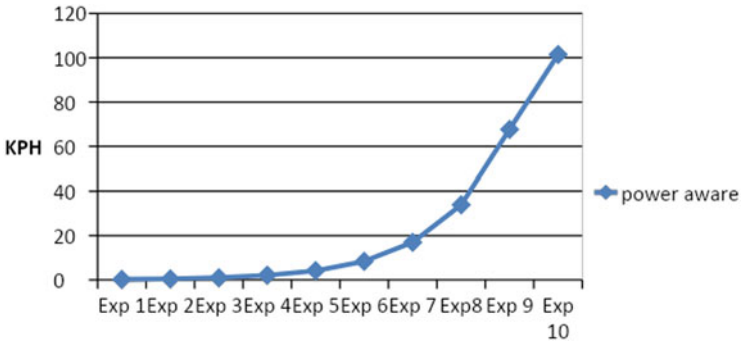

Fig. 8 Energy consumption value with DVFS-enabled technique

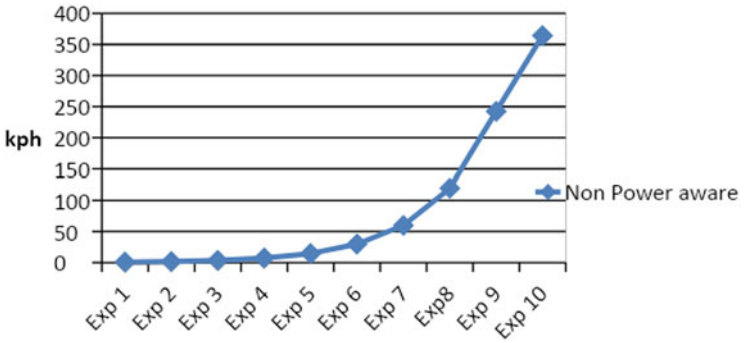

Fig. 9 Energy consumption value of non-power-aware technique

6.3.2 Experimental Results in Non-Power-Aware Technique

The prime goal of this experiment was to investigate how energy consumption increases in the cloud data center that has not deployed any power-aware technique for energy reduction compared to the cloud data center that uses the DVFS technique for energy reduction. The experimental results clearly revealed that increase in the number of resources and services in a cloud data center reflects on energy consumption. The energy consumption values were set to the initial default value as stated in the power-aware experimental setup and then doubled, recorded, and analyzed for variations in energy consumption like the experiments done on DVFS enabled setup. The experiments based on energy consumption value in non-power-aware technique are shown in Fig. 9. Again, the minimum and the default values for this experimental setup were set to 10 Hosts, 20 VMs, and 20 Cloudlets.

The simulation results revealed that the data center without power reduction technique can have a greater energy consumption rate and the simulated experiments have proved it. The energy consumption rate of a data center without a power-aware technique has a significant increase in energy consumption, i.e., almost double from experiment 1 up to experiment 5 and more than a double after experiment 6. The difference between the two scenarios can be seen based on the calculation of the mean values of the energy consumption metrics. In experiment 10, the energy consumption values of power-aware (DVFS enabled) and non-power aware are 101.46 and 364.19, respectively. This shows that the difference in power consumption in non-power the aware cloud data center is 300% higher than the power-aware data centers (DVFS enabled).

Figure 10 presents the comparative analysis of power-aware and non-power-aware simulation results. The results indicate that these techniques can be appropriately applied in analyzing the demand for cloud-based computing products and services. The autonomic decision-making tier, i.e., Energy Efficient Cloud Service Broker can use such technology to check the greenness or energy efficiency of cloud data centers. The cloud customers can get online support services in

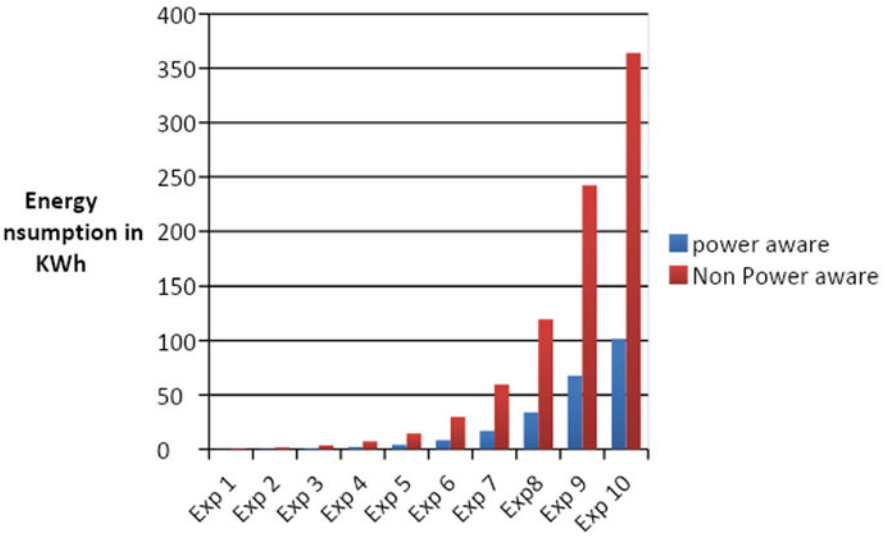

Fig. 10 The simulation result of energy consumption value of non-power-aware versus power-aware (DVFS enabled)

selecting the energy-efficient data center in terms of cloud services, products, and to make ICT usage greener. These experimental results using DVFS technique in a simulated environment can be used as new knowledge support for individual cloud customers, organizations planning for migrating their ICTs over cloud in general, and developing country's educational institutions in specific. This research can be utilized as a great contribution of new knowledge in green ICTs design and deployments towards the reduction of energy consumption, minimizing the CO_2 emission, saving the environment, and its adverse effect on human life.

6.4 Green Cloud Service Broker Model

This study proposes a green cloud broker model. The model consists of four major components: (1) cloud service user/admin/manager, (2) cloud service provider (CSP for public and private cloud, (3) green cloud service broker (GCB), (4) green information directory (GID). Figure 11 proposes an autonomic green cloud service broker model. This model can be used for selecting the green (energy-efficient) cloud services from a set of pooled cloud services over cloud data centers via autonomic green cloud service broker. As illustrated in Fig. 5, the proposed model components are as follows:

- Cloud customer/admin/manager can submit their cloud service requests to green cloud service broker (GCSB) that manages the selection of either most energy-

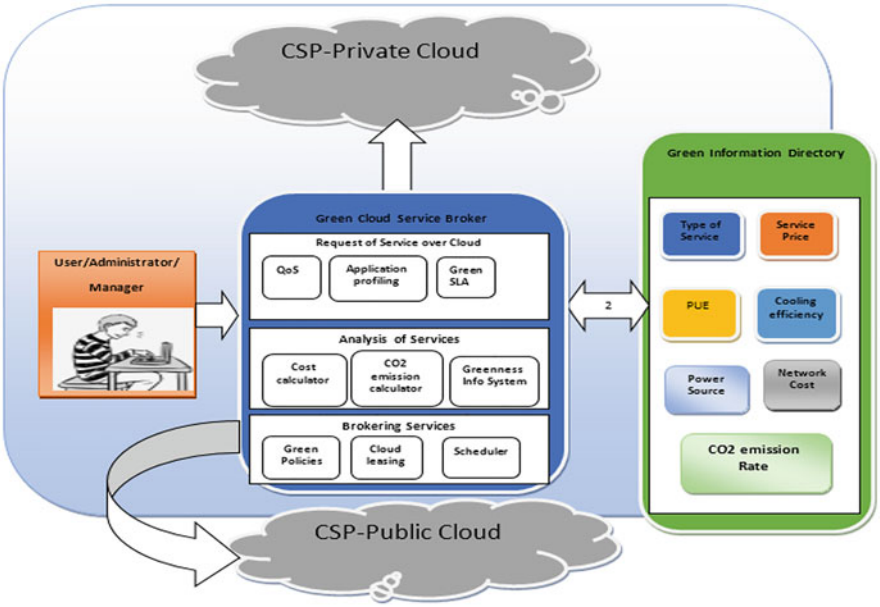

Fig. 11 Proposed autonomic green cloud service broker model (DP-MS GCSB)

efficient (green) cloud services available over CSPs data centers or as per the energy efficiency/greenness specifications provided by the cloud customer.

- GCSP collects the current status of energy the efficiency of pooled cloud services at cloud data centers of CSPs which are registered in the green (energy efficiency) information directory declared by green certification agency, competent authorities, and CSPs themselves.
- Afterward, the GCSB analyzes the energy efficiency specifications and rate of carbon emission of all the cloud service providers registered and offering the requested green cloud services.
- Now the GCSB selects the most feasible services from the pooled services of the CSPs and forward the requests for green SLA negotiations between customers and CSPs.
- Finally, on behalf of the cloud customer, GCSB negotiates and allocates the most feasible cloud services provided by the appropriate CSP.

The whole process of computing, communication, and negotiation is proposed to be done in an autonomic cloud computing environment with minimum or without human intervention. In case, if no exact match is found for the requested service specifications or Green SLA between the cloud customer and CSP failed, the green cloud service broker can select the alternative green CSPs and reiterate the negotiation process.

7 Conclusion

Autonomic computing has evolved as an emerging area of design and development of autonomic hardware, software systems, and the applications that can manage themselves in autonomic manners. These autonomic system capabilities are new thrust for modern computing and communication systems and their multidisciplinary applications. This chapter provides a very brief outline of the modern autonomic computing drive-in alignment with IBM's initiative. The simple features and generic architecture models are covered with a specific implementation of autonomic components. A greenness measurement in an autonomous cloud environment has been discussed with the development of autonomous components for applications. With the autonomic systems and their functionalities, the complexity is at the rise, which has led to more usage of autonomic computing in various technologies. This is somewhat emergent out of the implemented autonomic systems that we can visualize around us in computing and communication environments. There are several autonomic models and frameworks designed and implemented by several scientists and technocrats for the fields of hardware and software automation and adaptive operations. These models are emerging every day using salient design artifacts of functionalities, security, and QoS. This chapter covered a proposed green cloud broker that can be adopted and implemented by either QoS monitoring or regulatory bodies in a cloud computing environment where autonomic features are extensively applicable. The chapter covered the salient types of autonomic computing and communication models and their applications and concluded with a thrust for efficient autonomic cloud broker components between cloud customers and the CSPs. The prominent challenges observed were the performance evaluation and the fault tolerance capabilities that how well an autonomic system is performing and how it can tolerate the multiple to continue its operations, so as to meet the SLA specifications, management policies, and the QoS throughout the service cycle. The robustness is also one of the challenges to be achieved in both autonomic hardware and software components to ensure high-end availability with hardcore fault tolerance capability.

The primary aim of the proposed autonomic green cloud service broker was to investigate and analyze the ICTs usage by using a case study of a selected organization's data center. As a final research contribution; the study proposed an energy-efficient green cloud broker model for selecting the most suitable energy-efficient ICT products and services overcloud. The educational institutions like AMU in developing countries were considered for testing the proposed broker for decision support. After a detailed analysis of the investigated facts, observations, and case-based simulation of conceptual and technical models, an energy-efficient cloud broker model DPS-MS GCSB is proposed for green ICT advisories overcloud. This model supports the Green Cloud Service Brokerage services and advises the educational institutions or individuals for selection and smart adoption of green cloud services. In this model, the energy directory (i.e., important and advisory component of the model) provides information about the CO_2 emission rate, PUE,

the power source used by cloud service providers along with other green offers announced by cloud service providers to the cloud customers. This information is proposed to be declared by the CSPs voluntarily or by national or international green ICTs statutory organizations. The proposed model is a foundation road map for the adoption of green cloud computing services. Thus, the GCSB integrated with third-party outsourced Green Information Directory can be a significant middle-tier decision support system to help cloud customers, organizations, and decision-makers. Further, the international or national green regulatory authorities can standardize the green measures and parameters in ICT-based products and services overcloud. As a conclusion, although autonomic computing has become progressively motivating and widespread technology, still it needs rapid development for robustness and success. Real-world applications, research, and projects will mature this technology. The future of industry 4.0 will definitely provide a great boost in salient disciplines such as robotics, fleet design, and automation. In future the autonomic computing will create new paradigms in distributed, cloud, and fog computing towards re-engineering of their design artifacts with advanced self-decision, self-management, self-healing, adaptive learning, and actions in computing and communication environments.

References

1. Villegas, H. M. N. M. (2017). Architecting software systems for runtime self-adaptation. Science Direct.
2. Tate, A., Levine, J., & Dalton, J. (2000). Using AI planning technology for army small unit operations. In APIS.
3. Azzam, A. R. (2016). *Survey of autonomic computing and experiments on JM autonomic computing and experiments on JMX-based* (pp. 1–93). Berlin: Springer.
4. Jaleel, A., Arshad, S., & Shoaib, M. (2018). A secure, scalable and elastic autonomic computing systems paradigm: Supporting dynamic adaptation of self-* services from an autonomic cloud. *Symmetry, 10*(5), 141.
5. GNU. The nervous system of an animal coordinates the activity of the muscles, monitors the organs, constructs and also stops input from the senses, and initiates actions. Science Daily. 20.
6. Dewangan, B. K., Agarwal, A., Choudhury, T., & Pasricha, A. (2020). Cloud resource optimization system based on time and cost. *International Journal of Mathematical, Engineering and Management Sciences, 5*(4). https://doi.org/10.33889/IJMEMS.2020.5.4.060
7. Wadhwa, M., Goel, A., Choudhury, T., & Mishra, V. P. (2019). Green cloud computing-A greener approach to IT. 2019 international conference on computational intelligence and knowledge economy (ICCIKE) (pp. 760–764).
8. Kaur, A., Raj, G., Yadav, S., & Choudhury, T. (2018). Performance evaluation of AWS and IBM cloud platforms for security mechanism. 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS) (pp. 516–520).
9. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2018). Privacy and Security of Cloud-Based Internet of Things (IoT). Proceedings – 2017 international conference on computational intelligence and networks, CINE 2017. https://doi.org/10.1109/CINE.2017.28.
10. Al-Sharif, Z. A., Jararweh, Y., Al-Dahoud, A., & Alawneh, L. M. (2016). ACCRS: Autonomic based cloud computing resource scaling. *Springer, 20*(9), 2479–2488.

11. Dehraj, P., & Sharma, A. (2019). Autonomic provisioning in software development life cycle process. In Proceedings of international conference on sustainable computing in science, technology and management (SUSCOM), Amity University Rajasthan, 2019, Jaipur - India.
12. Dehraj, P., & Sharma, A. (2020). A review on architecture and models for autonomic software systems. *Journal of Supercomputer (Springer), 80*.
13. Gure, A. T., & Sharma, D. P. (2019). Assessment of knowledge sharing practices in higher learning institutions: A new exploratory framework–AT-DP KSPF. *The IUP Journal of Knowledge Management, 17*(4), 7–20.
14. Bansal, S., Gulati, K., Kumar, P., & Choudhury, T. (2018). An analytical review of PaaS-cloud layer for application design. Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358374.
15. Dinote, A., Sharma, D. P., Gure, A. T., Singh, B. K., & Choudhury, T. (2020). Medication processes automation using unified green computing and communication model. *Journal of Green Engineering, 10*(9).
16. Dai, Y., Xiang, Y., & Zhang, G. (2009). Self-healing and hybrid diagnosis in cloud computing. In IEEE international conference on cloud computing, Berlin.
17. Hill, R., Hirsch, L., Lake, P., & Moshiri, S. (2013). *Guide to cloud computing*. London: Springer.
18. Gebreslassie, Y., & Sharma, D. P. (2019). DPS-Yemane-Shareme CSMM model for client-side SLA of green cloud services measuring and monitoring. *IUP Journal of Computer Sciences, 13*(3), 34–46.
19. Anithakumari, S., & Chandra Sekaran, K. (2014). Autonomic SLA Management in Cloud Computing Services. In SNDS, Berlin.
20. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
21. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. *Data Engineering and Intelligent Computing*.
22. Yadav, A. K., Tomar, R., Kumar, D., Gupta, H. (2012). Security and privacy concerns in cloud computing. *Computer Science and Software Engineering*.
23. Maurer, M., Breskovic, I., Emeakaroha, V. C., & Brandic, I. (2011). *Revealing the MAPE loop for the autonomic management of cloud infrastructures*. Kerkyra: IEEE.
24. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
25. Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—Degrees, models, and applications. *ACM Computing Surveys, 40*(3).
26. D. C. J. C. C. Emmanuel Bernard. (2013). Transactional, object oriented, self-tuning cloud data store. Cloud-TM.
27. JBossDeveloper. (2012). Research projects at JBoss. Cloud-TM.
28. Shekhawat, H. S., & Sharma, D. P. (2012). Hybrid cloud computing in E-governance: Related security risks and solutions. *Research Journal of Information Technology, 4*(1), 1–6.
29. Asres, K., Gure, A. T., & Sharma, D. P. (2019). Automatic surveillance and control system framework-DPS-KA-AT for alleviating disruptions of social media in higher learning institutions. *Journal of Computer and Communications, 8*(1), 1–15.
30. Sharma, D. P., & Gebreslassie, Y. (2019). DPS-Yemane-Shareme CSMM model for client-side SLA of green cloud services measuring and monitoring. *The IUP Journal of Computer Sciences, 13*(3), 34–46.
31. Melcher, B., & Mitchell, B. (2004). Towards an autonomic framework: Self-configuring network services and developing autonomic applications *Intel Technology Journal, 8*(4).
32. Bigus, J. P., Schlosnagle, D. A., Pilgrim, J. R., Mills, W. N., & Diao, Y. (2002). ABLE: A toolkit for building multiagent autonomic systems. *IBM Systems Journal, 41*(3), 350–371.

33. Parekh, J., Kaiser, G., Gross, P., & Valetto, G. (2006). Retrofitting Autonomic Capabilities onto Legacy Systems. *Springer, 9*, 141–159.
34. Arcaini, P., Riccobene, E., & Scandurra, P. (2015). Modeling and analyzing MAPE-K feedback loops for self-adaptation. In SEAMS '15: Proceedings of the 10th international symposium on software engineering for adaptive and self-managing systems.
35. Agrawal, D., Calo, S., Giles, J., Lee, K.-W., & Verma, D. (2005). Policy management for networked systems and applications. In Proceedings of the 9th IFIP/IEEE international symposium on integrated network management.
36. Batra, V. S., Bhattacharya, J., Chauhan, H., Gupta, A., Mohania, M., & Sharma, U. (2002). Policy driven data administration. In Proceedings of the third international workshop on policies for distributed systems and networks.
37. Sloman, M. (1994). Policy driven management for distributed systems. *Journal of Network and System Management, 2*, 333–360.
38. Magee, J., Dulay, N., Eisenbach, S., & Kramer, J. Specifying distributed software architectures. In *In proceedings of the 5th European software engineering conference* (p. 1995). London: Springer.
39. Schmerl, B., & Garlan, D. (2002). Exploiting architectural design knowledge to support selfrepairing systems. In Proceedings of the 14th international conference on software engineering and knowledge engineering.
40. Torii, K., Futatsugi, K., & Kemmerer, R. A. (1998). Architecture-based runtime software evolution. In ICSE '98: Proceedings of the 20th international conference on software engineering, Washington, DC.
41. Wise, A., Cass, A. G., Lerner, B. S., McCall, E. K., Osterweil, L. J., & Sutton, S. M.. (2000). Using little-JIL to coordinate agents in software engineering. In Automated software engineering conference (ASE 2000).
42. Bhola, S., Astley, M., Saccone, R., & Ward, M. (2006). Utility-aware resource Allocation in an event processing system. In Proceedings of 3rd IEEE international conference on autonomic computing (ICAC), Dublin, Ireland.
43. Dowling, J., Cunningham, R., Curran, E., & Cahill, V. (2006). Building autonomic systems using collaborative reinforcement learning. In *Knowledge engineering review journal special issue on autonomic computing*. Cambridge: Cambridge University Press.
44. Whiteson, S., & Stone, P. (2006). Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research, 7*, 877–917.
45. Wakabayashi, D. (2018). *California scraps safety driver rules for self-driving cars*. San Francisco: The News York Times.
46. Zhou, X., & Jiang, C. J. (2014). Autonomic performance and power control on virtualized servers: Survey, practices, and trends. *Journal of Computer Science and Technology, 29*, 631–645.
47. Pop, F., Dobre, C., & Costan, A. (2017). AutoCompBD: Autonomic computing and big data platforms. *Software Computing, 21*, 4497–4499.
48. Muda, J., Tumsa, S., Tuni, A., & Sharma, D. P. (2020). Cloud-enabled E-governance framework for citizen centric services. *Journal of Computer and Communications, 8*(7), 63–78.
49. Rasedur, M., et al. (2019). Hiding confidential file using audio steganography. *International Journal of Computer Applications, 178*(50), 30–35. International Journal of Computer Applications. Web.
50. Khatun, M. et al. Secrecy capacity via cooperative transmitting under Rayleigh and Nakagamim fading channel. Institute of Electrical and Electronics Engineers (IEEE), 2020. 82–85. Web.
51. Buyya, R. A. (2014). Energy efficient management of data center resources for cloud computing: A vision, architectural elements and open challenges.
52. Ambtman, E. (2011). *Thesis: Green IT auditing*. Netherland: Vrije Universiteit.

53. WSP. (2010). Environment and energy, accenture sustainablity. The environmental benefits of moving to cloud.
54. Hulkury, M. N., & Doomun, M. R. (2012). Integrated green cloud computing architecture.
55. R. K. S. A. A. J. Durga Prasad Sharma. (2008). Convergence of intranetware in project management for effective enterprise management. *Journal of Global Information Technology (JGIT)-USA, 4*(2), 65–85.

Issues and Challenges in Autonomic Computing and Resource Management

G. Sobers Smiles David, T. Hemanth, Pethuru Raj, and P. S. Eliahim Jeevaraj

1 Introduction

Autonomic computing gained prominence during the late 2000s. During this period, grid computing was ruling the information technology world. Due to the inherent developments and requirements, slowly cloud computing [1, 2] emerged. As the idea of distributed computing gained acceptance, the idea of automating the systems had to be crossed to overcome the scale. In autonomic systems, the high-level objectives set by the human drive the self-management of the systems. Actually, this is practically difficult to achieve. Adding to the complexity is the heterogeneous nature of the resources such as software, hardware, and middleware. The process of integrating, installing, maintaining, tuning, and configuring cloud components from different vendors is becoming very difficult to achieve [3]. The scale at which the computing demand is increasing is also very fast. It is becoming more and more difficult for the human operator to organize, configure, and operate cloud [4–6] systems. Self-managing systems have emerged as the only viable solution to this problem.

The concept of self-managing systems was first introduced by IBM Research as Autonomic Computing Initiative (ACI) in 2001. A manifesto released by IBM Research for Autonomic Computing pointed out the need as the size of connected

G. Sobers Smiles David (✉) · P. S. Eliahim Jeevaraj
Bishop Heber College, Tiruchirappalli, India
e-mail: eliahimps.cs@bhc.edu.in

T. Hemanth
York University, Toronto, Canada

P. Raj
Reliance Jio Platforms Ltd, Bengaluru, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_5

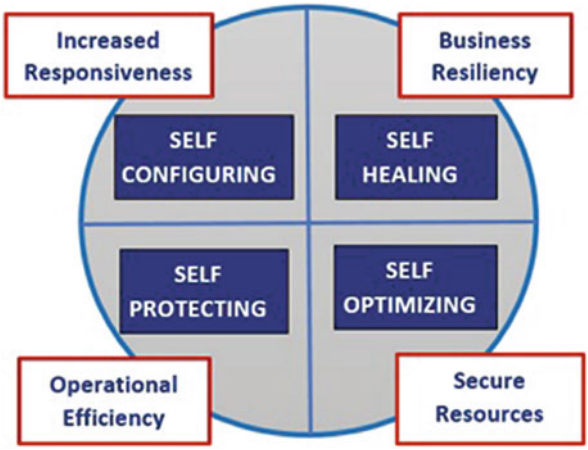

Fig. 1 The autonomic computing attributes

systems with Internet's exponential growth. The process of administering software systems has gone beyond individual software environments. Further, it pointed out the need to integrate heterogeneous environments to achieve scale. The complexities due to scale have surpassed human capabilities. And the need for interconnecting the systems and integrating the heterogeneous components has never stopped growing [7] (Fig. 1).

The Hewlett-Packard Labs too tried its hands on self-management systems and labeled it as Adaptive Enterprise Initiative (AEI) in 2002. Microsoft's version of self-managing systems was introduced as Dynamic Systems Initiative (DSI) in 2003 [8]. All three initiatives were focused on autonomic elements such as computing resources and autonomic systems. The resources include servers, middleware, storage, manager for workload, balancer for load, and broker for resources. The main thrust of the three initiatives is self-management. It includes maintaining and adjusting the operations despite changes in the workload, resource needed, load, software and hardware errors.

The heterogeneous nature of autonomic computing systems warrants effective mechanisms for interoperations between autonomic elements. Also, effective sharing of information between autonomic elements further strengthens autonomic computing. The relationship between low-level system requirements and high-level service-level objectives was studied by HP Labs and IBM Research. They came out with the observation that, "Models, and methods for learning the correlation between requirements and objectives automatically, are key to the functioning of autonomic elements." Models are handy to make templates that can predict future conditions. Also, models are very useful to forge relationship between the high-level priorities and low-level parameters that the autonomic elements have to monitor and manage. One of the major challenges is how to make the models learn and readjust accordingly. Also, the important thing in modeling is that, models must need less data for observing the situation and consume less time for training. After training, the model must act fast on the go. Queuing model is used for allocating the resources

and for optimizing the performance. The learning process for the model has no room for extensive trial and error. The models must capitalize on the old knowledge learned and leverage it with the new knowledge acquired without compromising performance [9].

Maintaining quality in such a scenario is based on the parameters of the quality of service (QoS). The QoS parameters include reliability of service, cost, time, latency, execution time, elasticity, and high availability. When a service is provided, the nature of the service and how it is going to be orchestrated are governed by service-level agreements (SLA) between the service provider and the receiver of the service. When a parameter in SLA is violated, the service provider is penalized. One of the most challenging tasks in administering cloud computing is how resources are managed in a cloud. This significantly assumes greater importance when a legal document such as service-level agreement (SLA) has to be satisfied with the quality of service (QoS) requirements. The problems to consider in SLA-based resource management are actual resource type required, mapping the resource, provisioning it, allocating it, adaptation, and service brokerage. The purpose of SLA is to ensure customer satisfaction, thereby improving the profit and market share. When a customer's requirement changes, the resources management system must ensure that resource reallocation takes place. If resource reallocation is not done for resource requirement changes, then, it has a direct impact on profit maximization. Due to the dynamic and heterogeneous nature of cloud, the resource management system must have the ability to adapt to the changes in the requirement of resources [10].

1.1 Classification of Autonomic Systems

There are two classifications of autonomic computing systems, self-optimization systems and self-healing systems. Cloud computing' service model, in particular, IaaS (infrastructure as a service) model enables system administrators to obtain resources on demand, and release the resource when the required operation is done, more importantly, pay only for the amount of time the resources have been utilized [11]. The autonomic resource provisioning in cloud computing is given in Fig. 2.

This scenario of remote infrastructure is slowly replacing data center of late. As a fallout the system administrators no longer need the specific hardware expertise. Still, the problem of resource provisioning in cloud computing continues to remain a complex task. Cloud computing has brought a revolution in the way; computing and storage devices are acquired and utilized. Organizations are no longer needed to invest in these resources, but can use the services of cloud service providers who rent these resources. The fact that the resources can be obtained for rent from cloud service providers shows that these resources are not part of the local physical infrastructure and hence results in lesser tasks that need specific hardware expertise. But, still the onus is on the system administrators to estimate the required computing

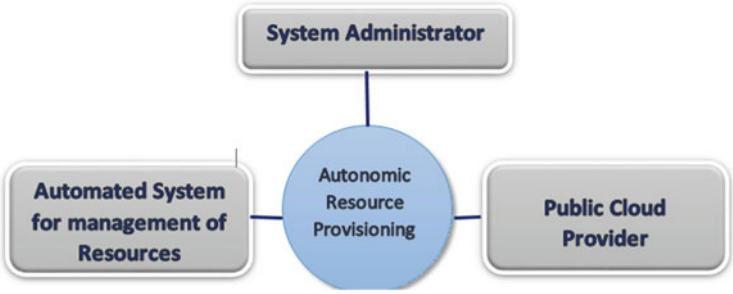

Fig. 2 Autonomic resource provisioning in cloud computing

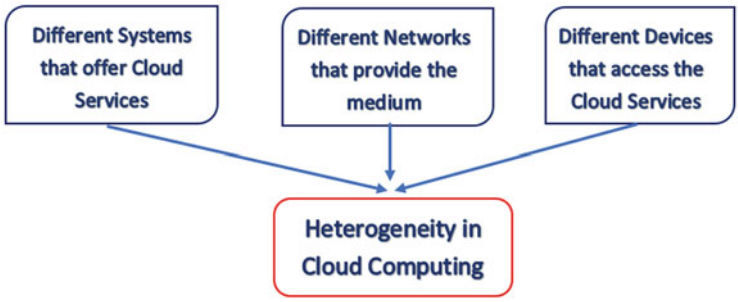

Fig. 3 Factors for heterogeneity in cloud computing

and storage resources, acquire them and maintain them. Autonomic provisioning of resources in cloud computing provides a healthy alternative to this scenario [12].

2 Challenges Due to Heterogeneity in Cloud Computing

The problem of heterogeneity is a fairly complex one as it is prevalent in three domains. (1) Heterogeneity in cloud systems that provide the services. (2) Heterogeneity in networks that provides the medium for providing the services. (3) Heterogeneity in devices that access the services [13]. This scenario is illustrated in Fig. 3 [14].

The cloud system's heterogeneity arises due to different vendors who provide different services through different platforms and APIs. This leads to challenges in interoperability and portability. Also, market competition makes the service providers employ heterogeneous frameworks, which further complicates the system. The heterogeneity in networks arises due to the heterogeneous nature of the technologies and mediums in use. Also, the tremendous rise in the number of users who have access to Internet has made the market very competitive. Mobility has become the mantra of all the service providers. The network coverage has to ensure access

to the network and therefore information to the customer at anytime, anywhere, and anyhow [14]. These variations in networks and their related technologies actually have a bearing on how the services are delivered. Heterogeneity in devices arises due to the different hardware and software in use. Also, the emergence of smartphones and the various devices that use different technologies, hardware, architectures, and differing operating systems.

Differences in cloud resources such as compute, storage, and networking result in degraded performance. The variations caused by different vendors create an additional headache for the customer. The customer has to analyze the various resources available and compare the cost involved and availability. The geographical displacement of resources and data makes the process of data management a challenging one [15]. In such a case, data interoperability and integration devoid of any common standards and platforms. Portability of codes between heterogeneous systems is far more complicated due to the geographical displacement of the systems [16].

3 Challenges in Autonomic Computing

3.1 Computing Challenges Due to COVID-19

International Data Corporation (IDC) in its survey done on Cloud Computing in January 2020 has come out with the following challenges. The survey presented the top with 74.9%, to challenge in security. Followed by performance, availability, integration with in-built IT, capability to customize, charging only on on-demand basis, getting back the in-house facilities, controlling requirements, and obtainability of supplies. In general, there will be greater investment in public clouds for technology-based infrastructure [17].

In the COVID-19 Impact on IT spending Survey, May 2020, done by IDC found that in India, there will be an increase of 64% demand for cloud computing in organizations. And, the remaining 56% cloud software will support the new scenario. As per IDC, there will be high demand for the IT infrastructure such as VPNs, collaboration suites, endpoint encryption, and cloud tools. COVID-19 has made work from home a new norm. Nearly 80% of the organizations in the domains such as online teaching, online entertainment, online business, collaborative work domain, and software development have adopted the work from home model. The work from home method had led to an increase in remote support services that are based on professional skills and cloud software and security of sensitive data. There will be greater demand for SaaS-based collaborative applications with an increased need to work remotely. One of the positive impacts of COVID-19 is the accelerated demand for pay as you use and public cloud models. This momentum will be sustained even after the COVID-19 crisis [18].

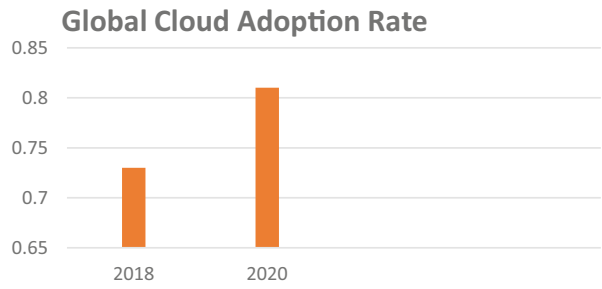

Fig. 4 Global cloud adoption rate

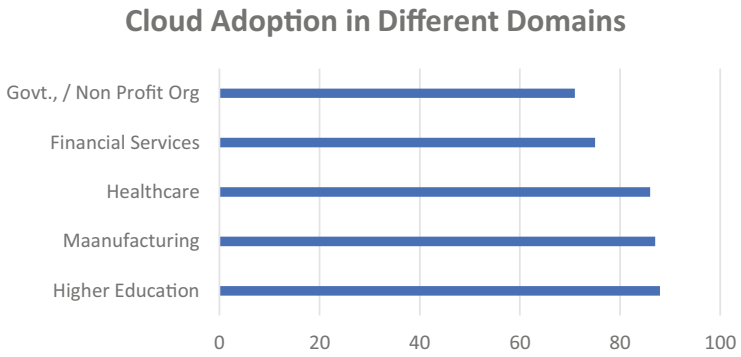

Fig. 5 Global cloud adoption rate by domains

International Data Group Inc., (IDG) is a research organization with its focus on the technology landscape. In IDG Cloud Computing Survey—exploring cloud usage trends, investments, and business drivers—2020, it has come out with the following findings [19].

- In the aftermath of the COVID-19 crisis, cloud computing is here to stay. It has established itself in the information technology environment.
- The survey involved 551 IT decision makers (ITDMs), who are involved in or who have planned to the process of purchasing cloud computing services and infrastructure.
- Cloud adoption growth is on a steady rise. In 2018, the rate of adoption was at 73% and in 2020, the cloud adoption rate has risen to 81% as illustrated in Fig. 4 [17].

The 81% represents people who have already moved into the cloud. The cloud adoption rate is expected to rise another 12% in 12 months. Organizations are already looking to moving to cloud in the wake of unprecedented COVID-19 crisis. Of these, higher education has the highest rate and government and nonprofit organization stand last. This is illustrated in the Fig. 5 [17].

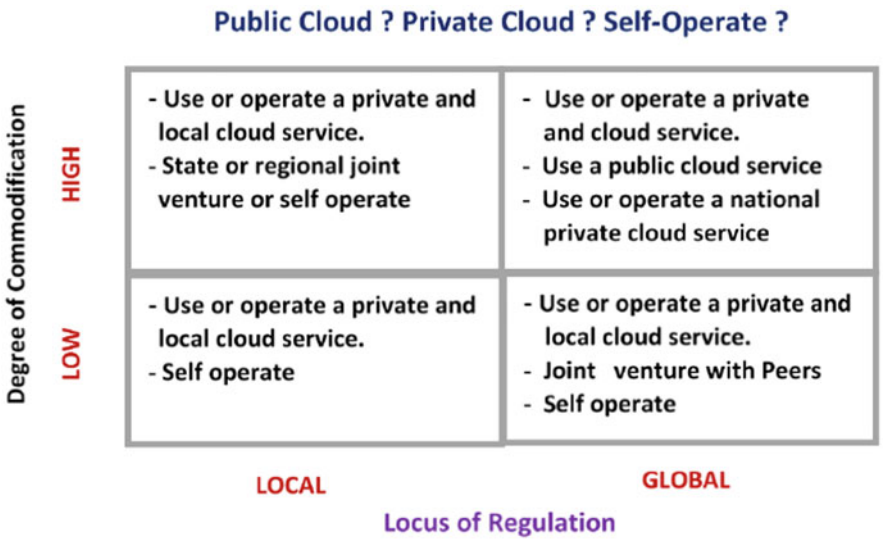

Fig. 6 Strategy for cloud adoption in higher education

3.2 Cloud Adoption Strategy for Higher Educational Institutions

Before COVID-19, higher educational institutions opted for commercial cloud service providers. The providers were able to make a proportionate saving in costs gained by an increased level of users. Looking at the advantages on offer by adopting cloud, the higher educational institutions ignored the cost levied by the commercial providers. Especially for teaching and learning in higher educational institutions, cloud offers tremendous efficiency. Cloud has enabled the higher educational institutions to focus on teaching and learning rather than software and IT configuration, which is illustrated in Fig. 6 [18]. This enabled the institutions to reduce IT expenses, to maintain their own data centers, and improve IT provisioning for researchers and students.

The applications and services migrating to the cloud are expected to witness a phenomenal increase in the next 18 months. This is illustrated in Fig. 7 [19].

In this survey, the respondents have given their opinion on challenges in cloud computing. The results are displayed in the Fig. 8 [20]. Organizations differ in their size and the complexities also differ from organization to organization.

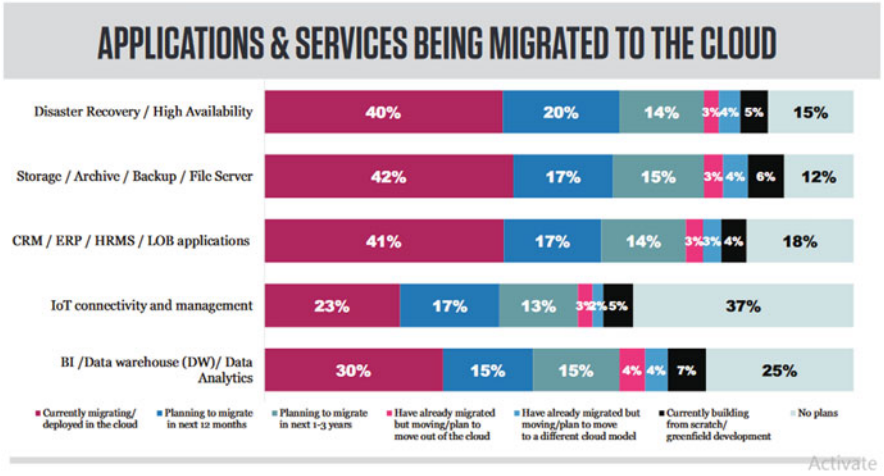

Fig. 7 Apps and services migration to the cloud

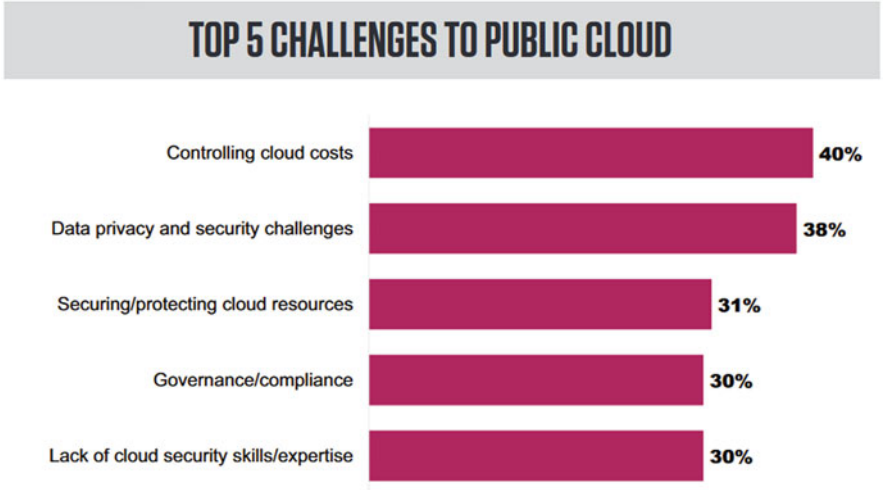

Fig. 8 Challenges in cloud computing

3.3 State-of-the-Art Problem

The tricky issue of cloud resource management is a hard problem. A hard problem is one in which, the need for resources grows exponentially with the input. In cloud, the complexity arises due to the amount of data. The heterogeneity of the resources and the unpredicted nature of the load. The fluctuating workload is a major challenge to providing elasticity in cloud computing. There are two ways in which fluctuations occur: (1) Scheduled spike and (2) unplanned spike. In the planned scenario, the

load fluctuation could be foretold in advance and the resultant resource management can be planned earlier. In unplanned spike, load fluctuation is handled on demand. That is, during the delivery of service, the load fluctuation has to be treated and the corresponding resource management has to be done only in real time [21]. There is a fundamental change in the manner in which policies are enforced in cloud systems compared to the traditional systems. The cloud-specific policies include

- Admission control, where a decision to admit a job for processing in cloud or not is done.
- Resource management, where depending on the request, VMs are provisioned onto the physical machines and jobs onto VMs.
- Quality of service (QoS) is ensured by its metrics. They include response time, operational cost, throughput, maximization of profit, and so on.
- Workload management, where the fluctuations in the load are managed. The efficient balancing of load ensures effective management of resources.
- Energy management, where the optimization of energy at the data center is done (Fig. 9).

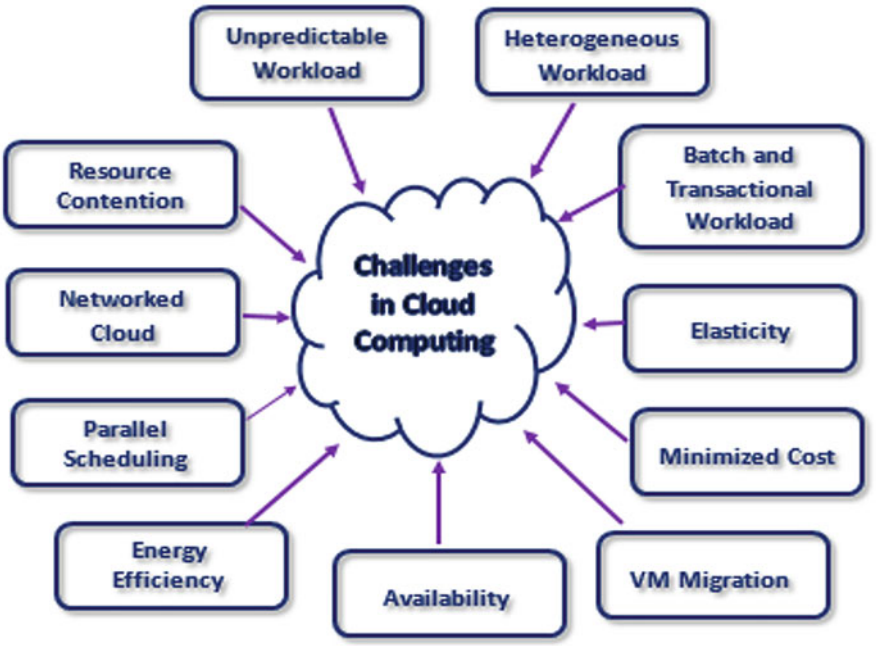

Fig. 9 Research challenges in cloud computing

3.3.1 Role of Virtualization in Resource Provisioning and Allocation

Virtualization is the process of creating a software-based virtual representation of a resource. It is a process of running a virtual instance. Virtual machines help to increase the efficiency with which the resources are utilized. The virtualized setting virtualizes all the components of a network, such as servers, storage, and networks. In addition to the physical infrastructure, a huge collection of logical instances of network termed as virtual networks exist. Hence, the network nodes and its topology are all virtual. The virtual nodes are accommodated by the physical host. The virtual path is established through physical path. The virtual network is logically detached. They are dealt by separate units. The physical host employs the virtual host, which is handled by a software program termed as virtual machine monitor. Its functions are to enable swift responses for requests between the hosts and the virtual machines. The virtual machines have the capability to operate in isolation. By this feature, they can maintain remoteness even with the other virtual machines employed in a particular physical host. They also offer flexibility for customization in their operating manner. Also, virtual machines exhibit mobility. This feature allows the transfer of virtual machines between physical hosts. Also, this transfer does not disturb any current execution in a physical host. The number of virtual machines that can be employed by a physical host is unlimited [22].

From the viewpoint of architecture, virtualization fulfills the subsequent design goals.

- Simultaneous presence of multiple virtual networks in the same setting
- Ability to generate new virtual networks over existing virtual networks
- Elasticity to implement ad hoc network topology and tailored control protocols
- Possibilities to have overall managerial control over a virtual network
- Reasonable segregation from each virtual networks
- Diverseness of physical setup

The use of virtualization has paved the means for the abstraction of computing resources. Through multiple logical VMs (virtual machines), a single physical machine can be employed to operate as multiple machines. Also, the capability of a physical machine to host multiple operating system environments on the same machine that are completely isolated from one another is an unique benefit of VMs. Another significant benefit of abstraction is the ability to configure VMs on the same physical machine to utilize different partitions of resources. For example, on a physical machine, while one VM can be assigned 15% of the processing power, the other can be assigned 30% of the processing power. Consequently, to meet the constantly changing resource requirements of users, VMs can be initiated and terminated dynamically. Specifically, different resource management policies could be enforced for different VMs for diverse user requirements and loads. Service-level agreement-oriented resource allocation is made possible by these features [23].

4 Definitions for SLA

SLA is defined by Dinesh et al. as: "An open declaration of hopes and requirements that occur in a business association between two organizations: the facility provider and consumer." HP Labs define SLA in the web services domain as "a contract used to ensure the distribution of web service. It describes the cohesive relations and prospects from provision provider and provision customer." SLA defined by research project in the networking domain as "an agreement between a network service provider and a customer that stipulates, the services the network service provider could supply. If the service provider could not satisfy the agreement, the fines that are to be paid to the consumer" (p. 19).

Internet NG in the Internet domain defines SLA as "the lawful basis for the distribution of services. The parties involved in managing service distribution and admission are the users of SLA. The means by which the Service consumer and provider sees SLA is different. For the consumer, the SLA is the lawfully binding description of the series to be provided by the service provider. For the service provider, the SLA acts as a definite, agreement of what is to be provided." Sun Microsystems Internet Data Centre Group in the Data Centre Management defines SLA as "an official contract to promise what could be provided and deliver what is promised." To sum it up, SLA defines the ability of a provider to deliver, customer requirement's performance target, the scope of guarantee to availability of resources, and the metrics' mechanisms including measurement and reporting.

4.1 Components of SLA

An extensive description of the SLA components is given in Fig. 10.

The following are the components of SLA approach

- *Objectives*: The objectives to be achieved by using an SLA.
- *Constraints*: In order to ensure the level of services requested, the mandatory steps and actions that are needed to be taken.
- *Duration of validity*: The legal operational time period of SLA.
- *Room for services*: Services that are to be provided to the customers, and services that are not covered in the SLA.
- *Transacting parties*: Anyone providing services or any individual accessing the services and their roles (e.g., provider and customer).
- *Service-level objectives (SLO)*: The agreed levels of services of both the parties, the provider, and the consumer. In some cases, service-level indicators such as performance, reliability, and availability are used.
- *Penalties*: When the service provided could not achieve SLOs or if the performance measurement is below par, penalties are imposed on the service provider.
- *Optional services*: Not so mandatory services which may be required.

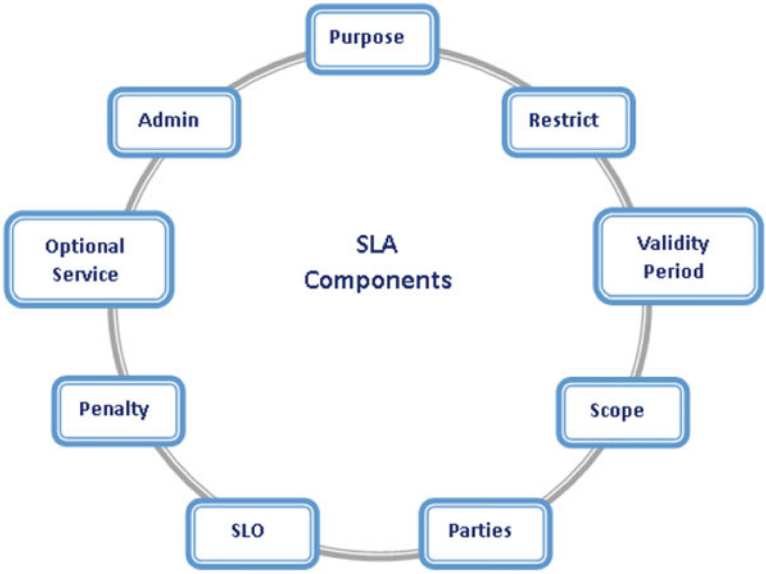

Fig. 10 Components of SLA

- *Administration*: Methods that are used to make sure the SLO achievement and the associated managerial tasks for controlling them.

4.2 Phases in SLA Lifecycle

There are three phases in the SLA lifecycle. The first phase is the creation phase. Here the customers find the appropriate service provider who matches their service requirements. The second phase is the operation phase. During this phase, a consumer has access to the SLA, but it is read only. The third phase is the removal phase, during which the SLA is terminated. All the associated configuration information of the terminated SLA is removed from the service. The three phases of SLA are shown in Fig. 11

The Sun Microsystems Internet Data Center Group [24] has proposed a more detailed SLA life cycle which is given in Fig. 3. There are six steps in this model. The first step is to find the service providers. The appropriate service providers are to be located in accordance with the consumer's needs. The second step is to define the SLA. It contains the services definition, service negotiation entities, policies for fine in case of SLA violation, and QoS parameters. In this step, the parties negotiate to mutually agree on the service requested and the service to be provided. The third step is to launch the agreement. In this step, an SLA prototype is set and to be filled in by specific contract between the entities. The fourth step is to monitor the

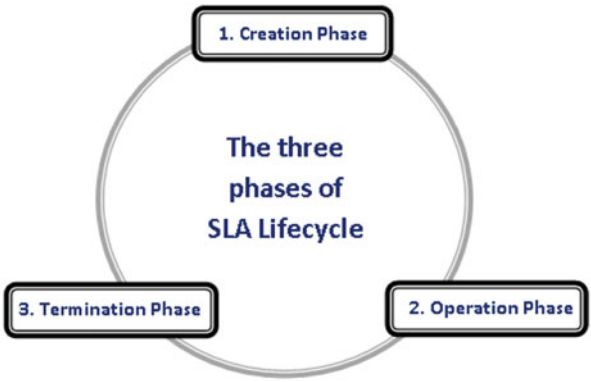

Fig. 11 The phases of SLA lifecycle

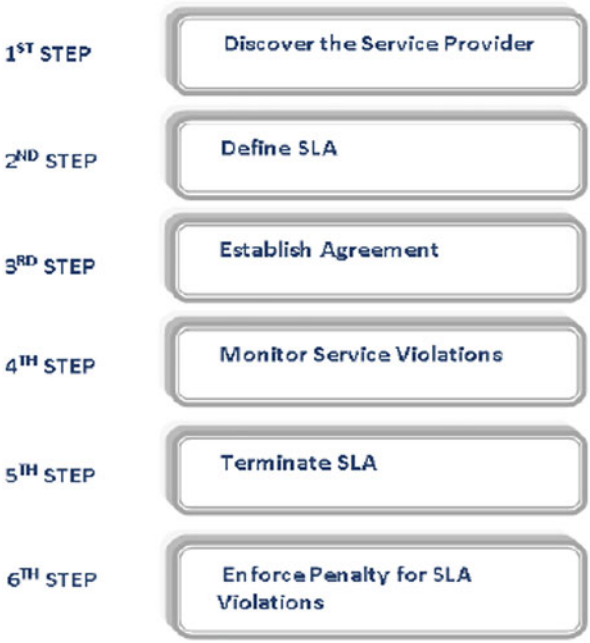

Fig. 12 Internet data center group's phases of SLA lifecycle

violation of SLA. Here, the provider's rendering enactment is calculated with the contractual bindings. The fifth step is to termination of SLA. An SLA is terminated whenever the time duration is completed or any other violation of the contract happens. The sixth step is to enforce penalties for SLA violation. Whenever any service party violates the contractual terms, the corresponding fine conditions are invoked and penalty is imposed. These steps are illustrated in Fig. 12.

4.3 The Need for SLA Approach for Cloud

The traditional resource management model is not capable of processing the task of resource assignment and allocating resources dynamically. As cloud offers the capability to access information anytime, anywhere, and anyhow, it is difficult for a cloud service provider to dynamically allocate resources efficiently. The need of the hour is a customer-centric resource management system that is market-oriented and is capable of meeting the needs of demand. The emerging service market's success is directly dependent on customer satisfaction [25]. The factors involved in service quality directly impact customer satisfaction. So, cloud service providers are under immense pressure to meet the customer's needs. Service-level agreement-based resource management methodologies are needed in order to satisfy consumers, in this case, the students. These proposed methodologies must have the capability to manage the service paradigms such as service request, response, customer feedback, pricing model, and effectively manage the ever-growing customer demand with the limited resources in an autonomic manner.

4.4 Challenges in SLA-Based Resource Provisioning Using Virtualization

The SLA-based resource provisioning is the foremost challenge in differentiating and satisfying the service request of users. Second, ensuring customer satisfaction has become a crucial factor. Customer satisfaction includes provisions for feedback, realizing the specific needs of the customer, security against risks, etc., the service requirement of users constantly changes with time [26]. The challenges in SLA-based resource provisioning in cloud environment are given in Fig. 13.

The autonomic resource provisioning mechanism must perform the following:

- Continuous monitoring of current service requests and self-manage the resources whenever there is change.
- Handling incoming service requests and amending them.
- Adjusting allocation, schedules, and prices in case of amendment.
- Automatic configuring for new requests concurrent existence of multiple virtual networks in the same environment.
- Options to create new virtual networks on top of existing virtual networks.
- Flexibility to execute ad hoc network topology and customized control protocols.
- Options to have complete administrative control over a virtual network.

4.4.1 DPS-Yemane-Shareme CSMM Model for Client-Side SLA

In order to overcome the discrepancies in the server-side SLA and to empower the client D.P. Sharma et al. [27] have proposed the DPS-Yemane-Sharme CSMM

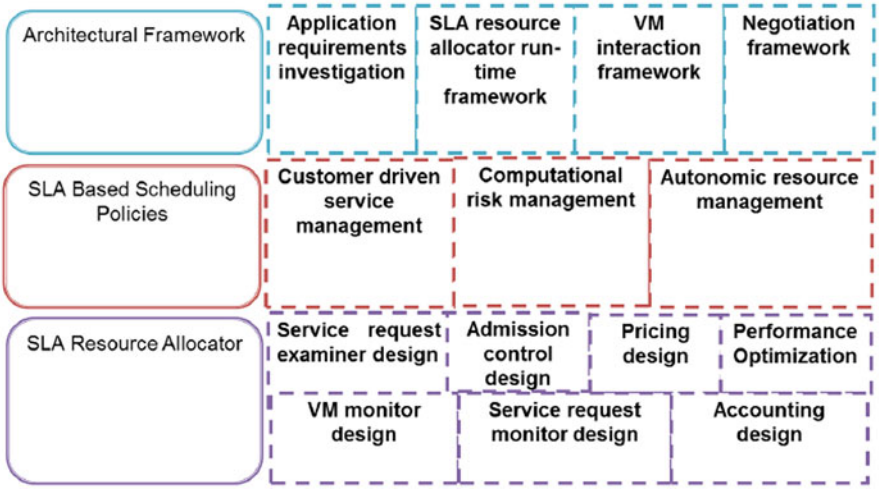

Fig. 13 Challenges in SLA-based resource provisioning

Model for client-side SLA. The issues such as performance, service outage, customer satisfaction toward the performance of the service provided, violation of SLA, and green-based SLA. This is a novel approach for client-side metering and billing system to enable the customers to cross verify the billing and ensures transparency thereby ensuring quality of service (QoS). It comprises components such service broker, green cloud broker, SLA measuring, and monitoring agent. This model empowers the customer to terminate the services in case of dissatisfaction in the service provided. The total IT load and total IT facility load were considered for benchmarking. The results show that the issues such as trust and transparency could be ensured by using this model.

5 Case Study: The Emergence of Micro Service and Containerization

With the ever-growing demand for mobile on-demand service, cloud computing technology is evolving with each day. One of the latest technology to have evolved out of cloud computing is the micro services achieved by using containers.

5.1 What Is a Micro Service?

It is a simple mechanism by which independent applications are combined in a loosely coupled manner to make a system. This architecture helps organizations

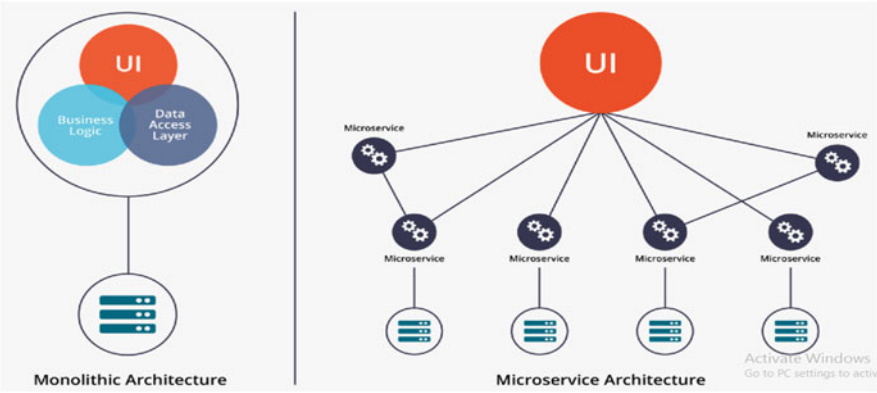

Fig. 14 Difference between monolithic and micro-service architectures

to update their application components independently. On the contrary, it is not possible to update components independently in monolithic architecture (Fig. 14).

In the existing architecture, different components of an application make up a single program. In order to update components, the whole application needs to be built and deployed, resulting in consumption of more time. The micro-service architecture enables faster release of software, frequent updating of software, and smooth and quick addition of new features in the software. In particular, scalability is the main advantage of micro-service architecture. This is achieved by deploying processes in the application as individual services. The individuality of processes makes it possible for the system to scale components individually. This is not possible in monolithic architecture [27].

5.2 The Containerization of Services

A container is an executable package. It is lightweight and supports stand alone. They have the capability to run on virtual infrastructure as well as bare metal. They are very different from virtual machines, which are heavy and dependent. This has paved the way for extensive use of two virtualization layers. (1) Virtual infrastructure layer for VMs. (2) Container virtualization layer. Containers enable creation of multiple instances of a single-threaded micro service [28]. Figure 15 shows the differences between virtual machines and containers.

The container technology has its roots in Linux. With the introduction of containers in micro-service architecture, many issues and challenges encountered in service-oriented architecture (SOA) such as modularity, horizontal scalability, infrastructure agility, and availability have been solved. The use of containers has greatly reduced the cost involved and has increased the Return on Investment (RoI) for businesses. The most popular business to adopt micro services is Walmart in

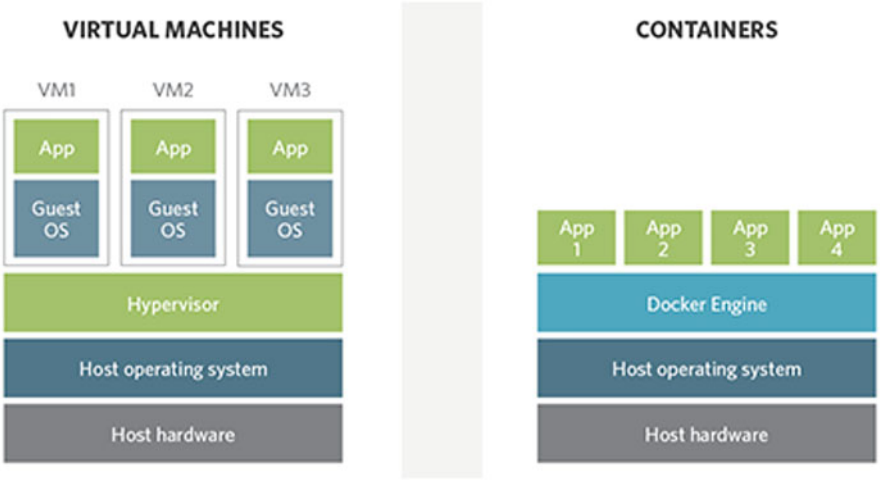

Fig. 15 Difference between virtual machines and containers

Table 1 The advantages and disadvantages of containers

S.no	Advantages	Disadvantages
01.	Container-based virtualization increases the utilization potential of a server	The inability of containers within an application to dynamically share resources based on current demand
02.	The ability to scale Application components independently	A container orchestrator such as Kubernetes is needed to run a container
03.	Containerization of virtual network functions (VNFs)	The higher dependency among data shreds and micro services
04.	Independent upgrade cycles	More points of vulnerability for attacks
05.	Fine-grained resource control	Container management requires resource allocation on a per-container level

2012, Uber, and Netflix. Table 1 illustrates the advantages and disadvantages of the containers [29].

The lack of universal performance model in cloud computing has paved the way for micro-services architecture. Although considerable amount of effort and time is needed for the implementation of micro services, the benefits on offer are huge and businesses can bank on it to improve cost and time [30].

6 Conclusion

The occurrence of COVID-19 has resulted in positive impetus for cloud computing. More and more organizations are looking up to cloud computing for bettering their service delivery. Cloud computing, in particular, autonomic computing has

emerged as the new norm and cloud has cemented its presence in the IT diaspora. Organizations are revising their budget to include costs associated with the cloud. With the emergence of machine learning, artificial intelligence, and deep learning, the cloud services can only get better. The challenges faced today will drive more innovations and best practices. The scale at which connected devices are being added, the cost involved in cloud devices should be brought down further as in essential services. Post COVID-19, organizations in every sector will adopt cloud computing [31].

References

1. Faruqui, N. et al. (2020). Innovative automation algorithm in micro-multinational data-entry industry. Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325, pp. 680–692). LNICST. Springer. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Web.
2. Chakraborty, P. (2017). Simulation of reducing broadcasting protocol in Ad hoc wireless networks. *International Journal of Scientific & Engineering Research, 8*(7), 295–301.
3. Mansour, I., Sahandi, R., Cooper, K., & Warman, A. (2016). Interoperability in the heterogeneous cloud environment: A survey of recent user-centric approaches. *ACM*.
4. Singh, B. K., Alemu, D. P. S. M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
5. Tomar, R., Khanna, A., Bansal, A., & Fore, V. An architectural view towards autonomic cloud computing. *Data Engineering and Intelligent Computing*.
6. Yadav, A. K., Tomar, R., Kumar, D., Gupta, H. Security and privacy concerns in cloud computing. *Computer Science and Software Engineering*.
7. Jennings, B., & Stadler, R. (2014). *Resource management in Clouds: "Survey and Research Challenges (2014), Journal of New System Management"*. *New York*: *Springer Science + Business Media*.
8. Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing, 14*, 217–264.
9. Mustafa, S., Nazir, B., Hayat, A., Khan, A. R., & Madani, S. A. (2015). Resource management in cloud computing: Taxonomy, prospects, and challenges. *Computers and Electrical Engineering, 47*, 186–203.
10. Singh, S., Chana, I., & Buyya, R. (2017). STAR: SLA-aware autonomic management of cloud resources. *IEEE Transactions on Cloud Computing, IEEE*.
11. Odun-Ayo, I., Ananya, M., Agono, F., & Goddy-Worlu, R. (2018). Cloud computing architecture: A critical analysis. *IEEE*.
12. Braiki, K., & Youssef, H. (2019). Resource management in cloud data centers: A survey. *IEEE*.
13. Gill, S. S., & Buyya, R. (2018). Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: From fundamental to autonomic offering. *J Grid Computing, Springer Science+Business Media B.V., part of Springer Nature*.
14. Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2013). Heterogeneity in mobile cloud computing: Taxonomy and open challenges. *IEEE Communications Surveys and Tutorials*.
15. Xia, W., & Shan, L. (2018). Joint resource allocation using evolutionary algorithms in heterogeneous mobile cloud computing networks. *China Communications*.
16. Hummaida, A. R., Paton, N. W. & Sakellariou, R. (2016). Adaptation in cloud resource configuration: A survey. *Journal of Cloud Computing: Advances, Systems and Applications*.

17. Rosa, F. D. (2020). Worldwide software as a service and cloud software forecast", 2020–2024, Aug 2020. *International Data Corporation (IDC)*.
18. Kurtzman, W. (2020). Worldwide collaborative applications forecast, 2020–2024: Connectedness driven by COVID-19. *International Data Corporation (IDC)*.
19. IDG Cloud Computing Survey. (2020). Exploring cloud usage trends, investments, and business drivers. 2020 executive summary. *IDG Communications, Inc*.
20. Kerres, M. (2020). Against all odds: Education in Germany coping with covid-19. *Post digital Science and Education, Springer*.
21. Bhardwaj, T., & Sharma, S. C. (2018). An autonomic resource provisioning framework for efficient data collection in cloudlet-enabled wireless body area networks: A fuzzy-based proactive approach. *Soft Computing, Springer-Verlag GmbH Germany, Part of Springer Nature*.
22. Garg, S. K., Gopalaiyengar, S. K., & Buyya, R. (2011). *SLA-based resource provisioning for heterogeneous workloads in a virtualized cloud data center*. Berlin: Springer-Verlag.
23. Singh, S., & Chana, I. (2015). QoS-aware autonomic resource management in cloud computing: A systematic review, *ACM*.
24. Puri, G. S., Tiwary, R, & Shukla, S. (2019). A review on cloud computing. *IEEE Computer Society*.
25. Boukerche, A., & Meneguette, R. I. (2017). Vehicular cloud network: A new challenge for resource management based systems. *IEEE*.
26. Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems, 79*, 849–861.
27. Yemane, G., & Sharma, D. P. (2019). DPS-Yemane-Shareme CSMM model for client-side SLA of green cloud service measuring and monitoring. *The IUP Journal of Computer Sciences, XIII*(3), 34–46.
28. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R, & Safina, L. (2017). Microservices: Yesterday, today, and tomorrow. arXiv:1606.04036v4 [cs.SE].
29. Rudrabhatla, C. K. (2018). A systematic study of micro service architecture evolution and their deployment patterns. *International Journal of Computer Applications, 182*(29).
30. Esposito, C., Castiglione, A, & Choo, K. K. R. (2016): Challenges in delivering software in the cloud as micro services. *IEEE Cloud Computing published by the IEEE Computer Society*.
31. Xu, D., Li, T., Li, Y., Su, X., Tarkoma, S., Jiang, T., Crowcroft, J., & Hui, P. (2020). Edge intelligence: Architectures, challenges, and applications. arXiv:2003.12172v2 [cs.NI].

A Holistic Approach: Issues and Challenges in Autonomic Computation Toward Industry 4.0

A. Gautami and Naveenbalaji Gowthaman

1 Introduction

Human–technology bonding paves way to bring a big step in our prosperity. Industry 4.0 is an interconnection of things, business models, and supply value chains. A factory with maximum reliability and resource efficiency is termed a smart factory [1]. The manufacturing sector is a key factor for a nation's economy. An industry works with big data-operated quality control demands more data scientists. Figure 1 represents the production sites, the manual labors can be reduced to the robot-assisted coordinator and machines. Instead of manual logistics, nowadays, self-driving logistics vehicles are used to supply raw materials and end products to the market. Supply chain coordinators can handle the supply decisions and maintain networking in the market. Emergence of data science and machine learning helps the users to check the product failure with predictive technology. 3D printing and additive manufacturing of complex parts help the manufacturers to solve the problems in the assembly area.

By adopting Industry 4.0, the Smart manufacturers should focus on quality, efficiency, reliability, and customer relationship and satisfaction. Urbanization helps the smart manufacturers to cost optimization, production transparency, concentrate on zero defects, smart infrastructure and development, and sustainable eco-friendly production policies [2]. Exponentially growing technology may be the important thing to the transformation to Industry 4.0.

A. Gautami (✉)
Department of ECE, SNS College of Technology, Coimbatore, India

N. Gowthaman
Department of Electronic Engineering, University of KwaZulu-Natal, Durban, South Africa

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_6

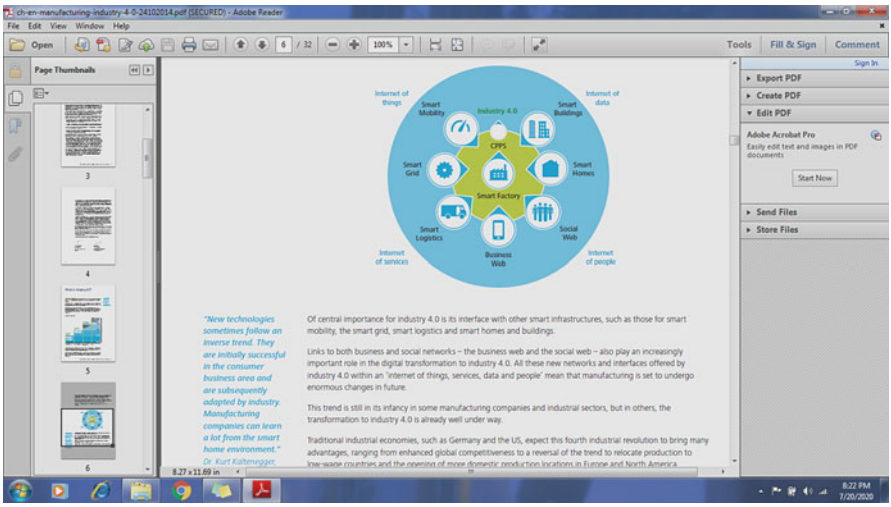

Fig. 1 Industry 4.0 in smart industries/factory

By research, Moore's law states the microchips, bandwidth, and computer systems double every 18 months and increase exponentially. 3D printing, sensor technology, AI, robotics, drones, and nanotechnology are a few examples of exponentially developing technology that is noticeably altering business processes, accelerating them with flexibility.

2 Industry IoT in Manufacturing

The Industrial IoT (IIoT) is a region focusing on high-quality sensing, computing, and networking, which constitute an important part of Industry 4.0 systems [3]. New featured Microprocessors and AI make the gadgets smarter with abilities of computing, communication, and autonomy management with social and economic needs.

The IIoT integrates gadget gaining knowledge of with clouds, grids nodes, and clusters for Big Data garage and analytics. In IIoT, end-user gadgets continuously generate and produce information, which results in truth and facts in Internet web page and Internet page site visitors inside the community among tool cloud verbal exchange. Big Data is turning into a primary contributor to enhance artificial intelligence in all elements of IIoT. Figure 2 shows the Architecture of the IIoT in the manufacturing industry.

- *Sensing systems*: Numerous sensing gadgets of various types are associated via the Internet to provide real-time facts constantly.

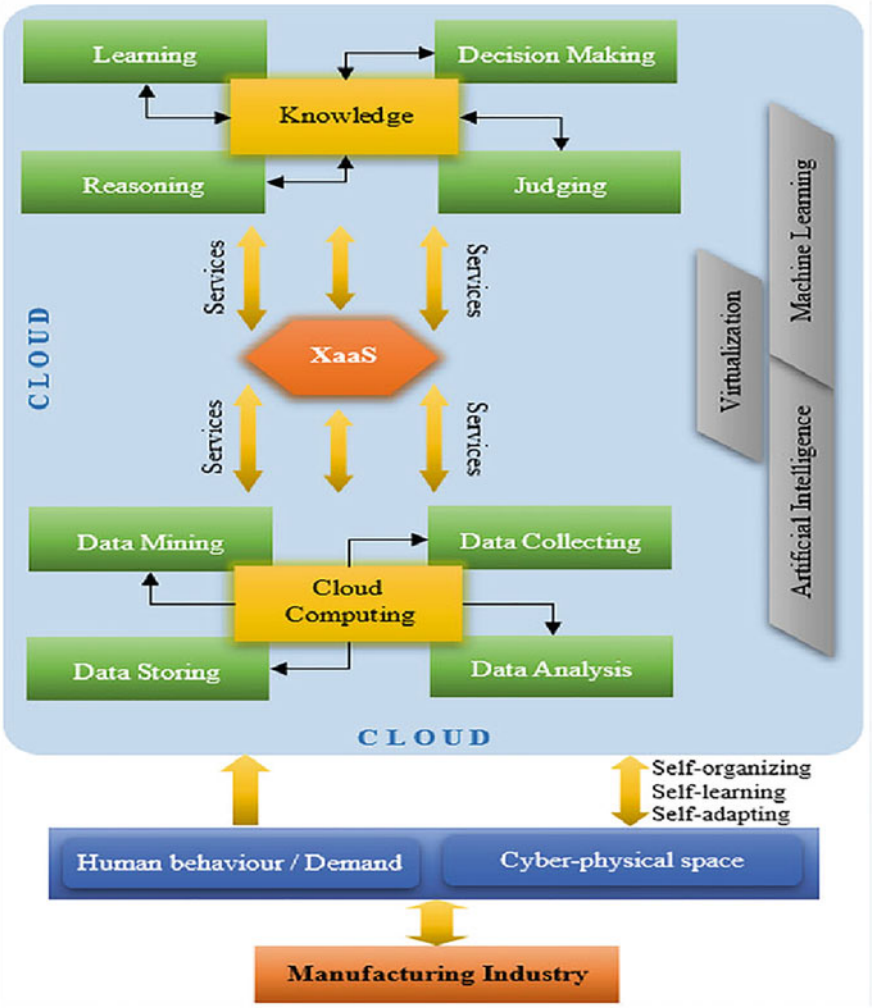

Fig. 2 IIoT architecture in the manufacturing industry

- *Outer gateway processor*: The servers for computation embody software program servers, element servers, switches, and routers, which might be electricity green and provide reduced delay and latency. The outer gateway processor assists to decrease bandwidth usage with the useful resource of decreasing and filtering statistics on the Internet.
- *Inner gateway processor*: It consists of micro-clouds and cloud-let servers that exist within the wide-area networks.
- *Outer processors*: It is a computing server used to record filtration and facts bargain and used for processing and routing at inner processors.

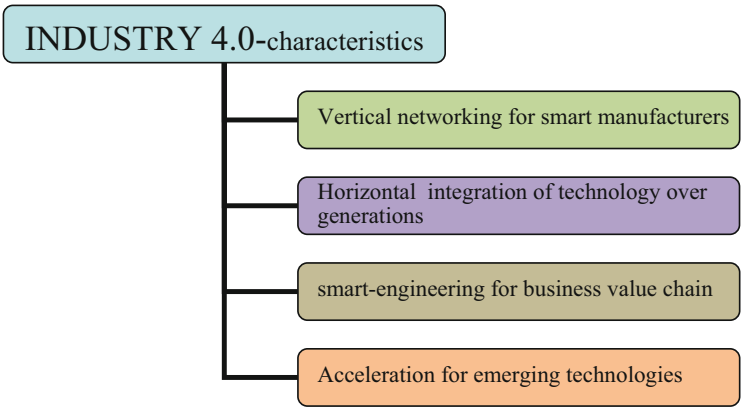

Fig. 3 Main characteristics of Industry 4.0

- *Inner exceptional processors*: It consists of clusters, grids, and clouds to stay a protracted way from the sensors.

3 Characteristics of Industry 4.0

The following four main traits of Industry 4.0 from the figure reveal the massive functionality that enterprise and traditional production have for trade: vertical networking, horizontal integration [4, 5], smart engineering, and acceleration. Figure 3 represents the characteristics of Industry 4.0.

3.1 Vertical Networking for Smart Manufacturers

Vertical networking is the major function of future smart industries. This vertical networking follows cyber-physical production systems (CPPSs) [6] to permit production gadgets to react hastily to alter in call for or stock stages and to defects. Smart industries arrange on its own and allow customer-specific manufacturing and individualized manufacturing. These calls and stocks need data for processing. Smart sensor-based technology helps with tracking and self-reliant business enterprise.

Networking for resources, raw materials, products, and parts may be positioned anywhere and everywhere and at any time. The discrepancies are logged properly and routinely for major processing tiers in the manufacturing arena. Equipment wear and tear, fluctuation, and equipment handling are maintained with great care. Such approaches additionally permit placed on and tear on substances to

be reviewed with greater efficiency. Symbolic emphasis is hooked up to resource performance and specifically, the use of renewable resources and manpower. The demands on people involved in major functionalities, which include manufacturing, warehousing, logistic firms, and maintenance, are also changing, which means that new abilities in green running with CPPs are required.

3.2 Horizontal Integration of Technology Over Generations

Horizontal integration, the second characteristic of Industry 4.0, is an upcoming generation of enterprise value networks. These networks are real-time networks with promising optimization that allows blanketed transparent characteristics and facilitate higher international optimization. The product records are traced and accessed at any time [7, 8]. This creates flexibility across device chains—from searching for through sales/production. Customer-precise necessities can be made hard inside the manufacturing; however, additionally, in the development phase, planning phase, composition, and distribution phase, allowing elements together with amazing, time, hazard, charge, and sustainable environment to be dealt with real-time and in any respect tiers of the price chain. This shape of horizontal integration of every client and commercial employer companion can produce new industrial organization models and new fashions for courting, representing an assignment for all the ones involved. Legal problems and the obligation to protection of intellectual belongings are getting increasingly more critical.

3.3 Smart Engineering for Business Value Chain

The next important function of Industry4.0 is smart engineering all through the rate chain and lifestyle cycle of each merchandise and client. This engineering takes place effortlessly at some level for improvement and the manufacture of recent products and services. New products need advanced manufacturing structures [9]. The improvement and manufacture of new merchandise and production structures are included and coordinated with product lifestyle cycles, allowing new synergies to be created among product improvement and production systems. Characteristic of this smart engineering includes product's life cycle, advanced techniques to be defined from records via prototype modeling, and the by-product degree.

3.4 Acceleration for Emerging Technologies

The fourth critical characteristic of Industry 4.0 is the impact of exponential technology which acts as a catalyst that permits flexibility and financial demands in

commercial techniques. Artificial intelligence (AI), automated robotics, and sensor technology have the capacity to boom in the future [10]. Functional nanomaterials and nanosensors can be used in manufacturing control talents to permit the manufacturing of next-era robots. For example, surveillance and maintenance robots in production halls and warehouses to deliver spare elements at any time of day or night inside the self-preserving and smart factories.

4 Elements of Industry 4.0

Industry 4.0 incorporates automation, and it, in particular, contains permitting generation including CPS, IoT, and cloud computing. Table 1 represents elements of Industry 4.0 with its key factors.

In Industry 4.0, the mixture of a virtual area with the physical global is completed using processor-based structures, semantic machine-to-system interaction, IoT technologies; except, a state-of-the-art generation of commercial company systems,

Table 1 Elements of Industry 4.0 with its key factors

Elements	Key factors
Internet of things	Sensors and actuators, RFID GPS Wireless sensor networks peer-to-peer networks D2D Services
Cyber-physical systems	Computational algorithm Smart communities Virtual objects
Cybersecurity	Application security Network security Disaster recovery Production and operation security User security
Big data	Cloud computing Volume Veracity Variety Velocity Validity Volatility
System integrity	Horizontal integrity Vertical integrity End-to-end integrity
Embedded tools	Augmented reality/virtual reality Robots 3D printing and additive manufacturing Simulation

alongside smart factories, is evolving to cope with the manufacturing within the cyber-physical setting. Some key elements and their functionalities are given in the table shown above.

5 Autonomic Computing Challenges

1. *Flexible client integration* is an opportunity to enhance performance and productivity. The corporations take issue inside the survey to combine their clients' needs into development and production.
2. *Customization* is a route of manufacturing enterprise in the future. Customers' need is a key to decide how their products are made by adopting major strategies based on problem statement at a trigger point.
3. *Scalability* represents each trouble in the setting of Industry 4.0. This trouble gathers the importance of physical items used in manufacturing networks and strategies.
4. *Resources Infrastructure*: Industry 4.0 calls for current installations to be tailored completely to new forms of IT infrastructure [11, 12]. Vast structures need to be networked and developed from the initial point of contact.
5. *Data Challenge*: Industry 4.0 is best for data analytics; it generates statistics for every technique and tests its consistency. In the production environment, various information is accumulated via one-of-a-type property such as machine sensors, product records, plant statistics infrastructure facts, and logistics facts, all of which contribute to facts' length [13].

   To conquer those situations, new algorithms, products, and models are required to be applied and advantage of records. Engineers are required to break down data and to discover the relationship among fact streams. Data are probably saved in heterogeneous database solutions.
6. *Data collaboration*: Outsourcing companies with their capabilities expose the information to the collaborators with who they have collaborated.

   They want to alternate information among their branches to preserve their optimized techniques. Industry 4.0 focuses on sharing its techniques and infrastructure as a service to its corporations for funding and investment benefits. Transparency of data is needed at the records of the corporation for selling their infrastructure or product to them [12, 13]. CRM- and ERP-level integration is regularly lacking. Normally, in a manufacturing firm, a product has best 3°, manufacturing started, in development, and finished, which are not properly protected with ERP systems.
7. *Cyber and data security risks*—Security risk is a major threat in Industry 4.0 requirements. Industries want to ensure their human, product, and manufacturing facilities' surroundings are secured from dangers. A smart device like sensors may be used for monitoring the premise of hardware and software programs. All gadgets that consist of business machines want to be up to date with time as a way to get comfy within the course of potential threats which can be arising on a

day-by-day basis. IoT devices with high quality have small processing energy, so a new set of tools are required for monitoring facilities [14, 15]. A cloud-related device is vulnerable to thefts and risks. The internet of things is a vulnerable platform for theft and attacks by hackers. Cyberattacks and viruses ought to have attack on networked and smart manufacturing.

6 Autonomic Computing Industry 4.0 Solutions

Based on the characteristics of IIoT, the range of solutions is considered from the table mentioned below.

6.1 Solution for Vertical Networking of Smart Manufacturers

Vertical networking solutions	
Integration with Information Technology	• The vertical networking requires new solutions for Industry 4.0. In the present scenario, IT infrastructures are very fragile and result in poor networking • Advanced solutions can be imposed as an additive from companies of sensors, control systems, communications networks, applications, and packages • Companies making the right preference of integrating the modern solutions will be a solution for extended-term market benefit
Data process management	• Collecting, analyzing, and processing huge information will create new insights, resource decision-making under industry 4.0 • Companies need to expand their abilities in the regions of analytics and information management
Cloud management	• The cloud network gives notable opportunities to host and make inexperienced use of big data • The advantages for decentral networked manufacturing structures will permit cloud-based applications • Cloud management is an answer for seamless integration of all ranges of providers' rate chains to save clients and new innovative products
Performance and efficient digital transformation	• The evaluation of data with proper assessment of the records collected from products and sensors permits operational safety, servicing, and safety [16] • Transparent characteristics make efficiency, and operational fee discounts for clients create extended-term advantages among customers

6.2 Solutions for Horizontal Integration of Technology Over Generations

Horizontal integration solutions	
Value chain modeling	• Industry 4.0 method adopts new tactics for commercialization of enterprise business in preference for making incremental upgrades to set up business models • Successful agencies have developed abilities to boom in all cutting-edge innovative segments in their company
Supply chain and IP management	• Research and development, production, and profits skills are main features of digitization advancements • Usage of higher communications to mix companies and customers' dreams is validated. New enterprise fashions and cooperation are blooming as a result of Industry 4.0
Logistics control and taxation models	• Integration of self-sufficient technology, new offerings, warehousing, and distribution and the interlinking of inner manufacturing are the major challenges in Industry 4.0 as far as taxation is concerned

6.3 Solutions for Smart Engineering of Business Value Chain

Smart engineering solutions	
Life cycle management	• Artificial Intelligence will use global skip-checking of devices in the market. Based on specifications, it performs checking mechanism on products with defects
Innovation management	• In product improvement, records generation is used to improve research and development [17]. The sharing of records among current worldwide networks with some traces of "assignment networks" can be used for developing a new community
Product and customer life cycle	• Industry 4.0 will move disciplinary engineering for rate chain in the future. It mainly focus on products and consumer lifestyles cycle in leading manufacturing firms

6.4 Solutions for Acceleration for Emerging Technologies

Acceleration solution	
Corporate acceleration	• Corporate acceleration gives appropriate opportunities for making an investment in new inclinations at an early stage for benefitting from innovation and exponential era
Learning process	• New thoughts, techniques, and enterprise organization segments are explored for migrating to the center of the commercial enterprise business enterprise

7 Future Trends in Autonomic Computing

Further trends have been rooted in optimization, understanding pressure on costs and charges, the release of new device upgrades, and/or one in each of a type of revolutionary solutions. There has been a specific recognition on following production places to globalization [18, 19]. With regard to new offerings, nearby issuer provision—consisting of precise spare additives and provider agencies and one-prevent stores for services and products—becomes particularly popular. Further dispositions stated by means of manner of character respondents protected preventive protection, automation of logistics, and smart factories.

These added values and conditions are related to the problems raised via Industry 4.0 and exponential technology. The virtual transformation and the need for additive manufacturing in the production location, alongside any other innovative technology, will permit organizations to take a greater effort to explore those opportunities to the masses under stressful conditions.

References

1. Havle, C. A. & Ucler, C.. (2018). Enablers for industry 4.0. ISMSIT 2018 - 2nd Int. Symp. Multidiscip. Stud. Innov. Technol. Proc. (pp. 1–6).
2. Ervural, B. (2019). Overview of cyber security in the industry 4. 0 Era. September 2018.
3. Moktadir, A., Ali, S. M., Kusi-Sarpong, S., & Shaikh, A. A. (2018). Process safety and environmental protection. *Process Safety and Environmental Protection*.
4. Dawson, M. (2018). Cyber security in industry 4.0: The pitfalls of having hyperconnected systems. *Journal of Strategic Management Studies, 10*(1), 19–28.
5. Khan, A., & Turowski, K. (2016). A survey of current challenges in manufacturing industry and preparation for industry 4.0 (pp. 15–27).
6. Hanstein, B. Whitepaper: IT and IT infrastructure in the context of Industry 4.0. Rittal.

7. Frost, & Sullivan. (2017). Cyber security in the era of industrial IoT.
8. Bligh-Wall, S. (2017). Industry 4.0: Security imperatives for IoT — Converging networks, increasing risks. *Cyber Security: A Peer-Reviewed Journal, 1*, 61–68.
9. Sethi, P., & Sarangi, S. R. (2017). Internet of things: Architecture, issues and applications. *International Journal of Engineering Research and Applications, 07*(06), 85–88.
10. Alkhalefah, H. (2018). Requirements of the smart factory system : A survey and perspective. *Machines, 6*(2), 23.
11. Gorecky, D., Weyer, S., Hennecke, A., & Zühlke, D. (2016). Design and instantiation of a modular system architecture for smart factories. *IFAC-Papers Online, 49*(31), 79–84.
12. Iivari Petri, M. M. A., Komi, M., Tihinen, M., & Valtanen, K. (2016). Toward Ecosystemic business models in the context of industrial internet. *Journal of Business Models, 4*(2), 42–59.
13. Maslarić, M., Nikoličić, S., & Mirčetić, D. (2016). Logistics response to the industry 4.0: The physical internet. *Open Engineering, 6*(1), 511–517.
14. Fleisch, E., Weinberger, M., & Wortmann, F. (2014). Geschäftsmodelle im Internet der Dinge. *HMD Prax. der Wirtschaftsinformatik, 51*(6), 812–826.
15. Zhang, Y., Qu, T., Ho, O., & Huang, G. Q. (2011). Real-time work-inprogress management for smart object-enabled ubiquitous shopfloor environment. *International Journal of Computer Integrated Manufacturing, 24*(5), 431–445.
16. Stock, T., & Seliger, G. (2016). Opportunities of sustainable manufacturing in industry 4.0. *Procedia CIRP, 40*, 536–541.
17. Müller, J. M. (2018). What drives the implementation of industry 4.0 ? The role of opportunities and challenges in the context of sustainability. *Sustainability, 10*(1), 247.
18. Prause, G. (2016). Sustainable business models and structures for industry 4.0. *Journal of Security and Sustainability Issues, 2*(December 2015).
19. Faruqui, N. et al. (2020). Innovative automation algorithm in micro-multinational data-entry industry. Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325, pp. 680–692). LNICST, Springer.

Resource Management Issues and Challenges in Autonomic Computing

Palak Gupta, Sudhansu Shekhar Patra, Mahendra Kumar Gourisaria, Aleena Mishra, and Nitin S. Goje

1 Introduction

For a long time, computing and data innovation have been focused for the quest of rapid, VLSI, powerful computing, and so forth. As a result, the present day's computing and data frameworks have achieved a degree of complex nature, where the exertion required to prepare the frameworks for activity and keeping them in a working condition is becoming hectic. The current IT conditions are perplexing and heterogeneous, and programming from numerous sellers is getting progressively hard to coordinate, install, design, and keep up to date. At the current pace of development, even the most talented IT experts may think that it is difficult to regulate IT situations in a couple of years. A comparable issue was experienced around the 1920s in communication. Human administrators at that time were required to work with manual switchboards, and as the use of phone expanded quickly, there were significant issues about the number of trained administrators to work with switchboards. Luckily, there was exchange of autonomic branch which eliminated the requirement for human intervention.

Industry 4.0 is a large-scale idea that was characterized as an initiative to secure the fate of German industry. The majority of the IT business perceives that the main suitable remedy for this approaching emergency is to supply systems and parts that include the capacity to oversee themselves per significant level targets directed

P. Gupta · M. K. Gourisaria · A. Mishra
School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar, India

S. S. Patra (✉)
School of Computer Applications, KIIT Deemed to be University, Bhubaneswar, India

N. S. Goje
Department of Computer Science, Webster University, Erbil, Tashkent, Uzbekistan

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_7

by the humans. Some administration technologies are arranged manually, while some frameworks give programmed configuration initially but lack the support for dynamic setup. A couple of frameworks give dynamic configuration capability by adapting new component in their source code, while others give outside interface for the setup prerequisites. IBM presented the vision of self-managing frameworks in 2001 when it propelled the autonomic computing activity. Hewlett-Packard's Adaptive Enterprise activity and Microsoft's Dynamic Systems activity are connected industry endeavors that perceive the self-administrating segments and frameworks, which are fundamental to fate of IT. Autonomic computing without a doubt delivers significant effects both upon the control of processing and the IT ventures. It tries to enhance the frameworks of computing with a comparative goal of diminishing human contribution.

The prerequisite and support for autonomic computing depend on the consistently expanding complexity in the present framework. It has been stated that the IT industry's only aim has been on enhancing equipment execution, with the programming to augment on the extra limit, and neglecting the other virtual criteria. With the growing demand of consumers for an upgrade cycle, it has made this a trillion-dollar industry. Its heritage, however, is a bulk of unpredictability inside systems of systems, bringing out an expanding monetary weight for each PC (often estimated as the TCO: total cost of ownership).

Many researchers and disciplines were amazed to the autonomic initiative, as various artificial intelligence (AI) and fault-tolerant computing (FTC) have been exploring huge number of issues within AC for a long time. The longing for automation and viable powerful frameworks is not new, but something that is new is the autonomic computing's encompassing motive of collecting all the significant regions together to bring a change in the direction of industry, self-ware rather than the equipment, and software feature redesigned the pattern of the past that made the complexity and TCO untidy. For now, this is long-term vital activity with transformative expectations to en route.

2 Autonomic Computing

Autonomic computing (AC), as the name proposes, is a representation dependent on science. The autonomic sensory framework inside the body is fundamental to a considerable measure of nonconscious movement that permits us as people to continue with more elevated degree of action in everyday living. Common models that have been featured are heartbeat rate, breathing rate, reflex responses, after contacting a sharp or hot article, etc. The point of utilizing this analogy is to communicate the vision to empower something like being accomplished in registering, at the end of the day, to make self-administration a significant measure of the computing capacity to calm clients of low level the board exercises, permitting them to put accentuations on the more elevated level worries of maintaining their business, their trials, or their amusement.

2.1 Evaluation

Autonomic computing, propelled by IBM in 2001, is developing as a critical new key and all-encompassing way to deal with the structure of computing frameworks. Two of IBM's fundamental targets are to diminish the absolute expense of responsibility for and to discover better methods for overseeing their expanding unpredictability. Just as IBM, many significant programming and framework sellers, for example, HP, Sun, and Microsoft, have built up key activities to help make PC frameworks that oversee themselves, inferring that it is the main reasonable long-haul arrangement. The craving for computerization and compelling hearty frameworks is not new; in actuality, this might be viewed as part of best practice program designing. Thus, the wants for frameworks mindfulness, the consciousness of the outside condition, and the capacity to adjust are additionally not new but significant objectives of man-made brainpower (AI) research for some years. Figure 1 shows the development features of automatic computing. Research in autonomic computing is probably observing a more noteworthy coordinated effort between AI and programming building fields. Such a coordinated effort has been supported by expanding framework unpredictability and a more requesting client network. Thus, autonomic computing maybe best considered a key pull together for designing of powerful frameworks as opposed to a progressive new approach. Figure 2 shows the various characteristics of automatic computing.

Self-configuring: Frameworks adjust consequently to powerfully changing environments. When equipment programming frameworks can characterize themselves "on-the-fly," they are self-configuring. This perspective of self-managing implies

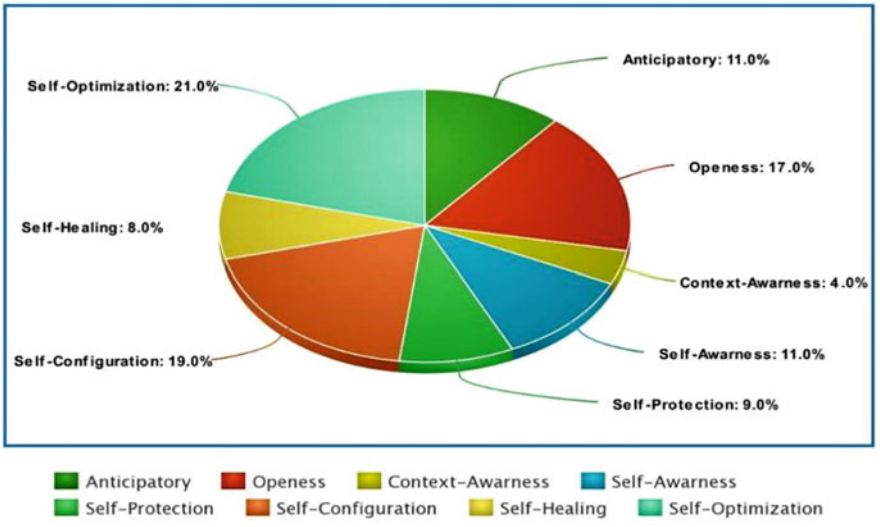

Fig. 1 Development features of autonomic computing

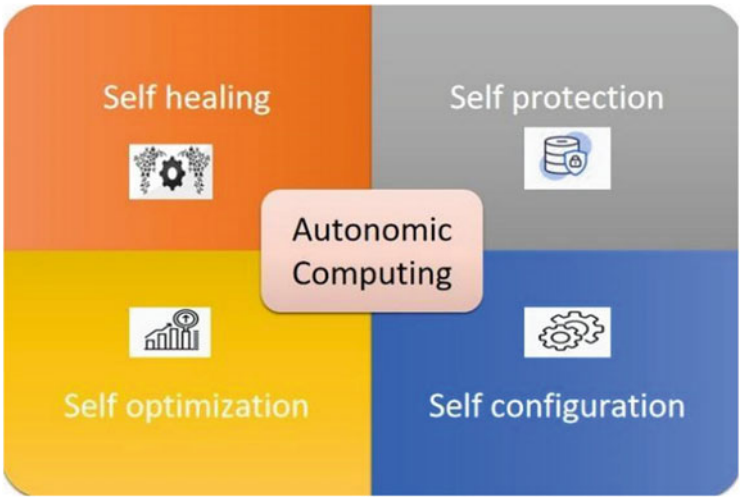

Fig. 2 Autonomic computing

new highlights and programming. What is more, servers can be progressively added to the venture foundation with no disturbance of the administrations.

Self-healing: Frameworks find, analyze, and respond to interruptions. For a framework to act naturally recuperating, it must have the option to recuperate from a breakdown part by first identifying and disengaging the breakdown part, taking it disconnected, fixing, or separating the malfunctioning segment, and reintroducing the fixed or substitution segment into administration with no clear application interruption.

Self-optimizing: Frameworks screen and tune assets naturally. Self-optimization requires equipment and programming frameworks to productively amplify asset usage to meet end-client needs without human intercession.

Self-protecting: Frameworks foresee, recognize, distinguish and furthermore shield themselves from assaults from anyplace. Self-ensuring frameworks must be able to characterize and oversee client access to all registering assets inside the undertaking, to secure against unapproved asset access, to distinguish interruptions and report and forestall these exercises as they happen, and to give reinforcement and recuperation abilities that are as secure as the first asset the board frameworks.

2.2 Frameworks

A wide field of autonomous research computing focuses on designing self-propelled technologies and management of properties. In both, work is done to plan structures, designs, and frameworks that help to reduce system complexity and promote autonomous behavior. Most scholarly works are supported by IBM. Such works

describe which project foundations are mainly geared toward the two design accesses, externalization and internalization.

- In the externalization strategy, self-management modules are beyond the managed program.
- Application of specific self-management within the managed system is done in the internalization approach.

The research work focuses on autonomy because of the high-level classification dependent on the methodology utilized in frameworks and infrastructures that should be readily accessible as given below.

(a) Biologically inspired Frameworks and Architecture:
Biological systems have inspired autonomic computing. There has been both centralized and decentralized solution. The centralized approach aims toward the job of human anxiety. Framework as controller for managing and keeping up another body system. Adopt decentralized strategies motivation from nearby and worldwide conduct of natural cell and the networks of the ant colony.

(b) Large-Scale Distributed Application Frameworks:
Another area is the development of large self-managed distributed databases and system scales. A project called Oceano prototype of an architecture extremely accessible and robust. Self-configuration is provided dynamically allocating and disbursing resources such as server appliances.

(c) Framework using Agent Architecture:
To build infrastructures supporting autonomic behavior by using agent architecture. The architecture of agents uses a decentralized approach to allow autonomy comportment. Each agent has their its own local control, and they interact with different operators for the formation of a self-guided system. Own local control is managed by each agent to form a self-guided framework. Unity uses a goal-driven approach to achieve self-recuperating, constantly self-improving, and self-assembly.

(d) Component bases frameworks:
For enabling autonomic behavior, component-based framework was proposed. The treaty frameworks help to build and maintain the autonomous applications in computerized grid environment. It enables self-configuration by separating segment actions from part cooperation.

(e) Technique-Focused Frameworks:
Many systems employ techniques such as Artificial Intelligence (AI) and hypothesis of control. In artificial intelligence, the predicate-based planning system was submitted. It achieves abstract aims defined by user's current account context and security policies. The forecast behavior is optimized by the controller according to the predetermined nature of administration requirement.

(f) Self-administrated service-oriented infrastructure:
An infrastructure for allowing autonomy was suggested in behavior in architecture oriented at service. Autonomy web service does monitor, analysis, and planning and life cycle execution. When an error occurs, functional web

administration subscribed to autonomous web service. The functional web service provides autonomous web administration with its log documents and strategy databases.

(g) Architecture for injecting automaticity into non-autonomous frameworks:

The architectures proposed for autonomous injection inheritance and non-autonomous behavior, where plan and data about the code are not accessible. A portion of the cadres identified utilize layered engineering, case-based thinking, and a rule-based way to allow automaticity in present-day frameworks. Layer of choice includes queries that every legacy framework is tailored to. The gauges are contained in a definition layer that maps data into device model. Controllers are contained by top-most decision layers that assess the consequence of data perceived.

2.3 Applications

2.3.1 Applications for Self-Healing Systems Autonomic Computing

The main feature of autonomic computing is its self-healing devices. It improves stability and reliability of the systems by continuous monitoring and control of the components. The IT industry needed a computer system that would allow experts to spend a little less time in issue solving and more time in actually performing the tasks. During runtime, self-healing systems maintain a satisfactory quality of systems to avoid inappropriate control systems behavior. This is the cycle's first phase. The second stage of the cycle is to discover and detect error. The final phase is the planning of an error-related corrective action. When the errors are corrected, the process begins again.

2.3.2 Requests of Autonomic Computing in Virtualized Environment

Managing the dynamic information technology sector structures is complicated. Virtualization is the strategy used to differentiate the device's resources into different operation environments. The data are securely transferred via virtual server systems using virtualization.

2.3.3 Autonomic Computing Applications for Business

The autonomous computing systems are reconfigured in a different manner to match the particular business strategy. We see some of its applications in customer relationship management. A simple algorithm handles and assesses a certain mistake that relates to budgetary allocations, resource allocations, revenue generation, and lead

execution. In addition, autonomic computing manages cost structure via different available schemes.

3 Literature Review

Decentralization of work process execution is a significant territory of examination. Basically, it is done to help business processes across organizations without utilizing an incorporated element. This sort of procedure decentralization can prompt higher adaptability, yet it additionally presents a few issues all alone, for example, the absence of a worldwide view over the procedure. It likewise does not address the adaptability and unwavering quality issues, as the issue is essentially meant to every node that gets executed in portions of the procedure. To address the above-referenced issue, many instruments have been suggested (e.g., GOLIAT) that utilize the normal attributes of the outstanding task at hand to make forecasts related to the execution of a specific design of the motor. At the time of deployment, these devices help framework administrators to decide intuitively on what number of assets the motor ought to be disseminated to accomplish the ideal degree of execution. With this growing methodology, autonomic computing methods can be utilized to supplant such manual and static design steps. In [1], a way to deal with self-upgrading PC frameworks has been created. This methodology utilizes an online control algorithm that depends on the task remaining to ideally reconfigure a web server as for QoS objectives over a restricted time skyline. This issue of adaptively imitating functionality to accomplish higher throughput has likewise been recognized by the database network in which the unbounded replication of functionality can prompt execution losses. The problem in this way is to reproduce particular functionality contingent upon the workload only when needed.

A significant assortment of tasks has been devoted to autonomic correspondence standard and attributes. But still a literature search does not yield applicable advancement in assessing and contrasting Autonomic Network Management (ANM) models. The significant difficulty is the scattering of research on the architectural zone of autonomic networking. In [2], a research has been done on autonomic system's frameworks and foundations. Initially, the creators have classified the existing arrangements into several zones: biologically motivated, large-scale circulated, operator-based, segment-based, strategy-centered, self-guided service situated, and non-autonomous framework explicit structures. In the subsequent part, the creators have specified and investigated the procedures that could be utilized to accomplish the abilities of self-administration. Despite everything, some significant problems still remain. The authors [3] have presented a diagram of the current methodologies concentrating on each of the administration functionalities is introduced. They have provided an arrangement system to characterize existing research endeavors. Moreover, the authors have distinguished properties required to empower the architecture with autonomic standards into six classes or zones. The

properties include level of action, capacity to learn, granularity of the insight, level of mindfulness, memory quality, and level of self-operation.

IBM has proposed a combination of Autonomic Processing Adoption Model Levels to portray the level of autonomicity of a system. The order proposed depends on how much the administrator is engaged with the assignments. In [4], for the comparison of autonomic computing solutions, many measurements and metrics have been given. These measurements and metrics incorporate quality of service (QoS), cost of self-sufficiency, granularity/adaptability, etc. To capture different execution parts of an appropriate ANM design, the authors [5] recognized a set of evaluation perspectives. Each view is depicted is similar to the MANET case analysis. The delineated views incorporate level of self-operation. Execution of autonomic reaction (QoS), cost of self-sufficiency, speed of autonomic reaction, affectability to change, and granularity. The utilization of radar diagrams such as a graphical showcase for contrasting between various perspectives of few frameworks has also been proposed. In [6], the authors have raised the problem of integrating different submetrics into a general proportion of autonomicity and calibrating scores across various frameworks.

The fundamental idea of control loop is based on, shown in Fig. 3, the architecture level of autonomic systems. This behaves as a chief to monitor, investigate, and apply appropriate activities on a specific set of predefined framework strategies. In [4], two classifications have been identified to deal with autonomic computing frameworks, namely, tightly coupled and decoupled autonomic frameworks. The

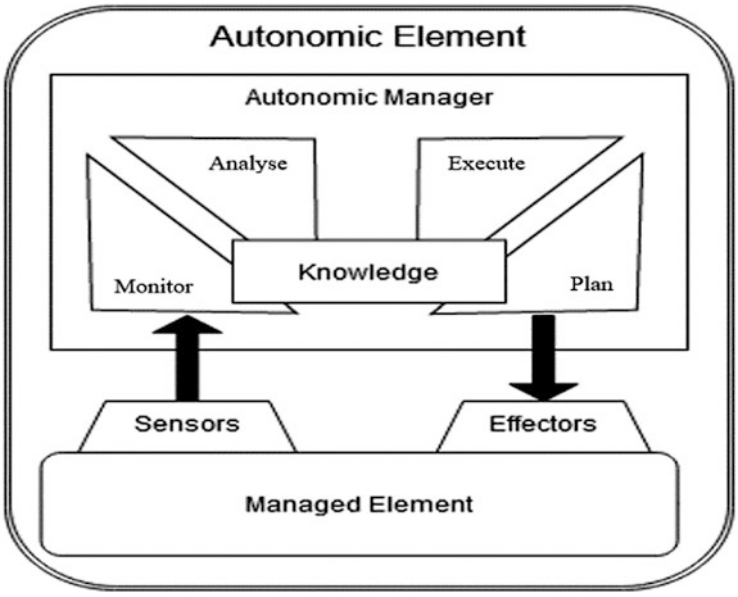

Fig. 3 Autonomic control loop

tightly coupled autonomic framework is constructed by utilizing the keen operators with their objectives, and the decoupled autonomic frameworks are the ones where the foundation manages the autonomic conduct of the framework. Both the categories have basic ideas, and sometimes it becomes an issue to examine the project in a specific classification. In these two methodologies, there is a requirement of two sorts of components for the execution of functionalities of the target framework and for presenting a few examples for framework self-administration. These arrangements of components depict two zones of system infrastructure in particular, intracomponent and intercomponent relations. A few imperative research subjects and mechanical arrangements address aspects such as composition formalisms that can assist with the automation generation of worldwide configuration by thinking about the certain constraints as well as its enhancements; smart components that can assist with adjusting to natural changes for giving structure blocks in self-administration framework; and hot swapping can assist with creating self-administration framework with the help of code interpositioning or code replacement.

The necessity of streamlining the administration and activities of IT-based frameworks prompted numerous merchants who have no uncertainty to enter the area of autonomic computing. The big undertaking merchants, for example, IBM, Sun Microsystems, HP, Microsoft, Intel, Cisco, and a few different merchants, have created a variety of frameworks and solutions. In [7], SMART and IBM have presented a database environment that decreases the complex nature and increases the quality of administration with the increment of self-managing capacities. Oceano, IBM, has designed and build up a pilot model of a versatile and sensible foundation for an enormous scale processing utility powerplant. Optimal Grid, IBM, has streamlined the production and execution of big-scale, connected, parallel grid applications by enhancing execution and incorporating autonomic matrix usefulness as a model middleware. In [8], AutoAdmin, Microsoft has made database framework self-tuning and self-regulating by empowering them to follow the utilization of their systems and to effortlessly adjust to application prerequisites. N1, Sun, manages server centers by incorporating resource virtualization, administration provisioning, and strategy automation procedures. The Adaptive Enterprise, HP, has encouraged clients to assemble a framework in three levels, specifically business, administration, and resource [9].

A striking number of projects in software engineering, software designing, and artificial intelligence are identified with autonomic processing, as it is a multidisciplinary research zone. OceanStore, US Berkeley, plans a worldwide steady data store for scaling to billions of clients [10]. In [11], the authors have supported the structure, usage, and assessment of P2P applications in view of multi-operator and transformative programming acquired from complex versatile frameworks. In [12], the research has been done on the novel strategies for building exceptionally reliable Internet administrations and recuperation from failures. Various investigations of the interactions between mobile, multispecialist frameworks, and artificial intelligence techniques are done [13]. Authors in [14] have created fault management strategies

that are planned for clearing the system's interior state to forestall the event of extreme failures of crash in the upcoming times.

In the undeveloped years of autonomic computing, a generous research exertion in this zone was done. The authors in [15] talk about the effect and machine structure [16]. Basically, it supports those analysts and AI specialists who hold the view that affect is essential for intelligent conduct [17, 18]. It has suggested three degrees for the structure of framework. Response is considered to be lowest level, where no learning happens; however, reaction to state data originating from sensory frameworks. Routine is the middle level, where generally routine assessment and planning of practices takes place. It gets contribution from sensors just as from the reaction level and reflection level. The topmost level is the reflection level which gets no sensory information or has no motor yield, and it gets contribution from underneath.

A fundamental problem for the accomplishment of autonomic computing is its capacity to move information related to the framework management and design from human specialists to the product dealing with the framework. Essentially, it is an information securing issue [19]. Here the authors have suggested to consequently catch the activities of the experts when performing a live system and progressively assemble a methodology model that can execute on another framework to repeat the same undertaking. Setting up an assortment of traces after some time ought to permit the way to approach to build up a conventional and versatile model. In [20], the Tivoli management environment moves toward this issue by catching the main attributes of a managed resource in its asset model. This methodology is being reached out to get the information from best procedures into the common information model (CIM) through clear logics at both stages of planning and deployment of the advancement life cycle [21]. Essentially, the methodology catches the information of framework from the creators, at last, to perform automated reasoning while dealing with the framework. The utilization of probabilistic strategies, for example, Bayesian systems (BNs), talked about in [22], is likewise vital to the research of autonomic algorithm selection. The framework utilizes the BN approach alongside self-preparing and self-upgrading to discover the best algorithm. As a result, the expansiveness and extent of the autonomic vision are featured by such works that use methodology of Artificial Intelligence for controlling the identification of the requirement for re-optimization of big business goals [23].

Today, AC is broadly accepted as a promising way to deal with growing new systems. Yet, associations proceed to deal with the heritage of frameworks or manufacture systems of systems involving new and legacy segments, including divergent advances from various vendors [24]. Usually, the designing of autonomic capacity into legacy frameworks includes giving a situation that monitors the sensors to the system and gives modification through effectors to make a control loop. One such foundation is KX (Kinesthetics eXtreme), which runs a lightweight, decentralized, and effectively incorporated assortment of dynamic middleware segments integrated by means of a publish–subscribe event framework. The Astrolabe instrument might be utilized to computerize self-setup and monitoring and to control adjustment [25].

Strategy-based administration turns out to be especially significant with the coming vision of autonomic computing, where a manager may essentially indicate the objectives of business, and the framework will make it so in terms of the required ICT. A strategy-based administration tool may lessen the intricacy of item and framework administration by giving uniform cross-item strategy definition and the foundation of management.

4 Issues and Challenges

The problem for autonomic computing needs something more than restructuring of the present frameworks. Even after various scholarly and modern undertakings acknowledged wonderful pieces of the autonomic computing vision, there still exist issues to manage in this field of study. Autonomic computing requires new considerations, new experiences, and new methodologies. Since autonomic processing is another thought in large-scale heterogeneous frameworks, meeting this challenge of AC presents major and significant research challenges that spread through all the zones. Some of them have been explained.

4.1 Autonomic System Challenges

4.1.1 Relationships Among Autonomic Elements (AEs)

Relationship among AEs has a main job in actualizing self-administration. All the connections have a life cycle comprising of detail, area, arrangement, activity, and end stages. Difficulty is there in every stage. Expressing the arrangement of yield benefits that an AE can perform, just as building up the grammar and semantics of standard administrations for AEs, can be a particular test. AEs additionally need conventions and procedures to set up the policies of exchange and to deal with the progression of messages among the negotiators. The main difficulty emerges for the planner to create and break down arrangement of algorithms and conventions and then figure out which exchange of algorithms can be viable.

4.1.2 Learning and Optimization Theory

The primary challenge here is the exchange of management system information from human specialists to ACSs. For this, the researchers have recommended to watch the conduct of people in solving an issue on various frameworks and by utilizing traces of their exercises, a strong learning technique can be made. Encouraging the information obtained from the human and delivering systems that incorporate this information likewise turns into a challenge. In modern frameworks,

individual components that connect with one another must likewise adjust in a unique environment and figure out how to solve issues depending on their previous encounters. Optimization is a problem too in light of the fact that in such frameworks, adjustment changes the conduct of agents to arrive at advancement.

4.1.3 Robustness

Robustness has been served in different sciences and frameworks, for example, nature, engineering, and social frameworks. One can decipher it as strength, dependability, survivability, and adaptation to noncritical failure despite the fact that it does not mean these. Robustness is the capacity as a framework to keep up its capacities in a functioning state and endure when changes happen in inside architecture of the framework or outside condition. It might be possible that segments of a system are not strong enough themselves; however, at the system level, the connections between them make robustness. Without change in its architecture, a powerful framework can play out numerous functionalities for resistance.

4.2 Issues in Open Autonomic Personal Computing

4.2.1 Security

Security arrangements are notorious for being hard to utilize. Secure keys for the clients must be produced and circulated. Additionally, testaments for the keys ought to be produced and a database of revoked authentications. The current absence of convenience of security scheme is fundamentally retardant to their widespread acknowledgment.

4.2.2 Connectivity

There is such a significant number of choices to interface with individual, neighborhood, and wide-zone networks that the decision of selecting network overburden the clients. Each area has its own availability, attributes, and in this manner versatility additionally includes unpredictability. The dynamic correspondence connections of a gadget ought to be rearranged to the most suitable status each time any of the changes, where a proportion of appropriateness would be reliant on the nature of administration, cost, security, accessibility, area, and other policy components.

4.2.3 Storage

Autonomic capacity will begin with computerization of the storage management that clients perform today. Information is frequently put away in various areas and numerous forms, and it is too simple to even think about losing track of where the information is found. As data are moved and duplicated, huge protection and security [26–28] prerequisites must be met. This implies manual work, if not computerized, will probably hinder the required progression of such data. The principle challenge in storage is to extract and manage both the physical area of information and the protection and security necessities of the information. This deliberation ought to permit applications created without autonomic storage in mind to work typically, however, may not be ideal as an application intended for autonomic storage. With the improvement in autonomic storage, most of the entire of this administration function should get automatic, guided by more significant levels of the executives that implement more extensive business-based strategy.

4.2.4 Peer Group Collaboration

The compensation for peer group coordinated effort is access to additional and progressively current information that may not be accessible through increasingly formal publishing means. Peer computing is a unique instance of dispersed computing with a few difficulties. The primary difficulty is the means by which to shape a peer group. Since autonomic function regularly requests executions that do not include guidance of human, the peer group must be shaped naturally. A subsequent difficulty is recognizing the particular collaboration type for the peer group. On this, numerous analysts are studying about the solicitations to make general, with the goal that only a restricted set of reactions are structured, and sharp, so just important reactions are created. The third main difficulty is deciding the level of trust that any part places in the data received from some other part in the group. Credibility can be built up through a history of securing helpful and exact information from a part, yet if no appraisal of validity is accessible, the utilizations to which obtained data can be put must be restricted.

4.2.5 Network-Based Services

The opportunity for network-based administrations to supplement and expand services implemented locally and the peer group is excessively costly; however, a few models demonstrate its latent capacity. To supplement autonomic availability, network-based administrations can flexibly add data about resources accessible in the current area, for example, IT resources. Perhaps the key network-based service is the administration registry from which a menu of administrations and how to access them can be acquired. For the clients to grasp a customer management utility, they should be furnished with start to finish security that makes sure about their enterprise

information, combined with adequately incredible remote administration capacities to empower ongoing program assurance, determination, and goals without the intervention of the end clients.

4.2.6 User Interface

Numerous autonomic computing capacities have no end-user noticeable conduct. There is additionally the topic of how the estimation of autonomic processing innovation is seen when the activities that convey that value are hidden. Numerous analysts accept that autonomic individualized computing will experience a few phases of advancement, separated to some extent by how the end user is aware about and participates in the management activities. As autonomic conduct turns out to be increasingly viable, it will be confided in additional, and the requirement for an end client to take direct control will diminish. During this time, clients will probably be required to choose or affirm activities that might be proposed by an autonomic manager.

4.3 Challenges in Autonomic Benchmarking

4.3.1 Injecting Changes

There exist two principle challenges in injecting changes. The first is to guarantee that the benchmark comments are reproducible despite the change that is injected. It is important to guarantee that all the individual changes are injected reproducibly and must arrange the injected changes with the applied task at hand. For this benchmark to be valuable in cross-framework correlations, changes ought to be reproducible across various frameworks. The subsequent key challenge is about the representativeness of the injected changes. In contrast to a remaining task at hand, which can be simulated in confinement, natural changes may require an enormous scope for recreation of framework moving toward the complexity of a genuine arrangement.

4.3.2 Metrics and Scoring

Autonomic benchmarks should quantitatively capture four components of a system's autonomic reaction: the degree of the reaction, the nature of the reaction, the effect of the reaction on framework's clients, and the expense of additional assets expected to help the autonomic response. Different issues have been identified with measurements and scoring that include the challenge of incorporating numerous submetrics into a general proportion of autonomicity and aligning scores across various frameworks.

4.3.3 Handling Partially Autonomic Systems

Partially autonomic frameworks incorporate some autonomic capacities yet require some human administration association to adjust completely. An autonomic computing benchmark should give helpful metrics to the frameworks to evaluate the steps toward a completely autonomic system. Similarly, as clarified in the inclusion of human during the time spent, benchmark process is to be viewed as where it would assist with expanding to incorporate parts of human client studies, with measurable methods used to give reproducibility. An elective methodology is to break the benchmark into discrete stages to such an extent that human intervention is just required between stages, and each stage would then be scored exclusively with a penalty applied by the measure of interphase human help required.

4.4 Autonomic Computing Research Problems

4.4.1 Conceptual

Conceptual research issues and challenges incorporate various things such as the characterizations of deliberations and structures for indicating, understanding, controlling, and executing autonomic practices; adapting the old structures and knowledge for machine learning, enhancement, and control of dynamic and multi-agent system; giving viable structures for the arrangement of autonomic components that could be used to build multilateral relationships among them; and planning measurable structures of large-scale connected systems that can help autonomic systems predict the general issues from a surge of sensor information from individual devices.

4.4.2 Architecture

Autonomic applications and systems are developed from autonomic components that deal with its inside conduct and the connections with other autonomic components per the arrangements set by humans or other components. Using this, the application or system will arise a self-managing behavior from the existing behaviors of the constituent autonomic components and their collaborations. The key research includes such systems and programming models in which the local and worldwide autonomic behaviors could be indicated, implemented, and also controlled in vigorous ways.

4.4.3 Middleware

The essential middleware-level research challenge gives the center administrations that need to be acknowledged by autonomic practices in a powerful, robust, and adaptable way, despite the dynamic nature and vulnerability of the framework and application. It incorporates discovery, informing, security, protection, and so on that is required by the autonomic components to recognize themselves and verify the identities of different elements of interest. The middleware should also be safe enough, dependable, and robust against the new and internal attacks which utilize self-administration depending on strategies of significant level for their own potential benefit.

4.4.4 Application

The main difficulties at the level of application are the development and improvement of frameworks and applications which are equipped for managing themselves. This incorporates software models, structures, and middleware administrations that help to build the autonomic applications such as dynamic and opportunistic component and the strategy, content, execution, and board of these applications.

4.5 Technology Transfer Issues of Autonomic Computing

4.5.1 Trust

Regardless of the groups that are managing to incorporate the right technology, the trust of the client may become a problem as far as giving over the control to the system. AI and autonomous agent domains have experienced this issue. It often happens that the rule-based systems get more priority over the neural networks and uncertainty in AI techniques, just because the client can follow and comprehend.

4.5.2 Economics

New models have to be planned, as the autonomicity may infer another property of self-centeredness. Any autonomic element will perform an operation in a specific environment after incurring a personal cost.

4.5.3 Standards

The broad vision of autonomic computing can be achieved through various standards. Especially, for the communication between AEs, standards are required. Also, at the same time, there should be other light ways to define them.

4.6 Open Problems in Autonomic Computing

4.6.1 How to Select the Right Formalism and Techniques?

There is a tremendous scope of formalisms and strategies that could be utilized for structuring, creating, testing, and keeping up autonomic frameworks. Everyone has their own preferences and weakness. Thus, recognize what is progressively fitting for which action, and how various formalisms and methods can be consolidated.

4.6.2 How to Incorporate Autonomic Behavior to Non-Autonomous or Semi-Autonomous Systems?

It is for all intents and purposes difficult to incorporate checking and activating functions because of the issues in frameworks in which the source code is not accessible or the coupling between the parts is large. Autonomic attributes are likely identified with quality and non-function requirements (NFR). Also, it gets imperative to arrange autonomic attributes as indicated by their quantifiable impact on the inside quality measurements and connect them to the quality components outside to which they are corresponded.

4.6.3 How to Study and Administer the Dependencies between Autonomous Elements to Address Business Policies?

The most fundamental point on the vision of autonomic processing is the structuring of interfaces for making an interpretation of business strategies into IT approaches. This element is identified with ease of use, and it is essential to figure out which segments are mutually independent.

4.6.4 How to Make More Open/Extensible Autonomic Tools?

Absence of open principles is one more difficulty for flourishing autonomic segments and frameworks in IT industry. Interpretability and applying worldwide approaches for security and setup, all rely upon presence of such principles.

4.6.5 What Are the Major and Minor Characteristics for the Evaluation of Autonomic Systems?

A system should be created for crossing over the holes between the attributes and the quality variables. Quality affirmation in autonomic frameworks is a zone that is still in its earliest stages.

5 Automatic Virtual Resource Management in Cloud

Independent applications are hosted on a shared resource pool which allocates computing capabilities to the applications on pay-per-usage fundament. This is possible only with the consolidation of several Virtual Machines (VMs) on the same physical servers. When necessary, the VMs are resized, and they may migrate to other Physical Machines (PMs). The major challenge for the cloud service providers is how to automate the virtual machines and manage them automatically while considering the QoS parameters of the hosted applications and costs for managing the resources. The autonomic resource management tool controls the virtualized environment which in turn decouples the provisioning of resources from the dynamic placement of VMs. The aim of automatic manager is to fulfill SLA and minimize the operating cost by optimizing the global utility function. A Constraint Programming approach can be applied to formulate and solve the optimization problem.

The automatic resource management system in cloud platform fulfills the following requirements:

- Automatize the dynamic provisioning plus place the VMs by considering SLAs as well as resource management costs.
- Support the heterogeneous applications as well as workloads, which include online applications with QoS parameters and batch applications.
- Support different topologies such as single cluster, monolithic, n-tier including capacity to scale by including extra resource to a single server or including more servers to the system.

As shown in Fig. 4, two levels are managed in the cloud infrastructure; in the provisioning stage, extra resource is added to the application in the form of VMs. VMs can then be mapped and consolidated to physical servers. The problem of VM placement is compelled by the policies governed by the policies of data center based on resource management costs. There is a need for policy to minimize the energy consumption through minimum active PMs.

The provisioning of VM can be separated from the VM placement, and Constraint Satisfaction Problems (CSP) can be formulated for both the problems. Both problems can be formulated as knapsack problem, and Constraint Programming approach problem is a good fit to solve them. Both are the NP-hard problems.

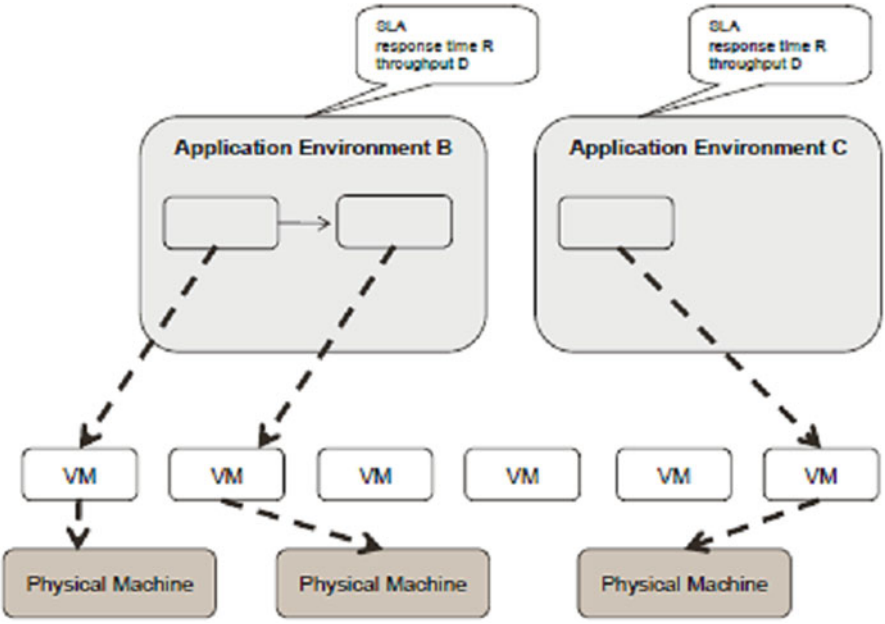

Fig. 4 Two levels of mapping in cloud infrastructure

The general idea of Constraint Programming is solving a problem through building relations among the constraint variables so that it satisfies the solution.

5.1 System Architecture

The system design for resource management in cloud is represented in Fig. 5. The data center has a set of PMs each facilitating a few VMs with the assistance of a hypervisor. It is assumed that that the PMs are fixed in number, and all of them belong to a similar cluster with the likelihood to perform a live relocation of a VM between any two PMs. Only one autonomic element is related to a running VM. Applications cannot demand a VM with an arbitrary resource ability regarding the power of CPU and its memory size. The VMs accessible to the application must be picked among a lot of precharacterized VM classes. Every class of VM comes along with a particular CPU and memory limit. An application-explicit Local Decision Module (LDM) is related to each AE, and every LDM gets the chance to allocate more VMs or release the existing VMs to/from the autonomic element based on the current outstanding task at hand utilizing administration-level measurements originating from application-explicit monitoring tests. The primary employment of the LDM is to register a utility capacity, which gives a proportion of application fulfillment with a particular resource assignment along with its

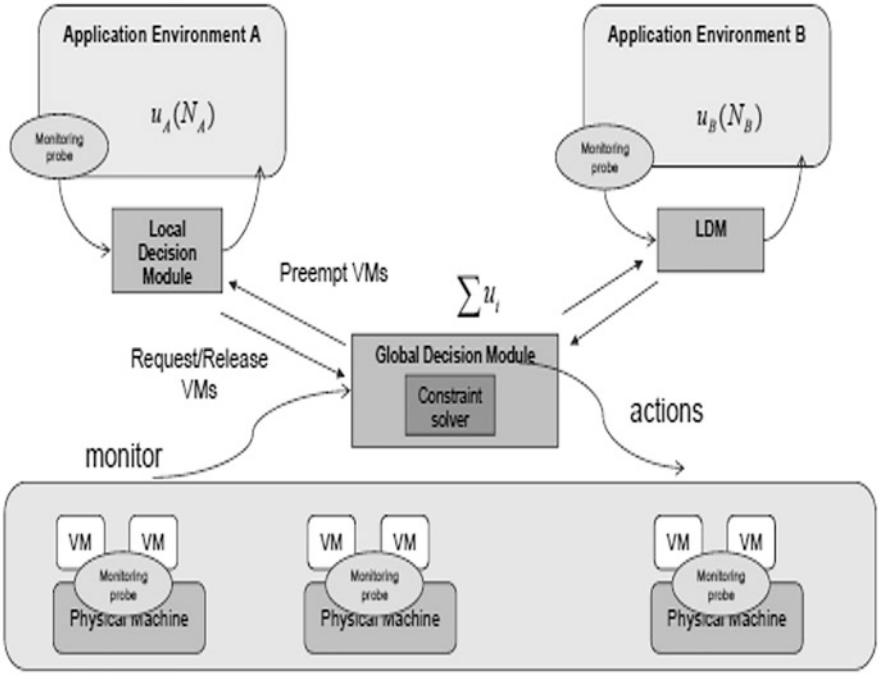

Fig. 5 Management system architecture

present workload and SLA objective. LDMs collaborate with a Global Decision Module (GDM) which is considered to be the element that makes decision inside an autonomic control loop. The GDM is answerable for referring resource prerequisites originating from each AE and treats all LDM as a black box without monitoring the idea of the application or the manner in which LDM processes its utility capacity. As input, GDM receives the utility abilities from each LDM, and framework-level measurements for execution from both the virtual and physical servers. The yield of the GDM comprises of the administration activities coordinated to the server hypervisor and the transfer of notifications to LDM. The latter tells the LDM if another VM with specific resource limit has been dispensed to the allocation, any upgradation or downgradation in the current VMs, and if a VM having a place with the application is being preempted such that, the application ought to relinquish it immediately. The administration activities incorporate the lifecycle management of a VM and the trigger of a live movement of a running VM, the latter activity being transparent as far as the host applications are considered.

The inner principle of LDM and GDM can be described as follows:

Let $A = <AE_1, AE_2, \ldots AE_n>$ the set of AE's and $P = <PM_1, PM_2, \ldots .PM_n>$ the set of PM's There are c classes of VM's available with set $S = <s_1, s_2, \ldots, s_c>$ where $s_i =< s_i^{cpu}, s_i^{ram} >$ denoted the CPU capacity and the memory capacity of the VM in MHz and MB respectively.

5.2 The LDM

The LDM is assigned to two utility function: a mapping from the service level to the utility value called the fixed-service level utility function and the other a mapping from the resource capability with a utility measure called the resource-level utility function. The latter one is communicated in each iteration to the GDM using automatic control loop. The latter function U_i for AE_i is defined as $U_i = f(N_i)$ where N_i = VM allocation vector of AE_i : $N_i = < n_{i1}, n_{i2}, \ldots n_{im}>$ where n_{im} denotes the VMs of class s_m related to AE_i.

The constraints can be expressed as follows:

$$n_{ik} <= n_{ik}^{max} \quad i \in [1, m] \text{ and } k \in [1, c] \tag{1}$$

$$\sum_{k=1}^{c} n_{ik} <= T_i^{max} \quad i \in [1, m] \tag{2}$$

Each AE provides upper bounds on VM number on every class $N_i^{max} = \left(n_{i1}^{max}, n_{i1}^{max}, \ldots n_{ik}^{max}, \ldots n_{im}^{max}\right)$ and the total number of VM i.e., T_i^{max} ready to accept.

5.3 The GDM

The GDM does two tasks: For each AE_i determines the VM allocation vectors N_i known as VM provisioning, and then placing these VMs in PMs and minimizes the active PMs known as PM packaging. The abovementioned phases can be described in terms of two constraint satisfaction problems and that can be handled by constraint solver as shown in Fig. 6.

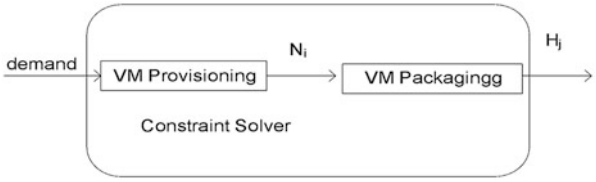

Fig. 6 CSP: constraint solving

5.3.1 VM Provisioning

Global utility value $global_u$ has to be maximized by finding the VM allocation vectors N_i for each AE_i

$$\begin{aligned} \sum_{i=1}^{m} \sum_{k=1}^{c} n_{ik}.s_k^{cpu} <= \sum_{j=1}^{q} C_j^{cpu} \\ \sum_{i=1}^{m} \sum_{k=1}^{c} n_{ik}.s_k^{ram} <= \sum_{j=1}^{q} C_j^{ram} \end{aligned} \tag{3}$$

Here CPU and ram capacities of PM p_j are denoted by C_j^{cpu} and C_j^{ram} respectively.

$$global_u = \text{maximize} \sum_{i=1}^{m} (\alpha_i.u_i - \varepsilon.\cos t\,(N_i)) \tag{4}$$

5.3.2 VM Packaging

This phase takes the input the VM allocation vector N_i and outputs a single vector $V = < vm_1, vm_2, vm_3, \ldots, vm_v >$ the VM's running at the current time.

The physical resource constraint is expressed as:

$$\begin{aligned} \sum_{l=1}^{v} r_l^{cpu}.h_{jl} <= C_j^{cpu} \quad j \in \{1, 2, 3 \ldots, q\} \\ \sum_{l=1}^{v} r_l^{ram}.h_{jl} <= C_j^{ram} \quad j \in \{1, 2, 3 \ldots, q\} \end{aligned} \tag{5}$$

Our aim is the minimization of the active PMs X:

$$X = \sum_{j=1}^{q} u_j,\ u_j = \begin{cases} 1 \text{ there exits } vm_l \in V \text{ such that } h_{jl} = 1 \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

Solving the equations gives us the VM placement vector H_j those are used for placing the VMs on PMs. As the GDM runs on periodic basis it finds the VMs needs migration. The VM migration cost is directly proportional to the sum of allocated memory to the VM.

6 Conclusion

Autonomic computing is concerned with moving the weight of self-managing frameworks from individuals to innovations. At the point when self-managing abilities are conveyed by IBM and different merchants can team up, the components of a complex IT framework can cooperate and oversee themselves depending upon a common perspective of system-wide approach and targets. Autonomic processing is a big test that requires progress in fields of science and innovation, especially frameworks, software design and building, strategy, enhancement, and numerous parts of man-made reasoning, such as planning, learning, information portrayal and thinking, negotiation, and new conduct. Coordinating these innovations and embedding them in a suitable framework design to accomplish the ideal self-managing properties is an exploration challenge in itself.

The mission to achieve the autonomic frameworks was begun late. The dynamic vision will comprise a progression of improvement for framework and programming structuring alongside the participation in various fields. Early R&D introduced features that tend to gain vitality in all perspectives to reach at the objective. The difficulties in reconstructing the current frameworks to tomorrow's inescapable and ubiquitous calculation and correspondence will require binding together of norms, new monetary systems, and certainty of the customers, similar to advancements for tending to knock out specialized outcomes. It has been said that in Autonomic Computing's underlying organization take-up, numerous researchers and designers have focused on self-advancement as it is simpler to convert into dollars. Basically, this emphasis on improvement from the four self-qualities might be considered as running contrary to the original order of things where the innovation is driving us to quicker machines, but such fine enhancement is certainly not a matter of concern. For Autonomic Computing to prevail in the long term, all of its attributes must be attended to similarly and in a coordinated manner. Apart from addressing the complexity issues, it additionally assures a lower expense of ownership and a diminished support load. Achieving the vision will probably set generous expectations on the budget plans in the present moment as autonomic function and conduct are structured into frameworks.

It is crucial to move toward a definitive vision of autonomic computing, and maintain the exploration fair by building models that quantifiably show a consistently expanding capacity for self-administration. In such a manner, it is important to build up an open platform autonomic computing that will fill as a core for AC prototype.

References

1. Kandasamy, N., Abdelwahed, S., & Hayes, J. P. (2004, May). Self-optimization in computer systems via on-line control: Application to power management. In International conference on autonomic computing, 2004. Proceedings (pp. 54–61). IEEE.
2. Khalid, A., Haye, M. A., Khan, M. J., & Shamail, S. (2009, April). Survey of frameworks, architectures and techniques in autonomic computing. In 2009 fifth international conference on autonomic and autonomous systems (pp. 220–225). IEEE.
3. Samaan, N., & Karmouch, A. (2009). Towards autonomic network management: An analysis of current and future research directions. *IEEE Communications Surveys and Tutorials, 11*(3), 22–36.
4. McCann, J. A., & Huebscher, M. C. (2004, October). Evaluation issues in autonomic computing. In *International conference on grid and cooperative computing* (pp. 597–608). Berlin: Springer.
5. De Wolf, T., & Holvoet, T. (2006). Evaluation and comparison of decentralised autonomic computing systems. CW reports (p. 10).
6. Brown, A. B., Hellerstein, J., Hogstrom, M., Lau, T., Lightstone, S., Shum, P., & Yost, M. P. (2004, May). Benchmarking autonomic capabilities: Promises and pitfalls. In International conference on autonomic computing, 2004. Proceedings (pp. 266–267). IEEE.
7. Smart, IBM. Retrieved from http://www.almaden.ibm.com/software/dm/SMART/
8. AutoAdmin, Microsoft Corporation. Retrieved from http://research.microsoft.com/dmx/autoadmin/
9. Murch, R. (2004). *Autonomic Computing*. Upper Saddle River, MJ: Prentice Hall.
10. Oceanstore, UC Berkeley. Retrieved from http://oceanstore.cs.berkeley.edu/
11. Anthill, University of bologna. Retrieved from http://www.cs.unibo.it/projects/anthill/
12. Patterson, D., Brown, A., Broadwell, P., Candea, G., Chen, M., Cutler, J., et al. (2002). Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies (pp. 1–25). Technical report UCB//CSD-02-1175, UC Berkeley Computer Science.
13. Ebiquity, University of Baltimore County. Retrieved from http://ebiquity.umbc.edu
14. Autonomic Computing, the 8 elements. Retrieved from http://www.research.ibm.com/autonomic/overview
15. IBM. (2003). *IBM Systems Journal Special Issue on Autonomic Computing, 42*(1), 197.
16. Norman, D. A., Ortony, A., & Russell, D. M. (2003). Affect and machine design: Lessons for the development of autonomous machines. *IBM Systems Journal, 42*(1), 38–44.
17. N1, Sun Microsystems. Retrieved from http://www.sun.com/software/n1gridsystem/
18. Sloman, A. (1997). Review of: Rosalind Picard's affective computing. *AI Magazine, 20*, 127–137.
19. Lau, T., Oblinger, D., Bergman, L., Castelli, V., & Anderson, C. (2003, August). Learning procedures for autonomic computing. In IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico, 10th August.
20. IBM Tivoli Monitoring. IBM Corporation. Retrieved from http://www.tivoli.com/products/index/monitor/
21. Lanfranchi, G., Della Peruta, P., Perrone, A., & Calvanese, D. (2003). Toward a new landscape of systems management in an autonomic computing environment. *IBM Systems Journal, 42*(1), 119–128.
22. Guo, H. (2003). A Bayesian approach for automatic algorithm selection. In Proceedings of the international joint conference on artificial intelligence (IJCAI03), workshop on AI and autonomic computing, Acapulco, Mexico (pp. 1–5).
23. Aiber, S., Etzion, O., & Wasserkrug, S. (2003, August). The utilization of AI techniques in the autonomic monitoring and optimization of business objectives. In IJCAI workshop on AI and autonomic computing: developing a research agenda for self-managing computer systems, Acapulco, Mexico, 10th August.

24. Kaiser, G., Parekh, J., Gross, P., & Valetto, G. (2003, June). Kinesthetics extreme: An external infrastructure for monitoring distributed legacy systems. In 2003 autonomic computing workshop (pp. 22-30). IEEE.
25. Birman, K. P., Van Renesse, R., & Vogels, W. (2003, June). Navigating in the storm: Using astrolabe for distributed self-configuration, monitoring and adaptation. In 2003 autonomic computing workshop (pp. 4–13). IEEE.
26. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*, 1144–1148.
27. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
28. Singh, B. K., Alemu, D. P. S. M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.

Classification of Various Scheduling Approaches for Resource Management System in Cloud Computing

Ajay Jangra, Neeraj Mangla, Anurag Jain, Bhupesh Kumar Dewangan, and Thinagaran Perumal

1 Introduction

Resource administration is an umbrella activity including different periods of resources and exceptional jobs needing to be done from residual weight convenience to outstanding job needing to be done execution. Resource administration in the cloud fuses two stages: resource scheduling and resource arranging. Resource scheduling is described to be the stage to perceive palatable resources for a given residual job that needs to be done reliant on QoS essentials depicted by cloud customers; however, resource booking is arranging and execution of cloud client remarkable jobs that need to be done subject to picked resources through resource

A. Jangra
Department of Computer Science and Engineering, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

N. Mangla
Computer Science Department, MMEC, Maharishi Markandeshwar Deemed to be University, Ambala, India

A. Jain (✉)
Virtualization Department, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India
e-mail: anurag.jain@ddn.upes.ac.in

B. K. Dewangan
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

T. Perumal
Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra, Seri Kembangan, Malaysia
e-mail: thinagaran@upm.edu.my

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_8

scheduling. Directly off the bat, cloud customer submits interest for the residual job that needs to be executed as extraordinary weight nuances. Considering these nuances, master (resource Scheduling) discovers the appropriate resources for a specified extraordinary weight and chooses the credibility of the scheduling of resources subject to QoS essentials [1]. The operator sends sales to the resource scheduler for arranging after-productive scheduling of resources. Various obligations of a middle person include the appearance of extra resources to the resource group, data of provisioned resources (PR), and screen execution to incorporate or oust resources. Following resource scheduling, resource booking is done in the second stage. The PR is reserved in the resource line, while other outstanding resources are in the resource group [2]. Submitted remaining jobs that need to be done are set up in an exceptional main job line. Taking into account the QoS necessities, arranging resources for acceptable extraordinary jobs needs to be done which is a troublesome issue [3]. There is a necessity to uncover the investigation confronts in resource intending to carry out the rest of the weights without impacting the diverse QoS essentials.

Resource booking is a hotspot branch of assessment in the cloud [4, 5] in light of gigantic resource cost and execution time. Particular resource arranging models and limits are facilitated to different classes of resource scheduling algorithms (RSAs). The first period of resource administration is resource scheduling has been analyzed in our past study paper [4]. This investigation work discusses the second period of resource management for instance resource booking. Convincing resource arranging decreases the use of essential, execution time, and execution cost, considering diverse QoS necessities such as relentless quality and adaptability. Both the social affairs have different necessities: the provider needs to procure many advantages as could be normal with the least hypothesis and enhance the utilization of resources, while the customer needs to execute workload(s) with the least execution time and cost. In any case, carrying out several residual weights on one resource will cause hindrance among extraordinary weights which prompt dreary appearing and diminish customer reliability. To keep up the organization's quality, the providers excuse the sales that achieve flighty conditions [4]. The providers similarly consider erratic resources for booking and carrying out the rest of the jobs that need to be done. Arrangement of resources ends up in more testing because both customers and providers are not prepared to grant data to each other. The confronts of resource arranging fuse dispersing, weaknesses, and similarities of resources that are not settled with standard RSAs in cloud conditions [5]. Thus, there is a requirement to execute cloud sremaining weights compelled to manage these properties of the cloud condition.

System selector is used for picking the best possible arranging technique reliant on an extraordinary job that needs to be done with nuances depicted by a cloud purchaser [6]. Cloud condition and an RS that completes an assorted booking game plan are subject to the decision taken by the methodology selector. Taking into account the arranging system, the resources are circulated to the cloud exceptional jobs that need to be done.

The resource scheduler designs moving toward the cloud, with main jobs subject to the exceptional weight nuances. Above all, get cloud remaining jobs that need to be done to schedule and a short time later discover fitting and open resources and cloud remarkable main jobs arranged profitably reliant on the booking systems. The dispatcher is used to dispatch the rest of the jobs for execution. The rest of the job is dispatched just if the extraordinary weights will be carried out by the QoS limits referenced in SLA. Resource screen is utilized to verify the position of booking of resources similar to whether the important number of resources is given or not. QoS screen includes the data concerning QoS limits to confirm whether all the exceptional jobs that need to be done are carried out inside their foreordained variety or not. Deadline time is a QoS limit, so the QoS screen must check whether the exceptional main jobs are carried out before the deadline time. There is an encroachment of SLA if a remarkable job is executed after the deadline time [7–9].

1.1 Re-Orientation Motivation

Resource-making arrangements for cloud [12] is a method of the dynamic designation of resources to cloud exceptional jobs that need to be done after resource scheduling. Along these lines, this assessment complements resource arranging counts reliant on different booking measures.

We saw the need for systematic composing concentrate in the wake of considering dynamic assessment in resource getting ready for cloud computing. Therefore, we have summarized the open assessment reliant on an extensive and exact chase in the existing record and show the investigation confronts for forefront research.

2 Background

Cloud takes after a significant revelation, nothing within the cloud is observable to the cloud buyers. Cloud passes on computing as a function that is accessible to the cloud customers on requirements [10–12].

2.1 Coming Up Next Is, for the Most Part, Referred to Implications of Cloud Computing

1. *NIST*: Cloud computing is a representation of connecting all-inclusive, accommodating, on-demand sort out admittance to a common cluster of configurable resources.

2. *Rajkumar Buyya*: A cloud is an equivalent and circled structure including a grouping of related and virtualized PCs that are continuously presented and provisioned as at any rate one bound together computing resource(s) considering organization-level understanding set up through the course of action amid the authority community and clients [13–15].

Cloud computing is consists of three kinds of organization models. These organization models are true to the form and significance of the organizations given by cloud computing [16, 17].

3 Classification of Cloud Resources

Cloud computing gives a phase where resources are charged with the help of its cloud customers/cloud clients through the Internet. Cloud passes on computing as a service as it is accessible to the cloud purchasers on requirements. Several researchers have gathered many resources such as steady resources and physical resources or hardware resources and programming resources[18–20]. Cloud computing acts as utility-based computing as follows:

1. *Storage Utility*: Instead of storing data at neighborhood amassing contraption, we store them at a limit device that is arranged in a faraway spot. Limit utility involves many hard drives, streak drives, database laborers, etc. PC systems will without a doubt misfire over the period data reiteration is required here. On account of the cloud's time, a variety of organization models storing utility needs to give features such as cloud adaptability. Through limit, utility cloud computing gives Storage as a Service (StaaS) [21].
2. *Communication Utility*: It is named Network as a Service (NaaS) or Network Utility. Speedy limit utility and estimation service cannot be thought of without correspondence utility. Correspondence utility involves physical (temporary contraptions, has, sensors, physical correspondence interface) and rational (bandwidth, delay, shows, virtual correspondence associate) resources. In cloud computing, every help is given through a quick Internet. Bandwidth and deferment are commonly critical from sort out point of view.
3. *Power/Energy Utility*: Imperativeness cost can extraordinarily be reduced by utilizing power careful methods. Due to a lot of data laborers, the power use is especially high in cloud computing. UPS and cooling equipment are present at the point of convergence of such resources. They can moreover be considered as assistant resources [22, 23].
4. *Security Utility*: It is reliably a critical topic in any computing zone. Being cloud customers, we need particularly strong and secure cloud and trust-competent organizations [24, 25].

4 Literature Survey

Singh & Chana [26] introduced the Power-Aware Load Balancing Algorithm (PALB) for IaaS cloud. Creators had structured calculations in three portions: (1) balancing segment; (2) upscale segment; and (3) downscale segment. PALB keeps the position of all process nodes, and depending on its utilization, they choose the number of utilitarian register nodes.

Singh & Chana [27] proposed a showcase-driven resource allotment procedure. Creators created a discrete occasion-based VM scheduler for resource administration. Creators utilized a single supplier situation for spot case administration given by Amazon EC2. After performing the assessment, the creators guarantee that normal solicitation holding up time is decreased.

Xu & Li [28] had proposed a strategy for the production planning of resource demands onto a mutual substrate interconnecting different computing resources and embrace a heuristic approach to address the issue.

Yuan & Liu [29] had projected Combinatorial Double Auction Resource Allocation (CDARA). CDARA comprises of seven correspondence stages: (1) resource and advertising revelation; (2) generate packs; (3) informing the finish of sale; (4) winner assurance; (5) resource designation; (6) pricing model; and (7) allocation of undertaking and installment. Creators utilized CloudSim for reproduction in the cloud. It is a closeout-based model.

Aslanpour et al. [30] proposed a force and burden-mindful resource distribution strategy for half breed cloud. Creators attempted to limit power utilization and augment the usage of resources. Creators have created two separate calculations: (1) resource statement and (2) resource portion. Creators tried their calculations with a DVFS-based planning strategy.

Liu and Buyya [31] had a projected resource observing model for VM in cloud computing. Creators had checked dynamic and static data of live working nodes for future resource revelation and resource allotment models. Usage was completed utilizing Java and C/C++ language.

Madni et al. [32] had projected checking to engineer for cloud computing. To accomplish this, creators had incorporated the resource observing apparatus and its resource revelation convention. Usage of the equivalent is done in Java.

Harki et al. [33] had significantly centered around unique resource estimating in cloud computing. Creators guarantee that a powerful valuing strategy is consistently ready to adjust the number of fruitful solicitations and the quantity of distributed resources relying on the economic situation. Hence, it accomplishes better economic proficiency.

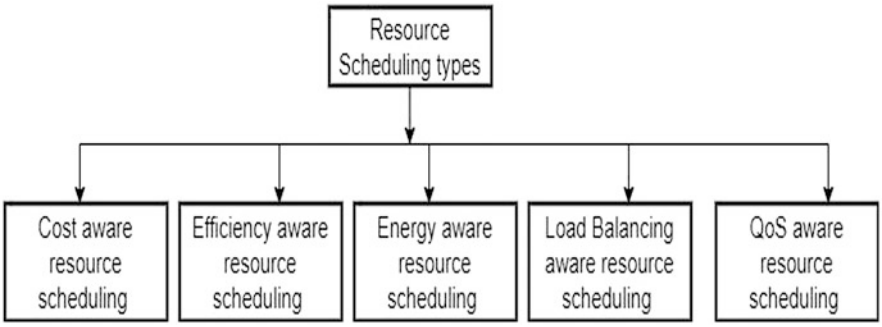

Fig. 1 Categorization of resource scheduling in cloud computing

5 Classification of Resource Scheduling

The RS plans are arranged into six mixture classifications, including cost-mindful RS, effectiveness mindful RS, vitality mindful RS, load-adjusting mindful RS, QoS mindful resource scheduling, and use mindful RS as appeared in Fig. 1. The motivation behind these characterizations is to construct the reason for future analysts in distributed computing conditions. This arrangement depends on the boundary used in the assessment of the presentation in different examinations for RS.

The primary grouping centers around cost-mindful resource scheduling that incorporates cloud suppliers' income and benefit, the value of resources, clients' use, and all-out expense. The second class centers around effectiveness mindful resource scheduling, which upgrades the exhibition by including need, diminishing the implementation time, makespan, and implementation cost, likewise expanding the transfer speed. The third kind presents the vitality-mindful RS that explains limiting the force and vitality utilization in the server farms. The fourth type explains the heap adjusting mindful resource scheduling by effectively dealing with the outstanding task at hand of numerous clients among various server farms. The fifth class presents QoS-mindful resource scheduling that manages the arrangements with unwavering quality, accessibility, SLA, adaptation to internal failure, throughput, and recuperation time. At last, the sixth classification manages to use mindful resource scheduling. It centers around effectively boosting the utilization.

- *Cost-aware Resource Scheduling* [31].

  Cost-aware resource scheduling plays a significant part in distributed computing, as the meaning of cloud proclaims that it conveys the administrations in the least expensive sums. A cloud supplier is answerable for conveying the clients' requests as a help, which brings about supplier income, benefit, and client consumptions.

- *Effectiveness-aware Resource Scheduling.*
  Effectiveness-aware resource scheduling communicates the measure of resources consumed for handling, focusing on the resources to upgrade efficiency.
- *Energy-aware Resource Scheduling.*
  Energy-aware resource scheduling procedures are needed to defeat the problems that arise because of high vitality utilization in the server farms. Under haze processing, decreasing vitality utilization and sparing the costs because of vitality are generous for the server farms and cloud suppliers. Information is expanding so quickly that continuously bigger workers and plates are expected to handle them rapidly inside the obligatory time frame. The loss or wastage of inert force is a significant reason for vitality insufficiency. Green registering is favorable to accomplish the productive preparing and usage of resources by smaller than usual mixing the vitality utilization.
- *Load Balancing–aware Resource Scheduling.*
  Load balancing is an attainable cycle that improves VMs and server farms overstacked with registering cloudlets, errands, or occupations through sharing burdens across server farm foundations to accomplish a capable execution of the frameworks. Efficient allotment and scheduling must guarantee that resources are effectively accessible on request and capably used under the state of high/low burden by sparing energy and cost.
- *QoS-aware Resource Scheduling.*
  QoS-aware resource scheduling is a central point of interest in distributed computing. It infers to plan effectively and requests assignment of clients to various resources as indicated by the QoS, which centers around accessibility, dependability, throughput, recuperation time, adaptation to non-critical failure, and SLA of both cloud supplier and clients. At the hour of scheduling of the resources, the QoS for the mentioned request ought not to diminish.

6 Conclusion

Cloud computing empowers cloud resources to be utilized as a service. By breaking down cloud computing for resource administration, this exploration work previously centered around ordering cloud resources. The scientific categorization of cloud resource administration was introduced with the goal that different exploration issues identified with resource management can be recognized depending on different stages referenced in this paper. In conclusion, different exploration works were inspected for recognizing research problems in cloud resource administration. This paper presents resource administration in cloud computing as a successive procedure for different strategies with their examination issues. This exploration paper additionally reasons that proficient cloud resource administration should meet models that resemble effective utilization of resources, cost decrease from the cloud suppliers' point of view, and vitality/power decrease. There can be a few future headings for this examination of work. One of the things to come to work is to

recognize different strategies of resource portion/reallocation through multi-target improvement procedures. Besides, novel upgraded procedures must be planned that ought to oblige previously mentioned standards.

References

1. Bittencourt, L. F., Diaz-Montes, J., Buyya, R., Rana, O. F., & Parashar, M. (2017). Mobility-aware application scheduling in fog computing. *IEEE Cloud Computing, 4*(2), 26–35.
2. Bittencourt, L. F., Goldman, A., Madeira, E. R. M., da Fonseca, N. L. S., & Sakellariou, R. (2018). Scheduling in distributed systems: A cloud computing perspective. *Computer Science Review, 30*, 31–54.
3. Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems, 25*(6), 599–616.
4. Gill, S. S., Garraghan, P., & Buyya, R. (2019). ROUTER: Fog enabled cloud based intelligent resource management approach for smart home IoT devices. *Journal of Systems and Software, 154*, 125–138.
5. Jain, A. & Kumar, R. A comparative analysis of task scheduling approaches for cloud environment. 2016 3rd international conference on computing for sustainable global development (INDIACom) (pp. 1787–1792). IEEE.
6. Liaqat, M., Chang, V., Gani, A., Hamid, S. H. A., Toseef, M., Shoaib, U., & Ali, R. L. (2017). Federated cloud resource management: Review and discussion. *Journal of Network and Computer Applications, 77*, 87–105.
7. Ge, J., Zhang, B., & Fang, Y. (2010). Research on the resource monitoring model under cloud computing environment. In *Web information systems and mining* (pp. 111–118). Berlin: Springer.
8. Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Energy-aware autonomic resource scheduling framework for cloud. *International Journal of Mathematical, Engineering and Management Sciences, 4*(1), 41–55. https://doi.org/10.33889/IJMEMS.2019.4.1-004.
9. Ghobaei-Arani, M., Souri, A., & Rahmanian, A. A. (2020). Resource management approaches in fog computing: A comprehensive review. *Journal of Grid Computing, 18*, 1–42.
10. Gutierrez-Aguado, J., Calero, J. M. A., & Villanueva, W. D. (2016). IaaSMon: Monitoring architecture for public cloud computing data centers. *Journal of Grid Computing, 14*, 283–297.
11. Haghighi, M. A., Maeen, M., & Haghparast, M. (2019). An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms. *Wireless Personal Communications, 104*(4), 1367–1391.
12. Jain, A., & Kumar, R. Critical analysis of load balancing strategies for cloud environment. *International Journal of Communication Networks and Distributed Systems, 18*(3–4), 213–234.
13. Jha, R. S., & Gupta, P. (2016). Power & load aware resource allocation policy for hybrid cloud. *Procedia Computer Science, 78*, 350–357.
14. Manvi, S. S., & Shyam, G. K. (2014). Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of Network and Computer Applications, 41*, 424–440.
15. Mell, P., & Grance, T.. (2011). The NIST definition of cloud computing.
16. Mihailescu, M., & Teo, Y. M. (2010). Dynamic resource pricing on federated clouds. In Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing (pp. 513–517). IEEE Computer Society.

17. Mohamaddiah, M. H., Abdullah, A., Subramaniam, S., & Hussin, M. (2014). A survey on resource allocation and monitoring in cloud computing. *International Journal of Machine Learning and Computing, 4*(1), 31.
18. Mustafa, S., Nazir, B., Hayat, A., Madani, S. A., et al. (2015). Resource management in cloud computing: Taxonomy, prospects, and challenges. *Computers and Electrical Engineering, 47*, 186–203.
19. Jain, A., & Kumar, R. (2014). A taxonomy of cloud computing. *International Journal of Scientific and Research Publications, 4*(7), 1–5.
20. Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervello-Pastor, C., & Monje, A. (2013). On the optimal allocation of virtual resources in cloud computing networks. *IEEE Transactions on Computers, 62*(6), 1060–1071.
21. Jain, A., & Kumar, R. Scalable and trustworthy load balancing technique for cloud environment. *International Journal of Engineering and Technology, 8*(2), 1245–1251.
22. Daramola, Olawande, and Darren Thebus. 2020. "Architecture-centric evaluation of blockchain-based smart contract E-voting for national elections." Informatics, 7, no. 2, p. 16. Multidisciplinary Digital Publishing Institute.
23. Sadashiv, N., & Kumar, S. D. (2011). Cluster, grid and cloud computing: A detailed comparison. In 2011 6th international conference on computer science & education (ICCSE) (pp. 477–482). IEEE.
24. Samimi, P., Teimouri, Y., & Mukhtar, M. (2014). A combinatorial double auction resource allocation model in cloud computing. *Information Sciences*.
25. Rasedur, M. D., Chakraborty, P., Zahidur, M. D., & Golam, M. D. (2019). Hiding confidential file using audio steganography. *International Journal of Computer Applications, 178*(50), 30–35. https://doi.org/10.5120/ijca2019919422.
26. Singh, S., & Chana, I. (2015). Qos-aware autonomic resource management in cloud computing: A systematic review. *ACM Computing Surveys (CSUR), 48*(3), 42.
27. Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing, 14*(2), 217–264.
28. Xu, L., & Li, J. (2016). Building efficient resource management systems in the cloud: Opportunities and challenges. *International Journal of Grid and Distributed Computing, 9*(3), 157–172.
29. Yuan, Y., & Liu, W.-C. (2011). Efficient resource management for cloud computing. In 2011 international conference on system science, engineering design and manufacturing informatization (ICSEM) (Vol. 2, pp. 233–236). IEEE.
30. Aslanpour, M. S., Gill, S. S., & Toosi, A. N. (2020). Performance evaluation metrics for cloud, fog and edge computing: A review, taxonomy, benchmarks and standards for future research. *Internet of Things*. 100273.
31. Liu, X., & Buyya, R. (2020). Resource management and scheduling in distributed stream processing systems: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR), 53*(3), 1–41.
32. Madni, S. H. H., Latiff, M. S. A., & Coulibaly, Y. (2016). Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. *Journal of Network and Computer Applications, 68*, 173–200.
33. Harki, N., Ahmed, A., & Haji, L. (2020). CPU scheduling techniques: A review on novel approaches strategy and performance assessment. *Journal of Applied Science and Technology Trends, 1*(2), 48–55.

Optimization in Autonomic Computing and Resource Management

Iqura Khan, Alpana Meena, Prashant Richhariya, and Bhupesh Kumar Dewangan

1 Introduction

Before discussing optimization, first, it is essential to understand *what is the meaning of optimization?*

In simple words, optimization is about making things in their best state or selecting inputs that produce the best outputs.

This can mean a variety of things like:

- allocation of available resources
- producing the best characteristics of a design
- selecting control variables for desirable system behavior

Optimization generally involves the word to maximum or minimum [1]. Optimization is additionally useful when constraints or limits are applied to the involved resources or restricting the boundaries for desirable solutions.

To search the answer for the optimization problems, a special kind of program is used which is known as optimization algorithm that will be discussed later in the chapter.

Optimization is applicable to a variety of situations. For example:

- To minimize or decrease shipment time, an optimal location of the warehouse is selected for potential customers.
- Designing an optimal cost bridge so that it can carry maximum load.

I. Khan · A. Meena · P. Richhariya
RGPV, Bhopal, India

B. K. Dewangan (✉)
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_9

- Regulating insulin secretion from artificial pancreas to minimize or reduce the difference between real and desired blood insulin level or sugar level.
- Designing the wing of an airplane to minimize weight while maintaining strength.
- Choosing the right stock's set to maximize the return-based investment in predicted performance.
- Regulating the temperature of the chemical process to maximize or increase the purity of the production.

From the abovementioned examples it can be noted that optimization is a strong tool that can be applied in many applications and mentioned are the few fields that adopt optimization techniques to improvise the solution's quality.

Summary:

- Optimization makes better results by helping input selections.
- An optimization algorithm is required in most optimization problems.
- Optimization is applicable to several disciplines.

Now, it is important to discuss *what resource management is in cloud computing?*

A Cloud user requests resource from the cloud provider for running some kind of application that is allocated by the provider.

- CSP's objective is to maximize resource utilization.
- Users will consume the resources to increase their overall performance and minimum cost.

Two stakeholders have contradicting requirements, and resource management is how these two objectives are met as shown in Fig. 1.

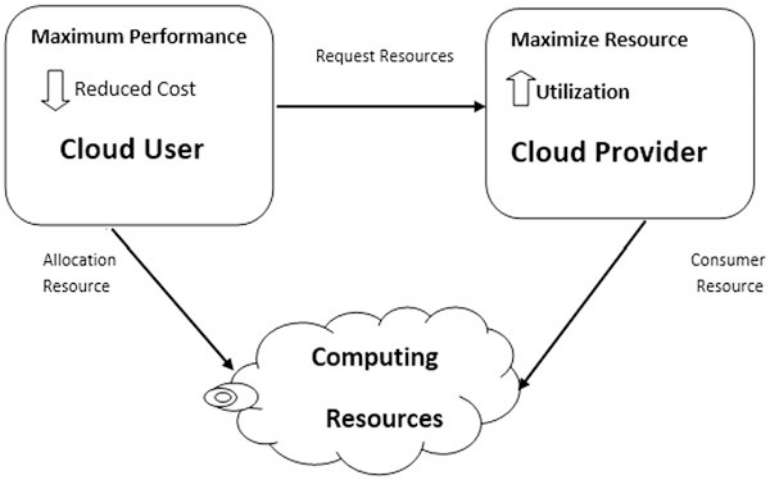

Fig. 1 Resource usage in cloud

A resource is an item that is used by consumers in terms of cloud computing [2–6]; it is the hardware of a machine such as CPU, memory, processors, I/O, etc.

- Resource Management is the utilization of resources to the best of its ability.
- The process of resource allocation for the consumer toward meeting the performance objectives.

In one of the studies, it was concluded that the server resources of the data centers are 20% utilized, and the rest 80% symbolizes the idle servers, and the total power consumption of those idle servers is 60% [2]. Also, resource management is essential for the efficient use of hardware [3, 4].

This can be understood with the help of an example; suppose a person plans for a party and requires help from his friends. So, he needs to:

- Make a list of his requirements.
- For smooth functioning or better results, he must assign or schedule the tasks for each one of them.

Not only resource management is important, but autonomic resource management is the demand of this era. When the resource demand of the user is dynamic or uncertain [5], then it becomes important that resource management self-configuration must be introduced to handle dynamic requirements [6].

1.1 What Is the Need for Optimization in Resource Management?

As continued from the abovementioned example, while making a schedule initially, it is figured out who can do what kind of work and what kind of dependencies the work item has on each other. So, the tasks were assigned accordingly. But later it was realized that few people are overloaded and few are underloaded.

The resource optimization process is bringing optimization in resource assignment to the activities to meet the performance objectives.

Optimization of resources is the major challenge in resource management. Also, efficient resource utilization is the major purpose of cloud providers. Misuse of resources would result in high energy utilization and cost because of underloaded hosts [7].

Resource management makes the cloud cost-efficient, whereas optimization of the resource means minimizing the cost and time or making the system overall efficient. The efficiency of a system can be increased by optimizing certain parameters [8].

So, to achieve efficient resources management optimization is required. Now, the question arises *how to do optimization?*

To generalize, optimization can be achieved by following four simple steps as shown in Fig. 2:

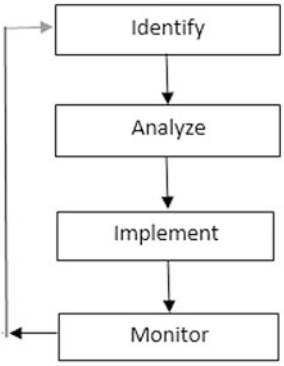

Fig. 2 General optimization steps

Step 1: Identity:

Choose the problem statement that needs to be optimized and then the purpose and goals of it are defined.

Step 2: Analyze:

An optimization algorithm is chosen, and then it is analyzed whether it is meeting the desired goals or not.

Step 3: Implement:

If step 2 is satisfied then the chosen algorithm is implemented.

Step 4: Monitor:

In this step, evaluation is done after implementing algorithms to obtain the desired results.

If the desired results are obtained, then the process is stopped here; otherwise, the same process is followed again and again to obtain the desired results.

The rest of the chapter will consist of the following sections:

- Sect. 2 provides a quick overview of the literature survey.
- Sect. 3 describes different optimization algorithms.
- Sect. 4 presents the future scope followed by the conclusion.

2 Related Work

Several researchers are interested in heuristic algorithms and autonomic resource management. Few of the research works are presented here.

Load balancing is one of the major challenges and area of concern in cloud computing environments; [9] it is the way of assignment and reassignment of the loads among free or available resources to maximize the resultant throughput while minimizing the overall cost and response time. It also includes performance improvement and resource utilization as well as energy-saving techniques [10].

For autonomic cloud resource management, the system must be self-configuring, self-healing, and self-optimizing. For this, the authors [11] considered two key

parameters to reduce the cost, one is energy consumption and the other is minimizing the cost.

The authors present a detailed description of resource management, its taxonomy, and its challenges. Also, the author describes various performance parameters and maps the goals that must be paid attention while designing a new RM technique.

Resource management is one of the important aspects of a cloud computing environment. The authors [12] give a detailed classification of cloud resource management based on its cost, profit, nature-inspired algorithms, and many other parameters. Then, they mentioned in brief about the issues and challenges in resource management. Also, the authors [8] discussed resource optimization systems based on time and implementation cost to increase the effectiveness of the system.

An author [13] proposed a tri-objective resource optimization algorithm that uses cost function for optimization and compared ACO, GA, BFO, PSO, and proposed TROA and concluded that TRAO is best in terms of performance compared to others. Another author [14] uses ant colony optimization for live virtual machine migration approach with seven modules and concluded that by using ant colony optimization, the energy consumption by live virtual machines reduces. The author prepared a survey paper on PSO algorithm in cloud computing. They described the mathematical part of the algorithm and discussed PSO-based scheduling for load balancing [15]. The author [16] described different optimization algorithms such as ACO, optimized ACO, etc. The author described task scheduling optimization based on heuristic search [17]. Sood [18] proposed hybrid algorithm for task scheduling, namely, HFGSA, and compared the outcomes with ACO, GSA, and FA. The proposed work results in improved response time, processing time, and make span.

3 Optimization Algorithms

The objective of optimization could be:

- to minimize or reduce the cost and
- to maximize efficiency.

An optimization algorithm is a practice that is executed repeatedly by correlating different solutions until an efficient or optimum or a satisfying solution is achieved.

3.1 Types of Optimization Algorithms

Based on the type of problem, optimization algorithms are categorized into several types. Choosing the right optimization algorithm is necessarily important for searching correct solutions for a problem. There are a variety of optimization algorithms, including gradient-based algorithms, derivative-free algorithms, and

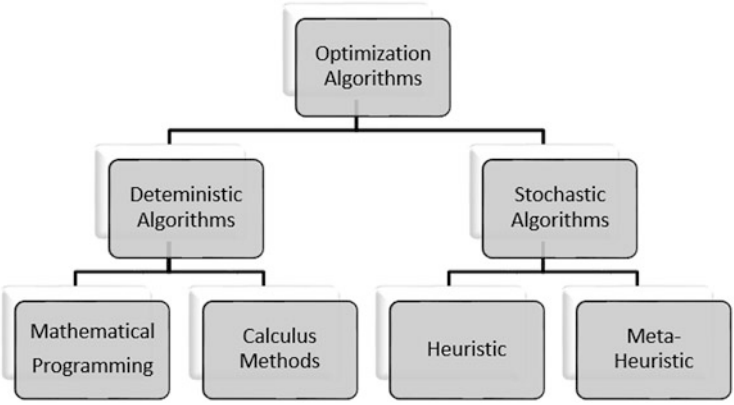

Fig. 3 Classification of optimization algorithms

meta=heuristics. In this chapter, optimization algorithms are classified into two categories based on their results. They are deterministic algorithms and heuristic algorithms [19] presented in Fig. 3.

1. *Deterministic algorithms*: These are the algorithms that produce the same output for specific inputs. The behavior or the output of the algorithm is determined or can be predicted.

   For example: Sorting, where the output can be predicted easily and for the same inputs the result of any sorting algorithm will be the same.

   Several algorithms can be categorized under it as shown in the Fig. 3. The following are the two major categories of deterministic algorithms.

   (a) Mathematical programming: Mathematical programming is the numerical method of optimization, including linear programming, integer programming, and so on.
   (b) Calculus methods: The calculus methods of optimization are the basic form of nonlinear programming.

2. *Stochastic algorithms*: These algorithms are random in behavior as their name suggests, and the following are two broad categories of it:

   (a) Heuristic algorithms: These generally work on trial-and-error methods [14], and their rules are based on past experience that is beneficial to make decisions.

   For example: Genetic algorithm, simulated annealing algorithms, etc.

   (b) Meta-heuristic algorithms: These algorithms are one step above compared to heuristics algorithms. Modern meta-heuristic algorithms are usually nature-inspired, and they are appropriate for global optimization [20].

For example: Ant Colony Optimization, Particle Swarm Optimization Algorithm, etc.

3.2 Why to Study Meta-heuristic Algorithms?

Resource allocation, resource scheduling, workload balancing, etc., all come under resource management of cloud computing. Resource allocation and its scheduling for high-performance computing system like an autonomic cloud computing environment is well known as NP-complete and specifically, parallel resource scheduling problems are NP-hard [21, 22]. Proven scientific research show that meta-heuristic algorithms give high-quality solutions for NP-hard optimization problems [23, 24].

Hence, it can be concluded that for best possible resource management, meta-heuristics algorithms are best fitted.

Heuristics produce acceptable solutions to complicated problems in practical time, but there is no guarantee that the produced solutions are best, and they can be nearly optimal [20].

Meta-heuristic optimization algorithms help solve real-world problems due to its

1. clarity and simple implementation
2. no obligation of slope information
3. local optima is avoided
4. may be used in several ranges of problems with many disciplines.

Different methods of the search process for optimal solutions make meta-heuristic algorithms unique.

It was earlier discussed what are meta-heuristic algorithms, and now, before discussing them, it is important to know about its types.

3.3 Types of Meta-Heuristic Algorithms

Hundreds of meta-heuristic algorithms exist, and they are classified into several categories based on certain parameters. In Fig. 4, meta-heuristic algorithms are broadly classified into two categories. These are nature-inspired algorithms and non-nature-inspired algorithms.

Also, meta-heuristic algorithms are classified based on its neighbor structure into multi-neighbor structured and single neighbor structured. So, the algorithms are classified as population-based algorithms and single-point algorithms. Figure 5 presents a chart for the same.

Other bases for classification of meta-heuristic algorithms are

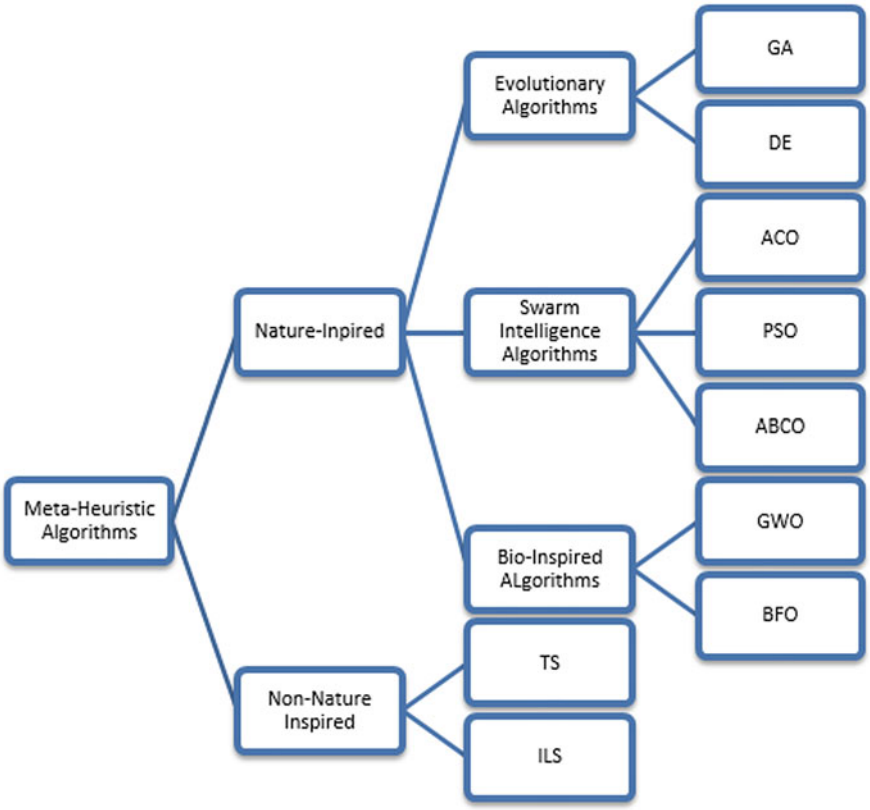

Fig. 4 Classification of meta-heuristic algorithms

- memory-based or memory-less
- whether they are hybrid or not
- whether they are iterative or greedy
- Metaphor-based or nonmetaphor-based.

So, the conclusion is that there are several meta-heuristic algorithms, and they are classified into different categories. But few algorithms are most popular among all as they give optimal results. In this chapter, the most popular meta-heuristic algorithms are discussed one by one.

1. *Genetic algorithm (GA)*: This algorithm was given by John Holland [25] with the concept that how genes are transferred from parents to their next generation and from the group of individuals, the best will survive according to the survival of the fittest. It is used to find optimal solution for NP-hard problems such as Travelling Salesman Problem. This algorithm comes under multi-neighbor structured or multipoint searching which is known as population. In this approach, the

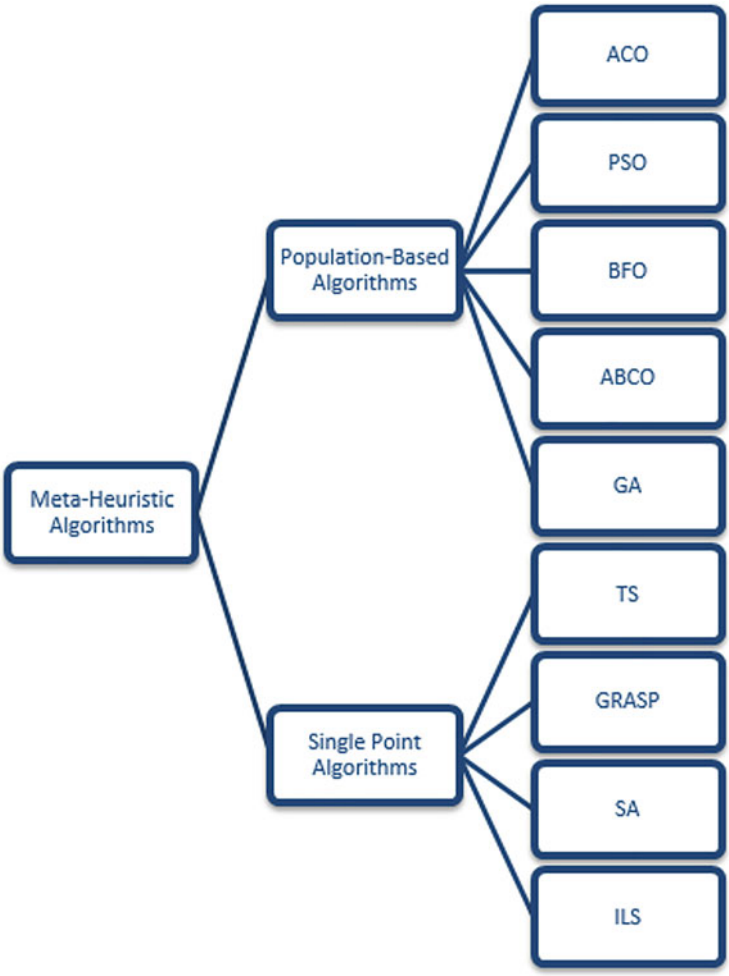

Fig. 5 Classification of meta-heuristic algorithms

solutions are represented as chromosomes that improve with each stage. Survival of the fittest is calculated using the fitness function. The general procedure of GA is mentioned below along with Fig. 6:

Procedure of GA

1. Initialization: Create an initial population which is diverse in nature consisting of chromosomes (existing solutions).
2. Fitness: By applying fitness function, the fitness value for each and every chromosome is calculated.
3. Selection: Select the chromosome based on its fitness value.

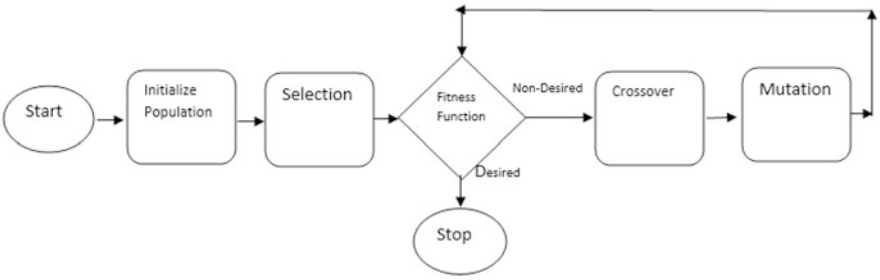

Fig. 6 Genetic algorithm

4. Crossover: Now crossover is performed on the chromosome's pair secured from step 3.
5. Mutation: Along with crossover, mutation is performed on the pair of chromosomes.
6. Fitness: Now, the fitness value of these recently generated chromosomes called offspring is again calculated.
7. Repeat steps 3–7 until desired results are obtained.
8. When highest fitness value is obtained end the procedure, and optimal solution is hence obtained.

2. *Differential evaluation (DE)*: It is classified among of the strong evolutionary algorithms that is efficient in global optimization. It again follows the same concept of candidate evolution. It works with three parameters that are population size NP, scaling factor, and crossover rate [26]. It focuses on evolution of population by performing mutation and crossover over the population. Then, the selection is done based on certain criteria. This process repeats until certain criteria are met. Moreover, it works like GA the main difference is created by its fitness function and the methods of mutation and crossover. So, they have differences in their performances. GA performed better for initial generations, while DE refined solutions to achieve impressive fitness values [27].
3. *Ant colony optimization (ACO) algorithm*: It is among the nature-inspired algorithm that is based on the concept that ants find the path for their food. When an ant finds the path of the food, then it releases a hormone called pheromones for others to find the optimal route. This is one of the most widely used algorithms in the field of cloud computing, where the ants are assumed as users or agents, the food is assumed as resources, and the nodes are assumed as virtual machines. The virtual machines are traversed by the agents in search of resources; likewise, the nodes are traversed by the ants in search of food. The general procedure of ACO is described below with the help of Fig. 7.

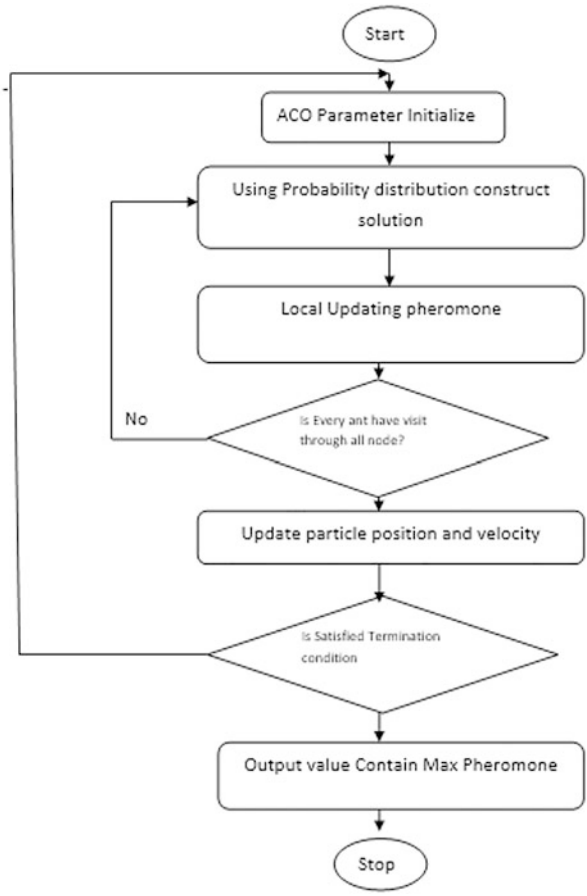

Fig. 7 Ant colony optimization algorithm

Procedure ACO

1. Initialization:

   (i) For each path between tasks and resources, initialize the pheromone value.
   (ii) Initialize optimal solution equals null.
   (iii) On random resources, place m ants.

2. Solution for each one ant:

Repeat for each one ant until each ant constructs its solution

(i) For the first task, put the first resource in the list.
(ii) For rest of the tasks.

Choose the adjoining resource rj by following rule of transition.
else 0.

3. Fitness: For each ant fitness value is computed.
4. Replacement: If the calculated fitness value in step 3 is better than optimal solution, then replace the older optimal solution.
5. Updating the Pheromone:

   (i) For each edge update local pheromone.
   (ii) Update global pheromone.

6. Empty the record of all ants.
7. Repeat the steps number 2 to step number 6 until optimum solution is achieved.

4. *Particle swarm optimization (PSO) algorithm*: This algorithm is also iterative in nature like GA and ACO to reach the optimal solution. In PSO, particles are the candidate solutions. Optimization involves the movement, which is guided by mathematical formulae of these particles around a search space. If better search spaces are found, then the better positions are updated to other particles as well; then the swarm is supposed to converge toward the optimal spaces achieving self-organization. General procedure of PSO is also shared below:

   Procedure of PSO

   1. Initialize: For each particle, the position vector's variable and velocity vector's variable are initialized.
   2. Convert: The constant position vector is converted to the discrete vector.
   3. Calculate fitness: Using fitness function, fitness value of each one particle is calculated.
   4. Calculating the best position: Each particle is allocated to its best or optimum position value.
   5. Replace: The current position of the particle is replaced if the calculated value in step 4 is improved than the previous.
   6. Update: Update each one particle's position.
   7. Repeat step 2 to step 6 until desired condition is achieved.

5. *Artificial bee colony optimization (ABCO) algorithm*: This algorithm is influenced by the lifestyle of honey bees, where three types of bees are involved which are employee bees, onlookers, and scouts. The onlookers are bees that make the decision for the selection of a food source. The bees that visit by itself to the source are the employed bees, and the bees that carry out unplanned search are the scouts. In the ABCO algorithm, the colony is segregated into two parts, the first is employed bees and the second is onlookers. There is a unique employee bee for each and every food source. The employee bees and sources of food are equal in number near the hive. The employee bee whose source of food is drained by the onlookers and employed bees will now become a scout [28].

   Procedure ABCO:

   1. Initialization.
   2. Iteration.

(a) Locate the employee bee on the sources of food in memory
(b) Locate the onlooker bee on the sources of food in memory
(c) Instruct the scouts to go to the search area for exploring new sources of food.

3. Repeat the above steps until requirements are met.

To summarize the ABCO algorithm involves three steps for each cycle:

(i) Sending the employee bees onto the sources of food and then calculating the amount of their nectar
(ii) Then the onlooker bees choose the sources of food after passing the information of the employee bees and about the amount of food present in the nectar
(iii) Locating the scout bees and then directing them to best possible sources of food.

6. *Grey wolf optimization (GWO) algorithm*: This algorithm was proposed by Seyedali in 2014 inspired by the lifestyle of wolves as they have great capabilities to catch their prey. According to the internal management hierarchy, the wolves are divided into four groups. These are alpha, beta, delta, and omega. The best individuals are termed as alpha, second-best as beta, third as delta, and the remaining as omega. The hunting is guided by the best wolves for the best searching spaces. The omega individuals will reallocate themselves based on the location of alpha, beta, and delta [29].

   Here, the hunting process is the optimization, the candidates are referred to as wolves, and their prey is the resource.
7. *Bacterial foraging optimization (BFO) algorithm*: This is included in global optimization algorithm which is based on the movement of bacteria *Escherichia coli*. The objective of the algorithm is that the way the animal finds its food is similar to the power consumption per time (P/T). This algorithm follows the procedure such as reproduction, elimination, chemotaxis, and dispersal.

   1. Chemotaxis: This relies upon the movement of the bacteria via flagella by immersing and breaking down. Its motion toward the food can be pulled or declined, and it attracts different bacteria.
   2. Reproduction: Now, the fitness value of the bacteria is evaluated, and the healthier ones will reproduce, and later after evaluation dispersal is done.

Table 1 presents a brief difference between nature-inspired algorithms based on certain features.

4 Conclusion

From this chapter, it can be concluded that meta-heuristic algorithms are best fitted for resource optimization, and they produce near-optimum solutions for NP-hard

Table 1 Comparison of GA, ACO, BFO, and PSO

Features	GA	ACO	PSO	ABCO	GWO	BFO
Category	Evolutionary algorithm	Swarm intelligence	Swarm intelligence	Swarm intelligence	Bio-inspired	Bio-inspired
Inspired by	The class of inheritance	The colony of ant to find the food [30, 31]	The behavior of birds or fish	The lifestyle of honey bees	The hunting mechanism of wolves	The motion of the bacterium as heuristics
Initialization	Existing population	"Ants" represents the state of the problem	"Particles" represents the state of the problem	"The three types of bees" represent the state of the problem	"The four types of wolves" represent the state of the problem	"Bacteria" represents the state of the problem
Basic concept	Natural laws of evolution and Darwin's theory of evolution	Self-organized [32]	Self-organized or decentralized system	Self-organized and centralized	Self-organized and centralized	Self-organized
Technique	AI technique where search heuristics are applied	Probabilistic technique	Computational iterative method	Computational iterative method	Probabilistic technique	Chemotaxis, reproduction, elimination, and dispersal
Rationale	Survival of the fittest	Pheromone from ants	Particle moves in the search space with varying speed	Division of labor from bees	Hunting mechanism from wolves so that prey can be catched guaranteed.	Power consumption per time (P/T)
Method	Fittest chromosomes among all are selected to produce offspring	Pheromone released by the ants to trace the path.	Particle is moved in the search space and measured by certain parameters	Division of labor to find best nectar	Movement is search space to find and catch the prey	Hyper-heuristics method

and NP-complete problems. Different types of meta-heuristic algorithms exist, and most of them produce optimum results under valid conditions. Heuristics is the name of trial and error, applying common sense, and many more. Several heuristics techniques are usually independent of the given optimization problems and share universal aspects. As discussed in the paper, different types of meta-heuristic algorithms can be applied to different varieties of problems, depending upon the nature of the problem and the desired outcome. Recently, only hybrid, combination of many, and meta-heuristics have been widely used. The benefit of combining distinct algorithms is widely recognized. A proficient combination of meta-heuristic algorithm with its components to other meta-heuristics algorithms are applicable to research techniques or mathematical programming models, artificial intelligence, or constraint programming or to complete algorithms like branch and bound. This may lead to obtaining better solutions for optimization problems. This field is called hybrid meta-heuristics which is beyond the range of classic meta-heuristics.

References

1. Törn, A., & Žilinskas, A. (1989). *Global optimization* (Vol. 350). Berlin: Springer-Verlag.
2. Lee, L.-T., et al. (2013). A dynamic resource management with energy saving mechanism for supporting cloud computing. *International Journal of Grid and Distributed Computing, 6*(1), 67–76.
3. Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *The Journal of Network and Systems Management, 23*(3), 567–619.
4. Gulati A, Shanmuganathan G, Holler A, Irfan A. (2011). Cloud scale resource management: Challenges and techniques. In Proceedings of 3rd USENIX workshop on hot topics in cloud computing (HotCloud 2011).
5. Mustafa, S., Nazir, B., Hayat, A., Khan, A. U. R., & Madani, S. (2015). Resource management in cloud computing: Taxonomy, prospects and challenges. *Computers & Electrical Engineering*. https://doi.org/10.1016/j.compeleceng.2015.07.021
6. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer, 36*(1), 41–50. Los Alamitos, CA (USA): IEEE Computer Society Press.
7. Nzanywayingoma, F., & Yang, Y. Efficient resource management techniques in cloud computing environment: A review and discussion. *International Journal of Computers and Applications, 41*(2), 1–18. https://doi.org/10.1080/1206212X.2017.1416558.
8. Jadeja, Y., & Modi, K. (2012). Cloud computing - Concepts, architecture and challenges. International conference on computing, electronics and electrical technologies [ICCEET].
9. Singh, P., Baaga, P., & Gupta, S. (2016). Assorted load-balancing algorithms in cloud computing: A survey. *International Journal of Computer Applications, 143*(7), 34–40.
10. Agarwal, A., Venkatadri, M., & Pasricha, A. (2018). Autonomic cloud resource management. 2018 fifth international conference on parallel, distributed and grid computing (PDGC), Solan Himachal Pradesh, India (pp. 138–143). https://doi.org/10.1109/PDGC.2018.8745977.
11. Mustafa, S., Nazir, B., Hayat, A., Khan, A. U. R., & Madani, S. A. (2015). Resource management in cloud computing: Taxonomy, prospects, and challenges. *Computers and Electrical Engineering, 47*, 186–203. https://doi.org/10.1016/j.compeleceng.2015.07.021.
12. Yamini, R., & Germanus Alex, M. (2017). Comparison of resource optimization algorithms in cloud computing. *International Journal of Pure and Applied Mathematics, 116*, 847–854.

13. Sutar, S., Mali, P., & More, A. (2020). Resource utilization enhancemnet through live virtual machine migration in cloud using ant colony optimization algorithm. *International Journal of Speech Technology, 23*. https://doi.org/10.1007/s10772-020-09682-2
14. Masdari, M., et al. (2017). A survey of PSO-based scheduling algorithms in cloud computing. *Journal of Network and Systems Management, 25*(1), 122–158.
15. Rana, M. S., Ks, S. K., & Jaisankar, N. (2013). Comparison of probabilistic optimization algorithms for resource scheduling in cloud computing environment. *International Journal of Engineering and Technology, 5*(2), 1419–1427.
16. Guo, L., et al. (2012). Task scheduling optimization in cloud computing based on heuristic algorithm. *Journal of Networks, 7*(3), 547.
17. Sood, K., Jain, A., & Verma, A (2017). A hybrid task scheduling approach using firefly algorithm and gravitational search algorithm. 2017 international conference on energy, communication, data analytics and soft computing (ICECDS). IEEE.
18. Özkaraca, O. (2018). A review on usage of optimization methods in geothermal power generation. *Mugla Journal of Science and Technology*. 130–136. https://doi.org/10.22531/muglajsci.437340
19. Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. London: Luniver Press.
20. Ullman, J. D. (1975). NP-complete scheduling problems. *Journal of Computer and System Sciences, 10*(3), 384–393.
21. Hall, N. G., & Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research, 44*(3), 510–525.
22. Blum, C., & Roli, A. (2008). Hybrid metaheuristics: An introduction. In C. Blum, M. Aguilera, A. Roli, & M. Sampels (Eds.), *Hybrid metaheuristics – An emerging approach to optimization. Studies in computational intelligence* (Vol. 114, pp. 1–30). Berlin: Springer-Verlag.
23. Hristov, A., Nikolova, I., Zapryanov, G., Kimovski, D., & Vesna Kumbaroska, V. (2016). Resource management optimization in multi-processor platforms. In *Proceedings of the third international workshop on sustainable ultrascale computing system (NESUS 2016) Sofia, Bulgaria* (pp. 23–29). Marid: University Carlos III de Marid. Computer Architecture, Communications and Systems Group (ARCOS).
24. Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
25. Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009, April). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Transactions on Evolutionary Computation, 13*(2), 398–417. https://doi.org/10.1109/TEVC.2008.927706.
26. Dewangan, M. B. K., & Shende, M. P. (2012). Survey on user behavior trust evaluation in cloud computing. *International Journal of Science, Engineering and Technology Research, 1*(5), 113.
27. dos Santos Amorim, E. P., Xavier, C. R., Campos, R. S., & dos Santos, R. W. (2012). Comparison between genetic algorithms and differential evolution for solving the history matching problem. In B. Murgante et al. (Eds.), *Computational science and its applications – ICCSA 2012. ICCSA 2012. Lecture notes in computer science* (Vol. 7333). Berlin: Springer. https://doi.org/10.1007/978-3-642-31125-3_48.
28. Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*(3), 459–471.
29. Wang, J., & Li, S. (2019). An improved Grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific Reports, 9*, 7181. https://doi.org/10.1038/s41598-019-43546-3.
30. Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Energy-aware autonomic resource scheduling framework for cloud. *International Journal of Mathematical, Engineering and Management Sciences, 4*(1), 41–55. https://doi.org/10.33889/IJMEMS.2019.4.1-004.

31. Faruqui, N., Yousuf, M. A., Chakraborty, P., & Hossain, M. S. (2020). Innovative automation algorithm in micro-multinational data-entry industry. In Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325, pp. 680–692). LNICST. Springer. https://doi.org/10.1007/978-3-030-52856-0_54
32. Venkatadri, M., Agarwal, A., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science, 152*, 204–211. https://doi.org/10.1016/j.procs.2019.05.044.

A Proposed Framework for Autonomic Resource Management in Cloud Computing Environment

Monika Mangla, Sanjivani Deokar, Rakhi Akhare, and Mehdi Gheisari

1 Introduction

It is well agreed that the human body is accustomed or prone to some kind of adverse situations like accidents. In such situations, it is noticed that the human body immediately reacts to overcome that particular situation. For instance, if we are caught in a fire, the inherent physical autonomous mechanism activates to handle such situations [1]. In such situations, the highly sophisticated autonomic nervous system in the human body also activates and accordingly adjusts and controls organs and functions. It can be easily understood by this example that if the water level in the human body goes down, some signal is sent to the throat that prompts the person to drink water. This whole process takes place in an autonomic manner under the guidance of the human brain owing to the inherent competence of our nerve system. Hence, a decision is taken at the top level, that is, the brain, and the same is executed at the local level (other body parts). This segregation and further delegation of activities exempt the human brain to perform intelligent tasks. The same kind of responsibility distribution and segregation is the working principle of autonomic computing [2].

From the above discussion, it is evident that monitoring, decision-making, and communication are the most important aspects of autonomic computing. This mechanism allows to manage complex systems through their components and thus empowers the central system by liberating it of routine tasks. As a result, the

M. Mangla (✉) · S. Deokar · R. Akhare
CSED, Lokmanya Tilak College of Engineering, Navi Mumbai, India
e-mail: manglamona@gmail.com

M. Gheisari
Islamic Azad University, Iran

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_10

system achieves self-healing, self-optimization, and self-correction referred to as Autonomic Computing (AC) [1].

As the complexity of the systems is increasing with the advancement in technology, it is opening avenues for AC [3]. Moreover, the advancement in Internet technologies further intricacies the process of obtaining a reliable and secure solution [4]. Also, technology has evolved so rapidly that researchers are focused on designing solutions based on biological systems to handle heterogeneity, complexity, and uncertainty. Resultantly, the current technological revolutionized systems need to incorporate various attributes such as availability, survivability, and maintainability to harness its full competence. This challenge is addressed by AC as discussed earlier as it calls for architectures that are self-optimized, self-maintained, and self-configured, and thus rarely need human administration and intervention. Understanding the demand, IBM has been working rigorously in the direction of AC [4] by building a system that is capable of managing itself by anticipating the workload and scheduling the resources accordingly. This research in AC is primarily inspired and motivated by the autonomic nervous system in the human body as discussed initially. Here in this chapter, the authors aim to present the AC model in a cloud computing environment.

The chapter is organized into various sections as follows. Here, the concept of AC is briefly introduced in the first section. Cloud computing [5, 6] is briefly discussed in Sect. 2 to maintain the completeness of the chapter. The incorporation of autonomous computing is mentioned in Sect. 3. Section 4 discusses the related work. The generic architecture and proposed framework for autonomic computing are presented in Sects. 5 and 6 respectively. Finally, the chapter is concluded in Sect. 7.

2 Cloud Computing

Cloud computing primarily works on the principle of providing different computing services as per demand like storage, processing power over the internet on a pay-as-you-go basis. This allows the users to avail these services on rent from a cloud service provider instead of owning these. Hence, cloud computing provides an economical solution while avoiding the cost and complexity of maintaining infrastructure. Hence, the major characteristics of cloud computing are as follows:

Shared infrastructures: Shared resources maximize the utilization of available resources by sharing processing capabilities, storage capacity, and other physical services among users across the Internet.

Heterogeneous network access: Application can be accessed from heterogeneous network access nodes.

Dynamic provisioning: It permits to acquisition of services based on demand to maximize resource utilization. Dynamic provisioning is allowed as a result of competent software that enlarges and shrinks in response to the demand. Thus,

cloud computing allows sharing of resources/services by charging for the actual usage, thus providing a cost-effective solution.

The above characteristics enable cloud computing to harness some quite exciting achievements over comparative approaches. Among various benefits, the most promising features are cost-effectiveness in addition to features like scalability, reliability, and low maintenance [4]. Despite all these supporting features, cloud computing also bears some challenges just like a coin has two sides. However, the challenges faced in cloud computing [7] can be addressed through careful and efficient planning. Among these steps toward handling associated challenges, the most promising step is the introduction of a fog layer among end devices and cloud layers.

Fog computing can be considered as a decentralized infrastructure where computation and storage take place somewhere between the data source and the cloud [8] known as fog nodes. This intermediate storage and processing in fog computing provide various advantages over cloud computing like reduced data transmission, reduced latency, and minimal network congestion. These advantages are obtained owing to bringing the storage and processing closer to its source. Resultantly, fog computing enhances the performance of cloud computing [9]. The general architecture of cloud computing is shown in Fig. 1 [10].

The advantages of fog computing in comparison to cloud computing open various avenues for its implementation in the real world with full strength [11]. As a result, fog computing has attracted researchers and academicians since its inception. However, the real-world implementations necessitate the model to be robust and thus always available. For the same, it needs to incorporate autonomy in the model to achieve a robust system. For the same, it necessitates building a self-healing and

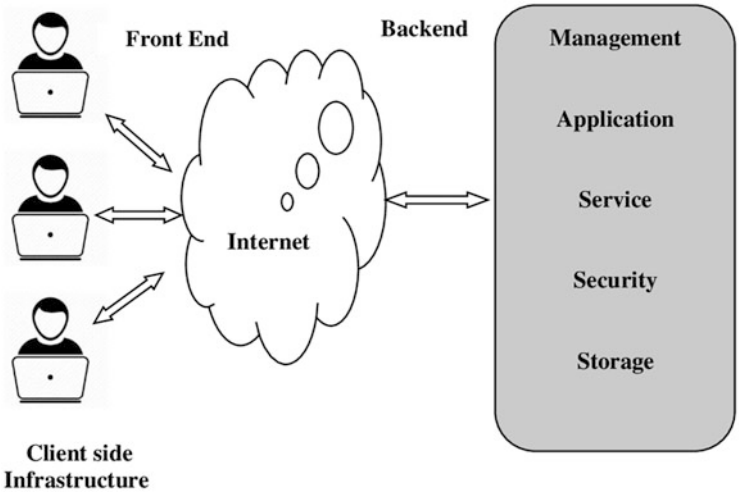

Fig. 1 Abstract model of cloud computing

self-diagnosis system that can sustain even during a crash. Hence, it is of paramount importance to design a self-diagnosis and self-healing infrastructure by inculcating autonomic computing in the cloud [12].

The requirement for inculcating autonomic computing can be understood by the fact that the excessive usage of resources may sometimes lead to performance degradation of the system [13]. If these preliminary indicators of system failure are not handled properly in time, it may lead to a system crash and the system may be completely unavailable for a few days. It leads to a drastic downfall in the system performance and hence early detection and healing are highly recommended. Previously, this maintenance was handled manually but cloud computing enables the handling of maintenance issues in an autonomic and centralized manner. The inculcation of autonomic computing in cloud computing is generally referred to as autonomous cloud computing [14].

3 Autonomic Computing in Cloud Computing

Basically, as per IBM, autonomic computing works over the concern of industry regarding its complexity [15]. Hence, it presented the AC as a tool that handles this issue as it contains multiple elements that handle detailed information about components, configuration, and optimization of the system through adaptive algorithms. Additionally, an autonomic computing system should also be competent in recovery from sudden attacks and failures. The basic functions in autonomic computing are illustrated in Fig. 2.

From Fig. 2, it is evident that general characteristics of autonomic computing can be considered as the ability to self-heal, self-optimize, self-configure, and self-

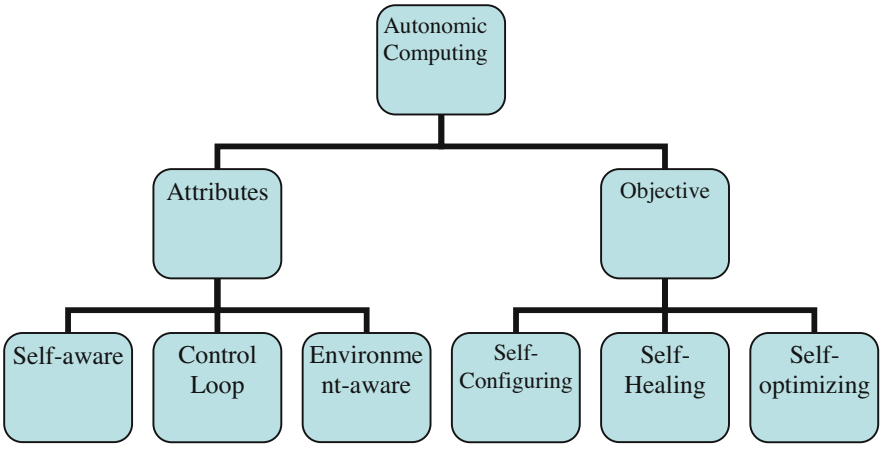

Fig. 2 Attributes of autonomic computing

protect. A system that can perform all these functions on its own without any human intervention is referred to as autonomic computing. AC system employs a high level of artificial intelligence and hence AC system functions autonomically based on the inputs it receives from the environment. The various functions incorporated in autonomic computing are as follows:

Self-Configuration: An autonomic system should be competent in reconfiguring itself to adjust according to unpredictable conditions to satisfy all goals and objectives.

Self-healing: The AC system must be competent in detecting, diagnosing, and recovering from unexpected problems to minimize service disruption. This property to recover from failures is referred to as self-healing. It may include identification of alternative resources in case of resource crash or upgradation of software, and so on.

Self-protection: To achieve self-protection, an AC system must be able to anticipate, identify, and protect itself from various threats to maintain availability and integrity.

Self-optimization: An AC system should be able to monitor methods and opportunities to enhance its operation with reference to numerous conflicting criteria.

To achieve these objectives, the AC system must be able to learn, organize, and regulate on its own. Additionally, it must also be able to acquire the knowledge and information from its environment to manage on its own. Hence, an AC system must be easily understandable without much description.

As discussed earlier, cloud computing basically consists of multiple data centers where each data center has tremendously huge storage and computational capacity. Although cloud computing gives an economic solution, it bears some challenges also. Some of these challenges are complexity and heterogeneity that necessitates integrating dynamic provisioning of resources and intelligent strategies for the autonomic environment [16]. This integration of autonomic computing in the cloud makes it a popular choice among clients from a business perspective as it provides the ability of self-monitoring, self-repairing, and self-optimizing that promises to enhance the overall performance of the system. Although autonomic computing can be incorporated in any environment it has proven to unleash unmatched performance enhancement in cloud environment owing to its dynamism, scalability, and complex behavior [17].

Through autonomic cloud computing, business organizations could save resources that were earlier deployed for optimization of security, scheduling, and maintenance of cloud systems as all these routines are automatically handled. Policies for autonomic systems are formed based on government guidelines and thereafter implemented to monitor the cloud infrastructure. Implementation of autonomic computing in cloud computing achieves several benefits such as:

Usage—Autonomic computing in cloud computing aims to achieve the optimal usage of all resources in the system.

Availability—In case of any failure, data remains available as it is backed up at other places as well. Additionally, it uses business service-level agreements (SLAs) for automated migration.

Cost—Autonomy in cloud computing accomplishes cost-effectiveness owing to automated movement of the workload from one cloud to another, if required.

Performance—Cloud autonomics leads to performance enhancement of the system as the workload is properly managed and distributed in the network.

Security—In accordance with the business policies, companies can opt for a change in network or endpoints of the system for security purposes.

From the above section, it becomes evident that cloud autonomics enhances the system performance multifold. Hence, cloud autonomics has become popular among researchers, academicians, and commercial organizations.

Autonomic cloud computing can also be referred to as a way that allows cloud infrastructures and platforms to take their own decisions to accomplish their tasks. Resource management is managed optimally through autonomic cloud computing. Moreover, in autonomic cloud computing, this resource management is managed without any human intervention [1]. Additionally, autonomic cloud computing also handles various performance metrics of the network in an optimal manner, which would otherwise need to be implemented using some realistic algorithms. Autonomic computing also incorporates algorithms for resource management and load prediction to manage the resources in the best possible manner. As a result of the efficiency of autonomic cloud computing, a significant rise has been experienced in the popularity and application of autonomic cloud computing.

The most significant factor that boosts the usage of autonomic cloud computing is reduced TCO (Total Cost of Ownership) and the maintenance cost. The requirement of manpower is also reduced by a huge margin. Another motive behind the popularity of autonomic cloud computing is its competence to provided improvised results, fault tolerance, and ease in handling. These objectives are achieved as a result of various evolutionary and genetic algorithms, artificial neural networks (ANNs), and combinatorial optimization heuristics. Implementation of these algorithms helps in reducing the operational, maintenance, and usage costs of clouds.

Considering the above features of autonomic cloud computing, it is evident that it maximizes system availability through server consolidation and cost minimization. Additionally, autonomic computing promises to simplify management. As a result, autonomic computing has observed its employment in various domains. Some of these are discussed in the subsequent subsection.

3.1 Applications of Autonomic Computing

Autonomic computing enables computer systems to perform basic maintenance and optimization on their own. This enhancement in automation relinquishes the

engineers and system administrators from routine work and allows them to focus on other activities [18].

Bootstrapping is an example of autonomic computing as it configures and initiates various processes during start-up. The process is also self-managed, which starts automatically whenever the computer starts. Thereafter, it proceeds through a self-diagnostic check that activates various hardware and software components accordingly.

Autonomic computing is also popular in Robotics and Artificial Intelligence (AI) as these require automation once the system starts. For instance, the AI system Amelia, developed by former IT specialist Chetan Dube, not only responds to verbal queries and answers questions; but it can also learn by listening to human operators particularly for questions that it cannot answer. Hence, as it learns and alters its programming, it is referred to as autonomic computing.

A Travel-Guide application also integrates autonomic computing to serve location-dependent service [19]. For the same, it tracks GPS and provides this input to an autonomic system that searches for available services in the vicinity. Social networking communication also generally has some autonomic properties like self-organization. These systems self-organize themselves to function with different configurations [20]. Few other applications of autonomic computing are monitoring power supply, automatic updating of software and drivers, and so on. Some real-time examples of autonomic computing are drone delivery service, medical diagnostic systems, cyberattack defense, and financial fraud prevention [21]. Apart from these applications, there are some broad categories of applications of autonomic computing as discussed below:

3.1.1 Autonomic Computing for Business Applications

Autonomic computing has observed wide applications in business applications [22]. Basically, the employment of AC in business empowers and embodies software with extra capabilities. The authors in [1] proposed an architecture that quickly realizes the failure and evolves autonomically. The various components of the framework are Knowledge Base, Autonomic Managers, Enterprise Service Bus (ESB), Policy Framework, and Language and Reusable Assets Repository for components (RAR) [23, 24].

3.1.2 Autonomic Computing for CRM

At the outset, autonomic computing has been significantly employed in CRM as well. The motive behind using autonomic computing in CRM is that traditional CRM starts sinking in response to the generation of huge data by producers. In such a scenario, the employment of AC achieves better manageability and quickly approaches the business goals. To employ AC in CRM, the framework has a self-optimizing generation manager that manages the load. Apart from the load manager,

the framework also has a pricing manager that optimizes pricing across various pricing policies. For the same, it may deploy various pricing schemes based on the season. Thereafter, it has a reminder manager that reminds about various activities through reminder [25].

3.1.3 Autonomic Computing for ERP

ERP systems handle tracking, planning, and controlling multiple resources in an enterprise. Here, the system gets so complex that each company employs a team to ensure proper functioning that incurs a huge cost. Moreover, the traditional system gets so complex that it becomes difficult to regulate communication among these modules. Hence, it necessitates an autonomic model of ERP to handle the functions of ERP in a simplified manner [26].

However, although autonomic cloud computing provides several benefits and has been employed in several domains, it also has some challenges as follows:

3.2 Challenges of Autonomic Computing

Apart from an abundance of advantages, autonomic cloud computing poses some challenges that are as follows:

- It requires strong Internet connectivity as it fails to perform optimally with a poor internet connection.
- Owing to the self-performing feature of autonomic computing, end users may even be unable to realize and detect errors.
- To handle cluster computing, software programming in autonomic cloud computing needs to be robust.
- For the full adaptation of autonomic computing, it requires incorporating some changes in the organization cloud system as it necessitates high technical standards.
- In autonomic cloud computing, there remains a geographical limitation as it remains unavailable in remote areas.
- Autonomic computing is not advisable for small-scale businesses as it costs more for the maintenance of cloud systems than other solutions. Moreover, autonomic computing contains numerous elements and hence necessitates proper coordination among them.
- It involves a huge cost to implement data security policies and their management.
- Owing to its dynamic nature, it involves different domains that lead to a huge trust issue among involved entities. Additionally, as completely automated means without any human interventions, it may also lead to inaccurate results sometimes.

In addition to the challenges discussed above, there are some additional issues as well [27]. Some of these issues are global system management, root cause analysis in case of some failure, testing/verification and standards, and so on. To address these challenges, autonomic computing, and cloud computing are used in integration. Some of such applications are discussed subsequently. Researchers have achieved some significant results related to autonomic computing. Some of the related studies have been presented in the following section.

4 Related Work

Authors in Kephart et al. [28] have implemented a high-level architectural blueprint that implements autonomic computing phase-wise. The proposed architecture claims that a control loop is implemented that monitors, analyzes, plans, and executes in response to the environment to achieve self-management. Such control loops can be incorporated in any run-time environments that need to achieve autonomy. The work is extended by Coutinho et al. [29] by proposing an extensible architecture of cloud computing based on autonomic computing concepts. In this, the authors designed two experiments that use micro-benchmarks on private and hybrid cloud environments. The results demonstrated that cloud computing and autonomic computing may be integrated providing elasticity. Furthermore, authors in Singh et al. [30] proposed an efficient Autonomous Agent-Based Load Balancing Algorithm (A2LB) that dynamically manages load in cloud environments. Authors in Singh et al. [31] extended the work by proposing a new Agent-based Automated Service Composition (A2SC) algorithm that processes requests. It also manages automated service composition phases but is not liable for managing comprehensive services.

Following the same line of research, Ghobaei et al. [32] presented a framework that controls Elasticity (ControCity) of resources using "buffer management" and "elasticity management". Additionally, authors in Nazir et al. [33] propose an autonomic computing security framework for Supervisory Control and Data Acquisition (SCADA) systems that handles cyber threats [34, 35]. This autonomic computing implements intelligent computing that triggers actions under given conditions. For the same, an integrated approach that combines cognitive approaches with discrete knowledge-based approaches is utilized to capture process- and systems-related threats.

Just like cloud computing, Etemadi et al. [36] present an efficient resource provisioning approach inspired by an AC model based on the Bayesian learning technique for fog computing. The proposed work is also validated in terms of effectiveness under three workload traces that illustrated that the proposed solution achieves a huge reduction in the total cost and delay violation. Moreover, it also achieves an increase in fog node utilization over comparative methods. Thereafter, authors in Kaur et al. [37] present the advantages of autonomic fog computing and claim efficient achievement of QoS parameters and thus advocates the strength

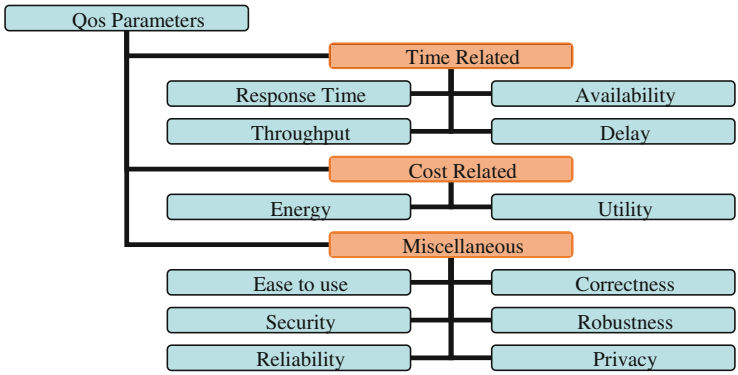

Fig. 3 Illustration of various QoS parameters

of autonomic computing in fog computing Kayal et al. [38]. The various QoS parameters are mentioned in Fig. 3.

Authors in Zhao et al. [39] addressed Autonomic Computing and Communications from the perspective of software-driven networks. Here, the authors discussed multiple perspectives of AC, namely, testing, integration, and deployment of network function. Lam et al. in [40] present an approach that makes it easy to develop interoperable IoT systems at a semantic level. Similarly, Khorsand et al. in [41] further propose a hybrid resource provisioning system that performs well for multitier applications. Moreover, the authors also validate the efficiency and effectiveness of the proposed approach under synthetic and real workloads. During this experimentation, it is established that the proposed solution performs better than existing approaches in terms of various QoS.

To self-heal from internal and external attacks, Singh et al. [42] proposed a technology called SHAPE. The proposed technique up to some extent addresses the issue related to increasing security demand in complex and heterogeneous networks. It also protects hardware, network, and application from failures and prevents the network against Denial of Service (DoS), User to Local attacks, and examining faults. Similarly, Gill et al. in [43] proposed a technique called BULLET for efficient resource management that used swarm optimization. According to BULLET, the resources can be properly assigned through optimization algorithms to handle highly scattered and heterogeneous resources. Through this approach, BULLET effectively decreases execution cost, time, and vitality utilization alongside different QoS parameters.

Singh et al. in [44] proposed an autonomic cloud framework named SOCCER (Self-Optimization of Cloud Computing Energy-efficient Resources) for effectively scheduling cloud assets. The Proposed SOCCER framework performs better in terms of the utilization of cloud assets. Thereafter, in 2019, Singh et al. proposed a technique called self-configuring and self-healing of cloud-based resources (RADAR) that manages resources in a cloud computing environment in an intelligent manner [45]. The activities handled in these related work are presented in a tabular form in Table 1.

Table 1 Various functions of autonomic computing by researchers

	Load balancing	Elasticity provisioning	Resource provisioning	Self-healing	Self-protection	Self-optimization
Kephart et al. [28]				✓		
Coutinho et al. [29]		✓				
Singh et al. [30]	✓					
Ghobaei et al. [32]		✓				
Etemadi et al. [36]	✓		✓			
Khorsand et al. [41]	✓		✓			
Kaur et al. [37]				✓		✓
Zhao et al. [39]				✓		✓
Gill et al. [43]						✓

5 Generic Architectures

To understand the existing architectures, it is imperative to comprehend the generic architecture for autonomic computing [46]. For autonomic computing, it requires QoS-based autonomic resource management that manages the resources in a natural manner in response to the demand in the environment. Just like a manual system, the autonomic framework adjusts itself to uncertain circumstances and adapts accordingly, inspired by the human nervous system for handling critical situations. For the same, there are various techniques in existence that satisfy the QoS parameters. Here, it is also worth noting that with the growth of the Internet and its widespread application, a quick response is always desired. For the same, some authors also suggest the incorporation of the fog layer as it caters to the requirement of reliability, security, resource utilization, and scalability without compromising response time [47].

The basic architecture for AC generally works in a heterogeneous environment. Here, this architecture primarily has two components, namely, Autonomic Manager (AM) and Autonomic Element (AE). Additionally, it has a loop that controls the flow of work among various AEs known as a control loop. The basic block structure of autonomic computing is illustrated in Fig. 4.

The prime role of AM is to implement a control loop that manages to preserve correct software architecture so that the four parts work efficiently in coordination and enhance autonomic loop functionality [48]. Furthermore, AE has additional objectives that need to be achieved. To meet these objectives, the components constituents an autonomic environment that consists of a management component (MC) within an autonomic element (AE). Finally, the control loop consists of sensors and effectors. Here, sensors sense the various parameters in the network and actuate accordingly [49]. Load balancing among nodes in cloud computing

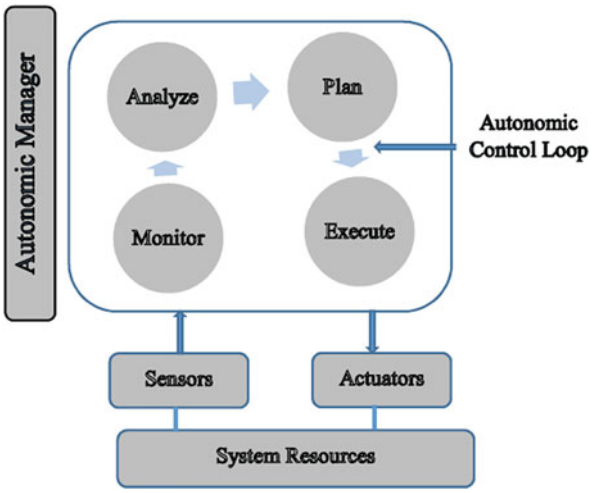

Fig. 4 Generic architecture of autonomic computing

environment is also a crucial parameter that can be efficiently handled by Autonomic Computing [50].

6 Proposed Architectures

In this chapter, the authors propose an architecture that goes beyond cloud computing and can also be implemented for fog computing. Fog computing is a mechanism that empowers the integration of cloud and IoT. As the cloud is unable to handle the massive volume of data generated by the IoT devices owing to the limited storage and computational complexity of IoT devices. Therefore, to overcome the issues and challenges of cloud computing, fog computing has been in existence for few years. It suggests introducing a computational layer between cloud and IoT, which act as an intermediary between them. More specifically, this computational layer is placed near the end devices so that latency time is optimized. Hence, the employment of fog layer originates a three-layered architecture and thus is in dire need of autonomic computing. Here, the fog layer consists of and handles the self-adaptive and autonomous components that achieves an efficient layered architecture from the perspective of processing and energy aspect.

As the fog layer consists of computationally efficient devices, the proposed framework proposes the incorporation of a control loop in the fog layer. For the same, it consists of two types of components, the fog master and the fog slaves as computational resources. Here, fog master is computationally more efficient in comparison to fog slave. The proposed work is illustrated in Fig. 5.

In the fog layer, the fog nodes perform the computations using computational resources like CPU and RAM. Furthermore, the results are stored using storage capabilities. As discussed earlier, the fog master is a powerful fog node with extended capability to perform IoT services and manage the fog slaves.

In the proposed framework, whenever a task is requested by IOT layer, it is forwarded to the fog master through fog slaves. Here, the fog master may receive multiple requests from the IoT layer. Upon receiving the request, the fog master may decide to perform the task itself or deploy it to fog slaves or even to cloud layers. This deployment is decided based on various network parameters like resource requirement of the task, current status of the nodes, and other miscellaneous information. In this approach, it is ensured that adequate resources (CPU time and memory) are available to accomplish the requested task in the stipulated time.

Here, the fog master performs the function of an autonomic manager and hence performs the following functions to achieve all required characteristics of autonomic computing:

- Here, the monitor routine handles the collection, aggregation, filtration, and reporting of the details obtained with the help of managed resources. Here, the useful information refers to topologies and metrics, and so on.

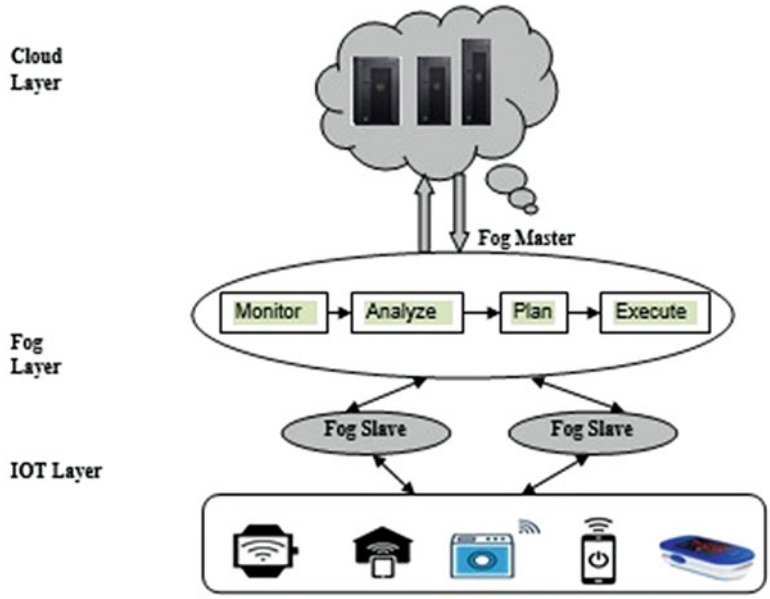

Fig. 5 Illustration of proposed framework

- Thereafter, analyze function handles correlating and modeling of the complex situations. Examples of a complex situation may be queuing models and time-series forecasting. Hence, it permits the autonomic manager to analyze the IT environment that aids to predict future situations.
- After monitoring and analyzing, the plan function is responsible to decide and frame the sequence of actions that are required to achieve set objectives. Hence, planning refers to various policy information to frame the sequence of actions.
- Finally, the execute function controls the execution of actions (set by plan phase) in consideration with dynamic behavior.

Hence, by performing all these functions in the fog layer, the proposed framework relinquishes the cloud layer from autonomic computing and hence performs better in comparison to traditional autonomic cloud computing. Moreover, as the fog layer is placed near the end devices, that is, near the site of data generation, it becomes easier to perform all functions related to autonomic computing in the fog layer.

7 Conclusion

The technological revolution in IoT leads to the generation of an abundance of data from the IoT networks. This data is generally uploaded to the clouds to elude

from maintaining a huge storage capacity. However, the storage of huge data may sometimes lead to performance compromise, which is not acceptable. This led to the emergence of autonomic computing that enables a network to adapt to requirements concerning huge storage and computational capacity. As a result, autonomic computing has attracted researchers during the past few years. In this chapter, the authors discussed the concept of autonomic computing with reference to cloud computing. The prime objectives of autonomic computing are self-healing, self-optimization, and self-configuring. The authors presented the generic framework of autonomic computing. Finally, the authors proposed a framework that suggests the implementation of autonomic computing in the fog layer. Implementation of autonomic computing in the fog layer enhances the performance by a huge margin.

References

1. Kurian, D., & Raj, P. (2013). Autonomic computing for business applications. *International Journal of Advanced Computer Science and Applications, 4*(8).
2. Alippi, C., Fantacci, R., Marabissi, D., & Roveri, M. (2016). A cloud to the ground: The new frontier of intelligent and autonomous networks of things. *IEEE Communications Magazine, 54*(12), 14–20.
3. Sterritt, R. (2005). Autonomic computing. *Innovations in Systems and Software Engineering, 1*(1), 79–88.
4. Dong, X., Hariri, S., Xue, L., Chen, H., Zhang, M., Pavuluri, S., & Rao, S. (2003, April). Autonomia: An autonomic computing environment. In Conference proceedings of the 2003 IEEE international performance, computing, and communications conference, 2003. (pp. 61–68). IEEE.
5. Akhare, R., Mangla, M., Deokar, S., & Wadhwa, V. (2020). *Proposed framework for fog computing to improve quality-of-service in IoT applications (In fog data analytics for IoT applications)* (pp. 123–143). Singapore: Springer.
6. Gheisari, M. (2012). Design, implementation, and evaluation of SemHD: A new semantic hierarchical sensor data storage. *Indian Journal of Innovations and Developments, 1*(3), 115–120.
7. Mangla, M., Satpathy, S., Nayak, B., & Mohanty, S. N. (Eds.). (2021). Integration of cloud computing with internet of things: Foundations. analytics and applications. New York: John Wiley & Sons.
8. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. *Data Engineering and Intelligent Computing*.
9. Yadav, A. K., Tomar, R., Kumar, D., Gupta, H. (2012). Security and privacy concerns in cloud computing. *Computer Science and Software Engineering*.
10. Deokar, S., Mangla, M., & Akhare, R. (2021). A secure fog computing architecture for continuous health monitoring. In Fog computing for healthcare 4.0 environments (pp. 269-290). Springer, Champions.
11. Abuseta, Y. (2019). A fog computing based architecture for IoT services and applications development. arXiv preprint arXiv:1911.02403.
12. Ganek, A. G., & Corbi, T. A. (2003). The dawning of the autonomic computing era. *IBM Systems Journal, 42*(1), 5–18.
13. Stoilov, T., & Stoilova, K. Autonomic computing applications for traffic control.
14. White, S. R., Hanson, J. E., Whalley, I., Chess, D. M., & Kephart, J. O. (2004, May). An architectural approach to autonomic computing. International conference on autonomic computing, 2004. Proceedings. (pp. 2–9). IEEE.

15. Chauhan, S. K. (2012). Autonomic computing: A long term vision in computing. *Journal of Global Research in Computer Science, 3*(5), 65–67.
16. Jaleel, A., Arshad, S., & Shoaib, M. (2018). A secure, scalable and elastic autonomic computing systems paradigm: Supporting dynamic adaptation of self-* services from an autonomic cloud. *Symmetry, 10*(5), 141.
17. Omer, A., Mustafa, A., & Alghali, F. (2014). Advantages of autonomic computing over cloud computing comparative analysis. *IOSR Journal of Electrical and Electronics Engineering, 9*, 56–60.
18. Furrer, F. J., & Püschel, G. (Eds.). (2017). *Autonomic computing: State of the art-promises-impact*. Dresden: Saechsische Landesbibliothek-Staats-und Universitaetsbibliothek Dresden.
19. Jimoh, F., McCluskey, T. L., Chrpa, L., & Gregory, P. (2012). Enabling autonomic properties in road transport system.
20. Exposito, E., Gomez, J., & Lamolle, M. (2009, November). Semantic and architectural framework for autonomic transport services. In 2009 computation world: Future computing, service computation, cognitive, adaptive, content, patterns (pp. 99–104). IEEE.
21. Boubin, J., Chumley, J., Stewart, C., & Khanal, S. (2019, June). Autonomic computing challenges in fully autonomous precision agriculture. In 2019 IEEE international conference on autonomic computing (ICAC) (pp. 11–17). IEEE.
22. Schlingensiepen, J., Nemtanu, F., Mehmood, R., & McCluskey, L. (2016). Autonomic transport management systems—Enabler for smart cities, personalized medicine, participation and industry grid/industry 4.0. In *Intelligent transportation systems–problems and perspectives* (pp. 3–35). Cham: Springer.
23. Anala, M. R., & Shobha, G. (2012). Application of autonomic computing principles in virtualized environment. First international conference on information technology convergence and services (ITCS 2012) (p. 203208).
24. Mangla, M., Akhare, R., & Ambarkar, S. (2019). Context-aware automation based energy conservation techniques for IoT ecosystem. In *Energy conservation for IoT devices* (pp. 129–153). Singapore: Springer.
25. Huebscher, M. C., & McCann, J. A. (2008). A survey of autonomic computing—Degrees, models, and applications. *ACM Computing Surveys (CSUR), 40*(3), 1–28.
26. Abeywickrama, D. B., & Ovaska, E. (2017). A survey of autonomic computing methods in digital service ecosystems. *Service Oriented Computing and Applications, 11*(1), 1–31.
27. Parashar, M., & Hariri, S. (2004, September). Autonomic computing: An overview. In *International workshop on unconventional programming paradigms* (pp. 257–269). Berlin: Springer.
28. Kephart, J., Kephart, J., Chess, D., Boutilier, C., Das, R., Kephart, J. O., & Walsh, W. E. (2003). An architectural blueprint for autonomic computing. IBM White paper (pp. 2–10).
29. Coutinho, E. F., Rego, P. A., Gomes, D. G., & de Souza, J. N. (2016, April). An architecture for providing elasticity based on autonomic computing concepts. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 412–419).
30. Singh, A., Juneja, D., & Malhotra, M. (2015). Autonomous agent based load balancing algorithm in cloud computing. *Procedia Computer Science, 45*, 832–841.
31. Singh, A., Juneja, D., & Malhotra, M. (2017). A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. *Journal of King Saud University-Computer and Information Sciences, 29*(1), 19–28.
32. Ghobaei-Arani, M., Souri, A., Baker, T., & Hussien, A. (2019). ControCity: An autonomous approach for controlling elasticity using buffer Management in Cloud Computing Environment. *IEEE Access, 7*, 106912–106924.
33. Nazir, S., Patel, S., & Patel, D. (2020). Cloud-based autonomic computing framework for securing SCADA systems. In Innovations, algorithms, and applications in cognitive informatics and natural intelligence (pp. 276–297). IGI Global.
34. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*, 1144–1148.

35. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*, 583–585.
36. Etemadi, M., Ghobaei-Arani, M., & Shahidinejad, A. (2020). Resource provisioning for IoT services in the fog computing environment: An autonomic approach. *Computer Communications*.
37. Kaur, M., & Kaur, H. (2019, February). Autonomic computing for sustainable and reliable fog computing. In *Proceedings of international conference on sustainable computing in science*. Rajasthan: Technology and Management (SUSCOM), Amity University Rajasthan.
38. Kayal, P., & Liebeherr, J. (2019, October). Poster: Autonomic service placement in fog computing. In Proceedings of the 2019 on wireless of the students, by the students, and for the students workshop (p. 17).
39. Zhao, Z., Schiller, E., Kalogeiton, E., Braun, T., Stiller, B., Garip, M. T., ... Matta, I. (2017). Autonomic communications in software-driven networks. *IEEE Journal on Selected Areas in Communications, 35*(11), 2431–2445.
40. Lam, A. N., & Haugen, Ø. (2018, May). Supporting IoT semantic interoperability with autonomic computing. In 2018 IEEE industrial cyber-physical systems (ICPS) (pp. 761–767). IEEE.
41. Khorsand, R., Ghobaei-Arani, M., & Ramezanpour, M. (2018). FAHP approach for autonomic resource provisioning of multitier applications in cloud computing environments. *Software: Practice and Experience, 48*(12), 2147–2173.
42. Singh, S., & Chana, I. (2015). Q-aware: Quality of service based cloud resource provisioning. *Computers & Electrical Engineering, 47*, 138–160.
43. Gill, S. S., Buyya, R., Chana, I., Singh, M., & Abraham, A. (2018). BULLET: Particle swarm optimization based scheduling technique for provisioned cloud resources. *Journal of Network and Systems Management, 26*(2), 361–400.
44. Singh, S., Chana, I., Singh, M., & Buyya, R. (2016). SOCCER: Self-optimization of energy-efficient cloud resources. *Cluster Computing, 19*(4), 1787–1800.
45. Bittencourt, L. F., Diaz-Montes, J., Buyya, R., Rana, O. F., & Parashar, M. (2017). Mobility-aware application scheduling in fog computing. *IEEE Cloud Computing, 4*(2), 26–35.
46. Kettimuthu, R., Liu, Z., Foster, I., Beckman, P. H., Sim, A., Wu, K., ... & Choudhary, A. (2018, June). Towards autonomic science infrastructure: Architecture, limitations, and open issues. In Proceedings of the 1st international workshop on autonomous infrastructure for science (pp. 1–9).
47. Srivastava, B., & Kambhampati, S. (2005, June). The case for automated planning in autonomic computing. In Second international conference on autonomic computing (ICAC'05) (pp. 331–332). IEEE.
48. Dimitrakopoulos, G., & Demestichas, P. (2010). Systems based on cognitive networking principles and management functionality. *IEEE Vehicular Technology, 5*, 77–84.
49. Exposito, E., Chassot, C., & Diaz, M. (2010, December). New generation of transport protocols for autonomous systems. In 2010 IEEE globecom workshops (pp. 1617–1621). IEEE.
50. Jain, A., & Kumar, R. (2017). Critical analysis of load balancing strategies for cloud environment. *International Journal of Communication Networks and Distributed Systems, 18*(3–4), 213–234.

Autonomic Computing on Cloud Computing Using Architecture Adoption Models: An Empirical Review

R. S. M. Patibandla, V. Lakshman Narayana, and Arepalli Peda Gopi

1 Introduction

Autonomous computing systems are capable of managing themselves and adapting to changes continuously according to changing or rigid policies and goals. These systems can perform management activities dependent on conditions in the IT world they experience or hear. Instead of implementing IT professionals' management activities, the system observes something and acts accordingly. It encourages IT staff to work on high-value projects as technology handles a more competitive market. Advances in networking, computation, and mobile technologies and tools have culminated in the proliferation of networked applications and information services spanning many aspects of our lives. These are highly complex, heterogeneous, and dynamic applications and services [1]. However, a huge number of independent computing and communications services, data storage, and sensor networks are interconnected worldwide in the basic information infrastructure (e.g., the internet), and it is similarly massive, heterogeneous, diverse, and complicated. The combination has resulted in application development, configuration, and management complexities breaking current computing paradigms based on static requirements, conduct, interactions, and compositions [2]. Applications, programming environments, and information infrastructures are therefore increasingly becoming weak, dangerous, and unmanageable. This involved work on a new system and application design paradigm focused on techniques used by biological systems to address similar challenges, such as sophistication, instability, and ambiguity, a concept known as autonomous computing [3].

R. S. M. Patibandla (✉)
Vignan's Foundation for Science, Technology and Research, Vadlamudi, India

V. L. Narayana · A. P. Gopi
Vignan's Nirula Institute of Technology & Science for Women, Guntur, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_11

The autonomous computing paradigm was inspired by human autonomy. The overarching goal is the development of computer and software systems and applications that can be controlled in accordance with human guidance [4]. Addressing the major challenges of autonomous computing calls for scientific and technological progress across a range of fields, as well as new paradigms and software and system architectures that encourage effective integration [5].

This chapter introduces autonomous computing, its challenges, and opportunities. In this chapter, we first have a description of the nature of the nervous system and use it to inspire autonomous computation. We also address the main problems of autonomous computing and include a detailed review of current autonomous computing systems and applications [6].

2 Literature Review

2.1 Self-Management Attributes and Capabilities of Autonomic Computing

In an autonomous environment, system components — including built-in control loop functions from hardware (e.g., storage devices, desktop computers, and servers) to software (e.g., operating systems, middleware, and business applications) [7]. Since these control loops have the same basic elements, their roles may be separated into four specific interconnected types of control loops [8]. These categories are considered the system component attributes and are defined as:

- Self-configuration—May adjust to evolving conditions dynamically – Materials for self-configuration dynamically respond to environmental shifts by IT practitioners' policies [8]. Such modifications could include the deployment or removal of new components or dramatic changes to system characteristics. Complex transition helps ensure consistent IT system efficiency and profitability and contributes to development and versatility for business [9].
- Self-healing—Can detect, diagnose, and respond to disturbances Auto healing components can detect system malfunctions without disturbing the IT environment and initiate policy corrections [10]. Corrective steps may include the modification of a product's condition or modifications to certain environmental components [7]. The whole IT infrastructure is getting increasingly robust, because routine processes become less prone to collapse [11].
- Self-optimization—Can track and adjust services dynamically. Self-optimization modules can be customized for end-user or company requirements. The reallocation of resources, for example, to respond to changing workloads, might lead to improved overall use or the timely completion of certain business transactions [12]. Self-optimization helps both end-users and business customers to achieve a high standard of service.

  When a program does not completely use its allocated computational power, it is not easy to turn excess storage space into less priority without self-optimizing

features. In these instances, customers have to buy and maintain a separate infrastructure for each application to meet the application's most demanding computing needs.

- Self-optimization—It is capable of predicting, recognizing, identifying, or defending against risks from all over the globe. Self-protective components can monitor and take corrective measures to mitigate their risk. Unauthorized entry and usage, ransomware intrusion and dissemination, and denial of service assaults may be aggressive behaviors. Self-protective technologies allow businesses to implement protection and privacy policies consistently.

With these qualities, the activities that IT specialists currently have to conduct in order to install, heal, optimize, and secure the IT infrastructure can be automated [13]. Systems management software can then perform a variety of system actions through these integrated control loops.

Components in the self-management framework may only change within their reach. For example, a self-optimizing autonomous server manager can only automate the server's functionality. However, this is not the only place in the IT world where self-management is feasible. The functions related to control loops that customize, restore, automate, and secure can also be included in the best practices and processes that the IT company uses.

2.2 Self-Sufficient Device Skills

There are many reports on the introduction to current networks of autonomous computing capability. In a commercial software project, Mulcahy et al. [14] identified autonomous computing capabilities in the legacy order placement framework. The resulting system can monitor itself, customize itself, and repair itself. The autonomous device capability decreased the human activity involved in order-making substantially, which lowered the costs of human error in machine operations. Mulcahy et al. [15] have suggested an alternative to the existing regulatory program by incorporating autonomous device capacities. The architecture focused on service was used to attach a new autonomous dimension to the original design. Their strategy has lowered human labor expenses for screening instructions and also minimized screening errors. Amoui [16, 17] proposed a strategy to incorporate autonomous computing capabilities in the evolution process to current applications. The method sought to accomplish self-adaptive specifications by way of a coevolution paradigm that applied a sequence of transformations to the autonomic specifications of the initial operating program. Our study is special and distinct from the aforementioned research. Although the current research focuses on the architecture nature and parameter tuning algorithms, our thesis reflects on the complexities of software development, such as training, to apply autonomous computing capability to established broad nonautonomous software systems. In particular, we concentrate on the problems we encountered through architecture

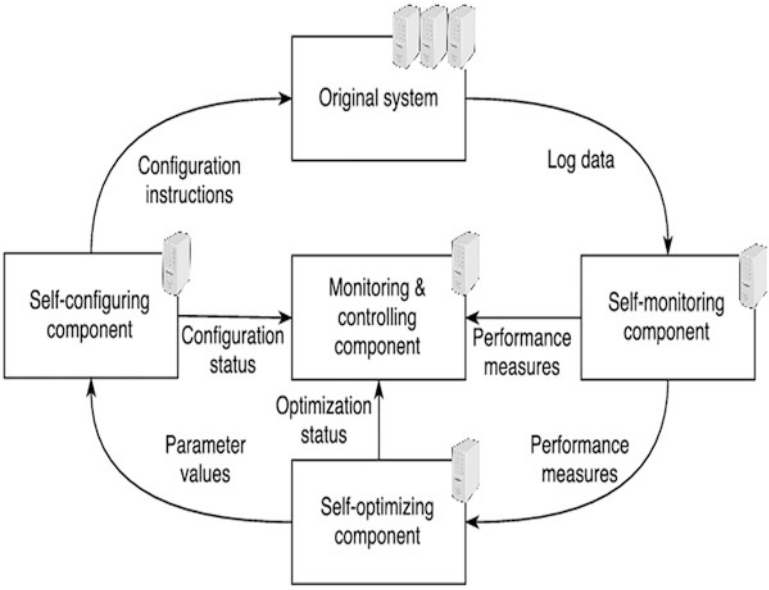

Fig. 1 Capabilities of autonomic computing

(e.g., how the initial program should have limited impact), deployment (e.g., how to prevent so many code changes), and testing (e.g., robustness testing) of the new autonomous computing capacity (Fig. 1).

2.3 Autonomic Computing Architectures

The design principles in this proposal describe a popular method and vocabulary for autonomous computing systems for self-management. Autonomous principles in machine architecture include a framework to analyze, evaluate, and contrast methods utilized by multiple vendors to provide autonomy in an IT environment.

Autonomous operating device organizes a computer framework automatically in layers and sections displayed in Fig. 2. Such components are linked by business service bus models, which allow components to operate in accordance with traditional structures such as web services. The bus incorporates the various building blocks of the program, including:

- Managed capital touchpoints.
- Information outlets.
- Self-employed administrators.
- Manual workers.

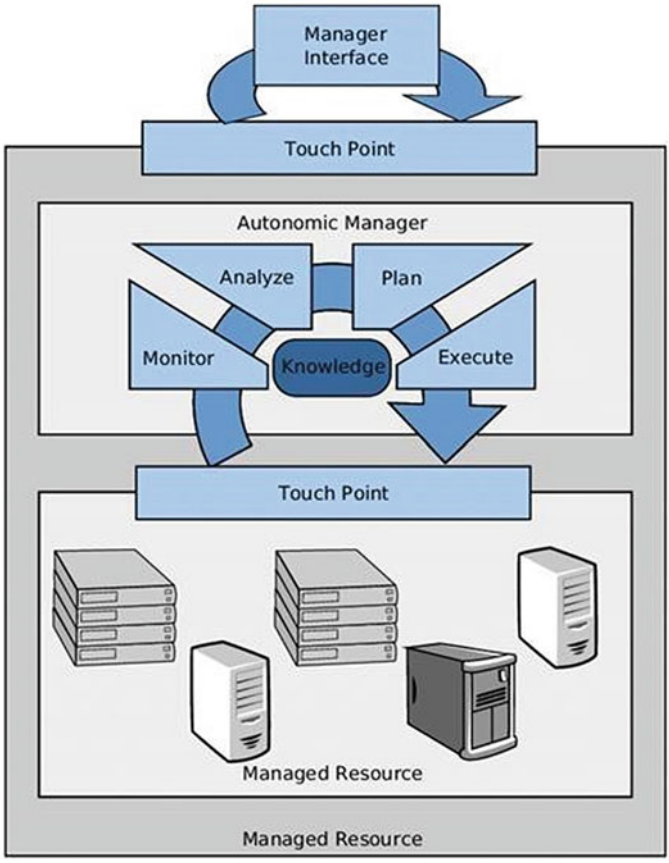

Fig. 2 Autonomic computing system

3 Adoption Models and Requirements of Autonomic Computing

Autonomous computing provides an efficient way to reduce device management sophistication, but early autonomous computing technologies are primarily used for physical resource control for the dispersed heterogeneous world. The cloud ecosystem did not contain many limiting variables such as large-scale virtualized systems, service level agreements, multiple frameworks, and complex shifts in the implementation process. The original Autonomous Computing Application framework cannot be used in the cloud environment directly [18]. For instance, on-demand configuration of monitoring software typically requires in the virtual layer to differentiate monitoring applications and resources. The standard MAPE loop must also be revamped to suit the cloud. As a result, cloud platforms have

virtualization, mobility, and restrict complexity characteristics, and the estimation of their protection is one of the most critical issues. This working framework uses the odd data mining concept as a guide, based on the topic of cloud health. It comprises four self-regulatory elements that can transmit information. Figure 2 displays the development structure. It consists of five modules, including a monitoring module for the network, a data analysis module, a response strategy module, an implementation module for the system, and a knowledge base or a virtual machine.

3.1 Plan for Cloud Protection Management

We also built an autonomous control methodology focused on a stand-alone computing concept to address the protection issue of the cloud network. This architecture comprises primarily of five modules, a module for network management, data processing, a solution plan module, a module for device execution, and a knowledge base. The management of services may be in two ways.

1. Active mode: The operating node includes a resource management portion and the virtual machine monitor gathers status data from a running virtual machine. By submitting their monitoring information, the monitor triggers the master node.
2. Master mode: The master nodes end request for the working node; the data tracking job node then returns to the theme node. A control device is found in a physical server, computer system, or other units. This is used to capture and securely preserve tracked device data at all rates. The data analysis framework provides a data correlation graph for analyzing the ance degree metric association, based on the traditionally tracked results. This uses the PCA (principal component analysis) to measure the Eigenvector of managed data and evaluate the data source's linear regression function in the cloud computing setting to determine device irregularities quantitatively. The item to be tracked in the next stage is selected by the solution approach modules depending on the importance of the event. The Poisson method is used to build a stability model for the operating program, and the risk of device failure is estimated on the basis of a testing duration irregular. The Device Development Module is the control agent for dynamic tracking entity modification and process monitoring. Knowledge base is a record of operation during the learning of load patterns and the corresponding own vectors to show the normal system operation. Real-time knowledge sources for the cloud storage world ought to follow the polling strategy-cyclical or event-driven path. Two instances are related to in periodic mode. In the first example, the operating nodes send their supervised details to the master node daily. In the second case, the node that is the main monitoring resource component is sent.

Daily demands for work nodes; then these work nodes collect knowledge and input on the master node. The event-driven way refers to the use of the old work nodes.

Several events are produced, and each event will be created by tracking and comparing the corresponding terminal resource status for the search. If the difference is larger than the level defined between the two cases, the function becomes either active or inactive [19].

3.2 Monitored Data Processing

The data processing research is a systematic, coordinated operation in the cloud infrastructure system that must be integrated with data management that reviews. It also creates a paradigm for data management and analytics, including data selection, data preprocessing, data interpretation, storage, and other elements. The original data stream is classified and the original data is generated after data collection. This is accompanied by the preprocessing method for editing the initial results. Ultimately, data collection and retrieval components derive valuable knowledge from mass data [20]. Users can use these procedures and software to directly use the data. This model of data acquisition and analysis achieves an integrated data process. Two important components of this model are data collection and storage analysis that provide customers with meaningful information.

Data collection: Data processing consists of three phases in the cloud storage environment, data extraction, data filters, and data classification. The initial data flow deals with relevant consumer data across these three steps [19]. After the data is captured, some useless information has to be filtered and removed and the rest categorized. Now the preprocessing component data description.

Application preprocessingQUERY: Data preprocessing encompasses a number of processes, including data cleaning, data integration, data transformation, and data decrease [20]. After choosing the correct attributes from the data mining attributes from the original data, the selection method concept would apply as explicitly as possible to attribute names and attribute values. It can be either information processing or software interpretation. The data collection layout shall be configured with the correct functions and conditions. To grasp the anticipated performance, firstly, let's talk about the data processing process. The initial model must also be measured and verified. The configuration is essentially modified according to the testing tests. Furthermore, the process for collecting data must be realistic. Finally, the causal relationship between the data is revealed, and it is statistically significant in the data collected. The development of the data model must also be constantly enhanced.

1. The changing window period shall be; several measurements are obtained for the tracked data as x = (x1), x2, ..., xm) including the symmetric for increasing data collected. The service and repair workers should respect their needs appropriately. In this case, m is a positive integer; xi is the metric number. The tracked data slides in order of time through the sliding glass. Monitored data in the Anm matrix sliding window (n rows and m columns).
2. Each Anm column shall have an average of 0 and a variance of 1. Zi = (xi−ui)/first, ui is the pattern of the dataset for the ith section, and 5–007i is the standard deviation from this range.
3. The covariance matrix is: C = [Covariance matrix is obtained. F.-D. D. 2 sts = 2 sts].
4. Measure the own vector of C, the concept of data distribution indirection.
5. The survey will be repeated nr times, or [0,1], and is a replication of the actual sample size, as fresh data is tracked to expand the effect of outliers on the key course of the shift. The ratio of matrices modified is as follows: A = A {xt, xt, ..., xt}.
6. Media and matrix update: u = (u + rxt)/(1 + r). Check the central directory patented vector: ut = −populate Au-populate.
7. Determine the anomaly of new results, using the similarity of cosine. The less related the two main Eigen vectors are, the greater the variance and the greater the abnormality. This text explains the anomaly degree: AutoCos = ut ... (4) "Approval" in this type reflects the substance. Collection of records: Information protection is the most critical compliance problem to tackle to maintain data privacy.

When a conventional mining algorithm is used to gather regular irregular details, irregular knowledge regarding recurrent deviations cannot be detected and regular data mining anomalies are a big concern. The data interpretation framework determines the link between input and evidence dependent on the past data observed.

Metrics shape the metric graph such that the metric can be analyzed. An anomaly data mining approach based on an optimized chaos algorithm is suggested. To measure device irregularities, the sensor data's own variable is measured. Next, the data source is merged into the least partially square form to be cleaned in the cloud processing setting and a nondimensional data matrix and a normalized aspect vector are collected. The matrix and the vector are described by analyzing variables and determinants predicting anomaly data and extracting principal component analytics to evaluate the linear regression equation of the data source in the cloud storage context.

1. The measured results are compressed into slightly smaller squares such that it is feasible to clean up the initial results with nondimensional details. The rule of xpotami=xij-xj SjS(D), ypotamia=yi-y Sy Sy Countable SjS(D). (5) In the case here, xij status data source space in the cloud-based computing world with inherent mode function, Sj reflect the weight vector of any data in the cloud computing environment, Sy stands for the steady-state likelihood of global data in the cloud computing environment.

2. We calculate in the next step the standardized data matrix, the standardized dimensional vector, and the database provided for the main component analysis. We obtain a normalized m as well as an order matrix representing X0 and a vector representing Y0, as following: Y0 = and i-n as well [y1-n order 2]. The main elements of X0 are x- = x validated by 11 names x validated by 1n. d. (6) Y1-approximately, y2-appropriately, ym-appropriately, and x-approximately 11, x- approximately 1, x-approximately 1N, x-approximately1, and x-roughly mn.
3. X0 and y0 are specified as the effects of the standardization observation value in the cloud computing environment for the predictive variables and as the determinant of frequent abnormal data. The following two formulae are used to delete Y0 and X0 from Y1 and X1, which allow X1 a complete representation of Y1 and introduce the key interaction variable.

   Where P1 is the typical abnormal data in cloud computing, XT 0 reflects the own vector data of the frequent anomalies in the cloud computing environment; P1 reflects the similarity of characteristic values for frequent abnormal data, and t1 represents the time complexity of matrix X0 decomposition.
4. The source data equation must be defined in the cloud storage system after extraction of the key components. Word is: (x-pop, y-pop) = X 1 is split into a 1-pop 1x1 + ′ 2 + x 2-pop+p + xp-pop. (8) (x) is the Cloud computing data linear regression coefficient. (x) is the relevance between the dependent variable and the normal data are offered by each variable, and (x) the correlation between the frequent abnormal data variables [19].

3.3 Political Response Method

Internet connectivity infrastructure is the essence of resources offered by the cloud storage network. Most existing models use the stochastic flow model of the traditional Access-Poisson telecommunication network as the basis for scheduling optimization of device resources. The Poisson process is an independent class of random events in which the time interval of events is regarded as independent random variables, equivalent to an update process [21]. P(N(t) − N(t − s) = n) = e − extremism[(t − (t − s))][βt − 5007(t − s)]nn is usually represented as a Poisson method! (9) In (9) N(t)is the amount and severity of events between 0 and t. The number and access frequency of internet access services can be seen as at times t; N(t) − N(t − s) is the rise in the amount of internet access services at this period t − s, t] [19].

If the data module is abnormal, it is sent to the appropriate strategy module. The Poisson propagation method is the core reliability engineering fault prediction pattern. Historical fault details are conventionally used to estimate the date of the next breakdown. This paper, therefore, strengthens the model and adds error evaluation to remove historic fault data and estimate the next failure date of the device. Therefore, the simultaneous mistake may be replied to in a timely fashion by determining gradient anomalies; the testing period is reduced automatically to a minimum. The total error should be calibrated to the irregular period stage. We

describe a machine malfunction like F(t) = w. So, you should measure the following failure's time interval: t = −ln(1 − w)/5007 to change the existing tracking duration. T = {Tβ,0} in st < β − ln(1 − st)μ + e, β in class in st < T5–007, in st < st in st in line1. (10) The minimal monitoring period in (10) is Tβ; average monitoring period is T5007; modification parameters are 5–007 and e. According to the task review, if the monitoring period and the minimum monitoring duration, the irregular monitoring duration is reduced by a rise in the frequency and the degree of shortness of the irregular monitoring as well as the number of increased rates.

3.4 Autonomic Cloud Computing Properties

Cloud autonomy will allow businesses by automating their management with a set of business decisions to make the best use of cloud computing. Organizations do not need to expend the time to automate cloud technology protection, usage, and costs – all in line with business objectives that describe any element of administration. All this is achieved automatically.

That form of automation is known as process automation. A company will need to set up an internal mechanism with appropriate governance rules and then focus on it to track and enforce the required improvements to get the cloud network back into line with the policy.

4 Autonomous Computing Gains

From now on, it will be obvious that software autonomy provides several advantages for organizations that plan to implement applications to handle their cloud environments. Below are the key benefits of cloud technology autonomy computing.

Use—Companies will be able to plan for an automated shutdown of idle or long-term infrastructure to support their policies (e.g., the shutdown of idle development infrastructures that run for over a week).

The automatic transfer of data to another area will help business service-level agreements (BSLAs). Such automatic replication may be used for data redundancy from small to small.

Cost—Companies will use cloud autonomists for automated reserved capacity purchases according to organizational or functional needs. A further downside could be that the workflow from one cloud provider to another could be transferred in pursuit of cost-efficiency.

Quality—Organizations may use cloud autonomy to automate the type of machine used to support the activity of nonhorizontal scaling workloads.

Security—Companies will benefit from the framework for modifying the network or endpoint security dynamically to those consistent with established business policies.

Clearly, cloud sovereignty is something that should certainly be of interest to companies. Experts agree that future challenges of cloud infrastructure management will continue, so that automating at least part of the process utilizing software autonomy can become the right solution for many companies.

This segment discusses a case examining autonomous cloud management for workflow systems for spatial-temporal data processing for online dengue fever prediction and its delivery to clouds to show the significance and effect of Autonomous Cloud Management.

Dengue is a mosquito-borne infectious disease occurring particularly in tropical areas including South America and Southeast Asia. There are 2.5 billion individuals who live in tropical dengue areas around the globe, according to the World Health Organization (WHO), rendering it a significant foreign public health issue. In densely populated areas where the disease can spread quickly, this is further worsened. Therefore, Singapore's forecasting and control of dengue is a highly significant public health issue [21], which led to the creation of predictive models for dengue disease spreading in the region. The application's data specifications provide multidimensional details, including recorded dengue events, environmental parameters, and regional details. Incidence data can be up to 100s of MB, and forecast data can be up to a few GB with ease. For instance, a single ECHAM5 climate model performance vector consists of 300,000 spatial points multiplied by 365,250 time intervals per quarter per scenario at regular resolution. The number of dengue incidences would be trackable by device consumers by day, week, month, and year from the 1960s to 2011. For visualization analysis, the period required for encoding, modeling, and interpolating data is roughly 30 min for analysis 1-day data on the Intel dual-core 2.93 GHz CPU and 4 GB ram. Therefore, the program needs to continuously delegate resources and maximize the application efficiency of cloud infrastructures (private, public, or hybrid clouds) to minimize processing time to enable for shorter time to time frame analyzes in real time to be of real value for dengue outbreaks. From the above, it is evident that autonomous cloud technologies are of vital importance to the goals of rapid dengue dissemination to activate health agencies. In this case study, we define our cloud-enabled workflow engine. A. The Cloudbus Workflow Engine [22] is an application to Grid-oriented Workflow Management. Cloudbus Workflow Management Program [23] enables commercial, public, and hybrid cloud workflows. It provided initial functions, such as a GUI-based overview of the process, program structure, cloud data transfer, and task preparation and administration. Therefore, automated control technologies have been expanded focused on iterative optimizations. Figure 2 offers a description of the Autonomous Workflow Management Model and its use in data processing systems. Quality criteria are achieved by splitting data into different parallel tracks and concurrently running these tracks on many virtual machines. The machine automates the output and determines an appropriate supply for usage and data management to accomplish this. Iterative optimization is designed for analytical workflow applications in which a subset of analytical tasks/functions is repeated during analytics and a sort of "loop" is created in the workflow. The workflow engine profiles early task execution when such loops are detected in applications,

and stores information about their execution time. Such profile knowledge is used for the optimum expense and time provisioning purposes.

A framework that dynamically increases up and down the tools offered to fulfill app output criteria constantly improves the analytics platform. For dynamic provisioning, the optimization issue in the scheduler consists of an initial S timetable, which assigns all workflow activities from the workflow graph G to the server, taking the previous limitations into account. Time t(S) and cost m(S) shall be specified respectively as the completion period and the monetary expense of Schedule S. The iterative optimization strategy is intended to build an optimum Sopt plan to achieve tmin(SG) and mmin(SG). As NP-complete is the question of mapping workflow activities to dispersed heterogeneous sources to achieve multi-objective optimization, we have suggested a heuristic algorithm for achieving a suboptimal solution and optimizing efficient workflow performance. Figure 3 defines this algorithm. The autonomous adaptive workflow model architecture enables the device to choose the most suitable resources depending on customer criteria (e.g., changes duration, prices, etc.) and plan data in private resources and is immune and private and accept failures. Cloud services and process activity management are dynamically carried out on a contractual basis and the program schedules assignments to tools that can maximize output in terms of overall time while meeting future budget requirements for application execution. Finally, it is worth remembering that.

Automated implementation of dengue fever workflows is just one potential case for the autonomous monitoring operation of clouds. When cloud services extend across a broad variety of sectors, such as eHealth, e-Science, e-Government, and e-commerce, the need for autonomous software management is distributed across the world [24].

Those are the lands. The basic concepts and technical characteristics of such autonomic systems, independent of the implementation scope, must therefore obey the architectural elements described in this paper.

5 Performance Evaluation

We present an assessment of the workflow engine's automated iterative optimization function. The theoretical testbed displayed in Fig. 4 contains a hybrid cluster of 45 g/h (hyper-threaded) 2.93 GHz processors and 96 GB of memory, including 48 Ubuntu CentOS 5.8 virtual machines of 2 to 4 g/h and 4 GB of memory, based at the A*STAR Institute (Singapore). This local network is augmented by 25 wide Amazon EC2 computing units deployed throughout the Asia Pacific (Southeast) region (2 kernels with 2 ECU and 7.5 GB of mémoire). The framework for the study is the program for dengue fever prediction, which utilizes recorded dengue cases and environmental evidence between 2001 and 2010 [25–27] (Fig. 5).

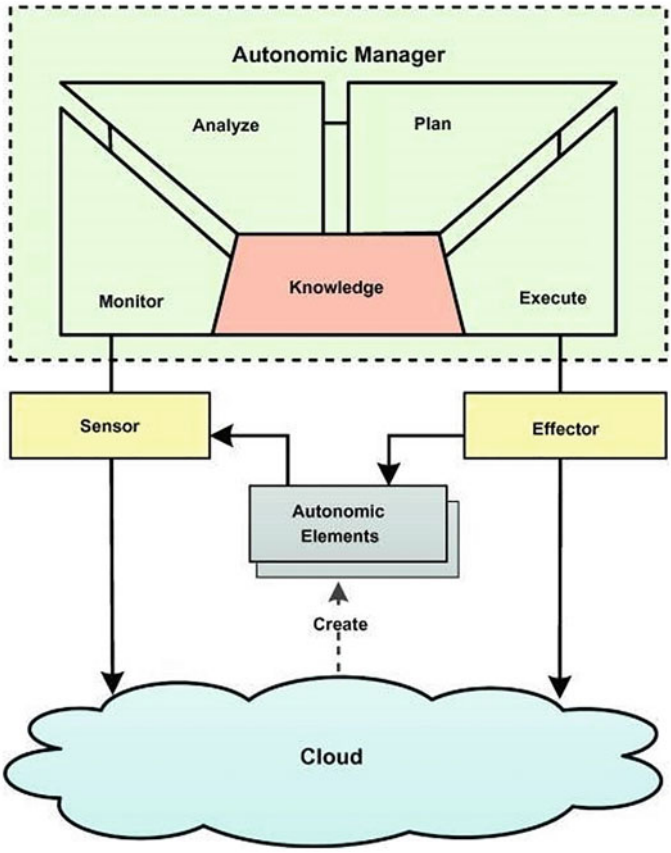

Fig. 3 Relationship between cloud and autonomic computing

The predictive model is based on a sliding window of parameter variables regularly modified by matching actual dengue cases of predicted tests. Such an integrated research model can be traced in Fig. 6 to the workflow. The iterative scheduling algorithm actively tries suboptimal approaches with data from previous workflow iterations. The iteration process is between the H and A activities as seen in Fig. 6. The iterative loop re-executes tasks labeled from B to G as each new iteration commences, with details linked to a particular time used as input activities. By the completion of each phase, the configuration method measures the estimated execution period and the expense of managing the current execution tools. When increases in the number of resources available will contribute to major adjustments in make-up or prices, this can raise or decrease the number of resources available. This helps the framework to optimize and modify distribution and preparation according to process and implementation setting characteristics.

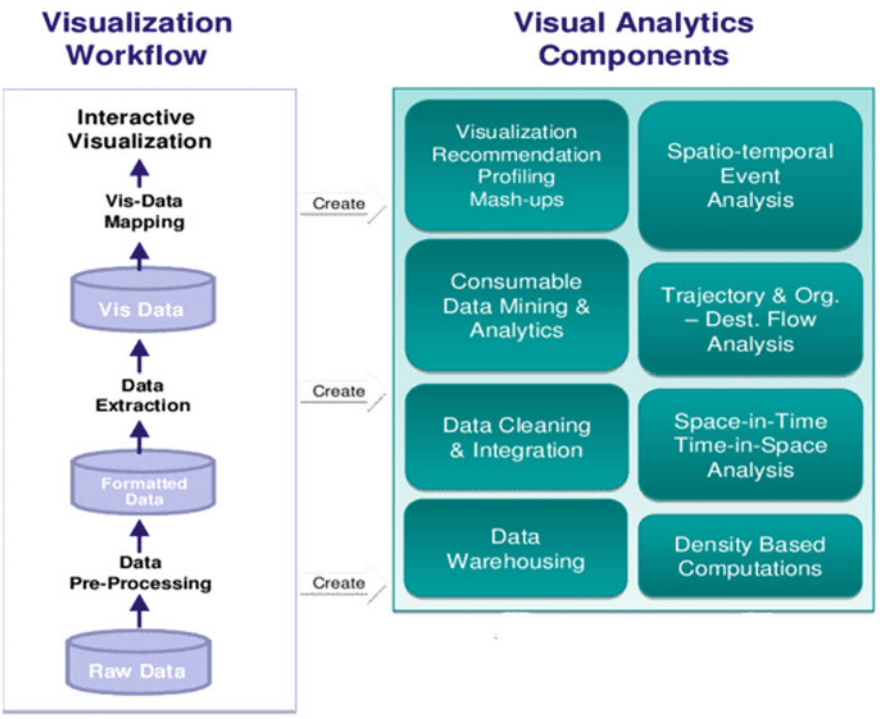

Fig. 4 Spatial-temporal analysis model

Step 0. To execute the tasks, launch the cloud resources.
Step 1. Use a greedy algorithm to reduce cost and resource constraints.
Step 2. Apply an initial plan to use the assigned machines fully by planning additional tasks for resources, as long as maquillage does not increase.
Step 3. Analyze whether the public cloud reduction enables the process to be done at the same time. If so, use a cheaper and simpler form of case.
Step 4. Run the tasks at the program nodes.
in results (makespan). The output of the application execution.

Fig. 5 Algorithm for iterative optimization

Figure 7 demonstrates the effects of differences in workflow system tools in the various iterations of prediction model implementation. Since collecting details regarding the real execution period of the tasks in the first step, the number of services offered has been changed to simplify the tasks into fewer cloud services. Additional changes have been made between versions 2 and 3. The workflow engine's automated iterative optimization function ultimately lowered the run-time

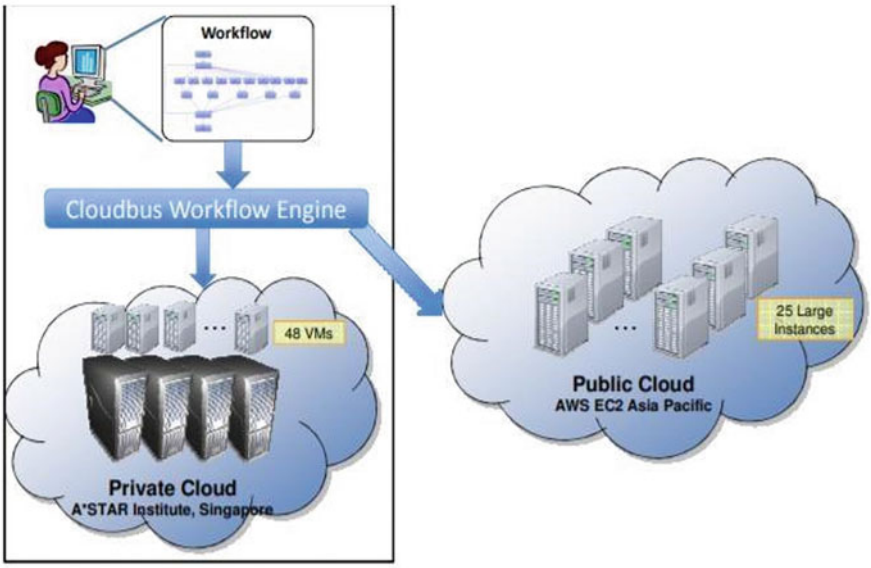

Fig. 6 Testbed

by 48% and expense of public cloud usage by 70% relative to the selfish approach for the configuration and scheduling of cloud workflow applications (Fig. 8).

6 Conclusion

In this chapter, we presented the autonomic computing that is enthused as a result of genetic systems like the nervous system of humans to develop the applications. Numerous research exertions motivated on empowering the autonomic stuff report four core areas, self-healing, self-protection, self-configuration, and self-optimization. And also, adoption models like Plan for Cloud Protection Management, Monitored Data Processing, and Political Response Methods. And also, the predictive model is based on a sliding window of parameter variables regularly modified by matching actual dengue cases of predicted tests.

.

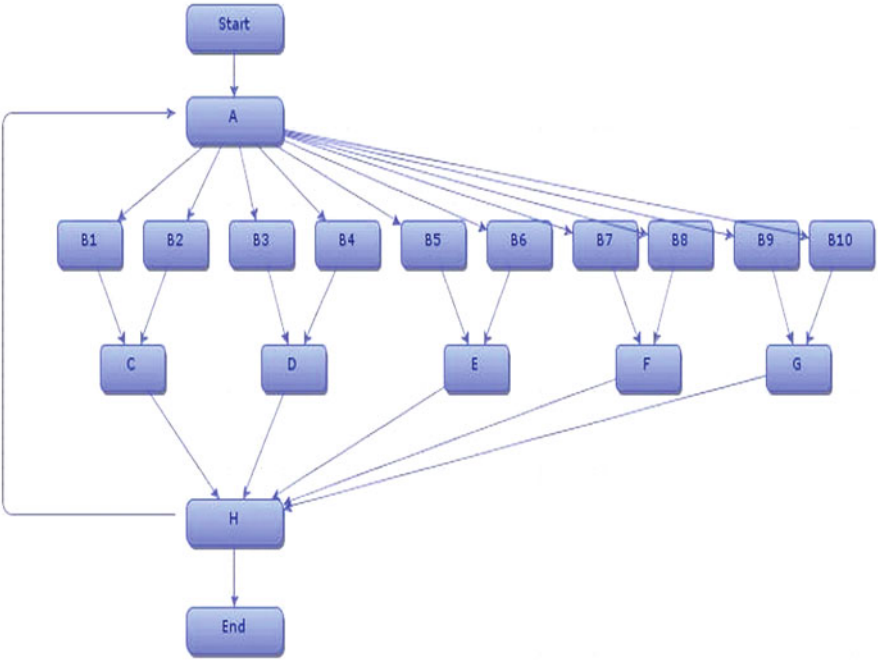

Fig. 7 Iterative dengue fever predictor software process software used in research. The repetition takes place between tasks H and A as seen in the figure.

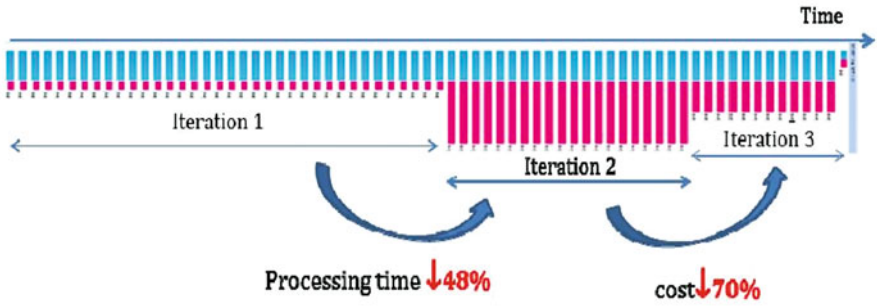

Fig. 8 Map. Impact of iteration optimization in the dengue fever prediction model

References

1. Kumar, M., & Sharma, A. (2017). An integrated framework for software vulnerability detection, analysis and mitigation: An autonomic system. *Sādhanā, 42*(9), 1481–1493.
2. Ibrahim, Y., Adamu, A., Abdulrahman, S., & Rilwan, A. (2017). Autonomic cloud computing: A review. *International Journal of Computer, 26*, 99–104.
3. DehrajP Sharma, A. (2019). An empirical assessment of autonomicity for autonomic query optimizers using F-AHP approach. *Applied Soft Computing, 90*, 106137.

4. Singh, S., Chana, I., & Singh, M. (2017). The journey of QoS-aware autonomic cloud computing. *IT Professional, 19*(2), 42–49.
5. Berekmeri, M., Serrano, D., Bouchenak, S., Marchand, N., & Robu, B. (2016). Feedback autonomic provisioning for guaranteeing performance in mapreduce systems. *IEEE Transactions Cloud Computing, 6*(4), 1004–1016.
6. Nazir, S., Patel, S., & Patel, D. (2017). Autonomic computing meets SCADA security. In *Proceedings of 2017 IEEE 16th international conference on cognitive informatics and cognitive computing, ICCI* CC 2017* (pp. 498–502). London: London South Bank University.
7. Vieira, K., Koch, F. L., Sobral, J. B. M., Westphall, C. B., & de Souza Leão, J.L. (2019). Autonomic intrusion detection and response using big data. IEEE System Journal. https://doi.org/10.1109/JSYST.2019.2945555
8. Farahani, A., Nazemi, E., Cabri, G., & Capodieci, N. (2017). Enabling autonomic computing support for the JADE agent platform. *Scalable Computing: Practice and Experience, 18*. https://doi.org/10.12694/scpe.v18i1.1235
9. Dehraj, P., Sharma, A., & Grover, P. S. (2018). Incorporating autonomicity and trustworthiness aspects for assessing software quality. *IJET, 7*(1.1), 421–425.
10. Tahir, M., Ashraf, Q. M., & Dabbagh, M. (2019). Towards enabling autonomic computing in IoT ecosystem. In: 2019 IEEE international conference on dependable, autonomic and secure computing, international conference on pervasive intelligence and computing, international conference on cloud and big data computing, international conference on cyber science and technology congress (DASC/PiCom/CBDCom/CyberSciTech) (pp 646–651). IEEE.
11. Jamshidi, P., Sharifloo, A. M., Pahl, C., Metzger, A., & Estrada, G. (2015). Selflearning cloud controllers: Fuzzy q-learning for knowledge evolution. International conference on cloud and autonomic computing (ICCAC) (pp. 208–211).
12. Singh, S., Chana, I., & Buyya, R. (2015). Agri-Info: Cloud based autonomic system for delivering agriculture as a service. arXiv preprint arXiv:1511.08986.
13. Balaram, V. V. S. S. S. (2016). Self directing and decomposition administration of wireless sensors networks. *International Journal Engineering Sciences and Research Technology, 5*(3), 124–129.
14. Aygalinc, C., et al. (2016). A model-based approach to context management in pervasive platforms. IEEE international conference on pervasive computing and communication workshops (PerCom 2016) (pp. 258–264).
15. Danelutto, M., De Sensi, D. & Torquati, M. (2016). A power-aware, self adaptive macro data flow framework. 9th international symposium on high level parallel programming and applications (HLPP).
16. Sangani, S. P., & Rodd, S. F. (2016). Injecting autonomic computing into legacy systems: A survey. *Bonfring International Journal of Software Engineering and Soft Computing, 6*(Special Isuue), 230–233.
17. Lynn, T., et al. (2016) Cloudlightning: A framework for self-organising and self-managing heterogeneous cloud. 6th international conference on cloud computing and services science
18. Hameed, A., et al. (2016). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing, 98*(7), 751–774.
19. Zhang, J., Wu, Q., Zheng, R., Zhu, J., Zhang, M., & Liu, R. (2018). A security monitoring method based on autonomic computing for the cloud platform. *Journal of Electrical and Computer Engineering, 2018*, Article ID 8309450, 9 p. https://doi.org/10.1155/2018/8309450
20. Bournemouth University. (2017). Best practice design for autonomic applications in the cloud. PhD studentship project description.
21. Ramezani, F. (2016). Autonomic system for optimal resource management in cloud environments. Ph.D Thesis, University of Technology, Sydney.
22. Dehraj, P., & Sharma, A. (2020). A review on architecture and models for autonomic software systems. *The Journal of Supercomputing*. https://doi.org/10.1007/s11227020-03268-0
23. Aguilar, J., Jerez, M., Mendonça, M. et al. (2020). Performance analysis of the ubiquitous and emergent properties of an autonomic reflective middleware for smart cities. *Computing*. https://doi.org/10.1007/s00607-2000799-5

24. Sharma, A., & Dehraj, P. (2015). Complexity assessment for autonomic system using neuro-fuzzy approach. In *CSI- conference*. Delhi: Springer.
25. Jain, R., Sontisirikit, S., Iamsirithaworn, S., et al. (2019). Prediction of dengue outbreaks based on disease surveillance, meteorological and socio-economic data. *BMC Infectious Diseases, 19*, 272. https://doi.org/10.1186/s12879-019-3874-x.
26. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*, 1144–1148.
27. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*, 583–585.
28. Farahani, A., Nazei, E., Cabri, G., & Capodieci, N. (2017). Enabling autonomic computing support for the jade agent platform. *Scalable Computing: Practice and Experience, 18*(1), 91–103.

Self-Protection Approach for Cloud Computing

Rishabh Malhotra, Bhupesh Kumar Dewangan, Partha Chakraborty, and Tanupriya Choudhury

1 Introduction

Nowadays the term "Cloud" [1, 17–21] is becoming so vast and important with the advancement of technology, which in itself means "unlimited." In terms of technology, it is a virtual storage where every individual, as well as companies, store their information in the cloud. Cloud storage is so big that it has millions and millions of data of each and everything. Cloud storage a part of cloud computing - delivers computing services with help of servers, databases, software, analytics, networking, and intelligence apart from storage to offer innovation, flexible resources, and economies of scale. Data stored on the cloud are not only helpful for people who are uploading but to those who are providing cloud services to make money out of such data in the cloud such as OneDrive, Google Drive, and so on [12]. There is a term known as Data Market Place where relevant information is extracted from the cloud depending on the requirement of the client and sold to them.

However, cloud computing provides the aforementioned advantages, but the most important question that arises is whether the cloud is secure? In other words, security concerns in cloud, the reason behind the question is as private information such

R. Malhotra (✉)
School of Law, University of Petroleum and Energy Studies, Dehradun, India
e-mail: malhotrarishabh15@stu.upes.ac.in

B. K. Dewangan · T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

P. Chakraborty
Department of Computer Science and Engineering, Comilla University, Cumilla, Bangladesh
e-mail: partha.chak@cou.ac.bd

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_12

as Adhaar Card, Bank Account Details, Passwords, and so on, of the individuals are stored on the cloud, as it is related to privacy. Some privacy issues are data violation; access, and data recovery. Not just individuals, additionally, the business who are holding their client basic data are in control for the data that will be passed on, and in this manner must know the sort of information to be redistributed, the legitimate specialists, and their reappropriating suppliers are dependent upon the danger of losing information and the class of the conveyed administration during the aftereffect of reappropriating a help into the cloud. Even, the Indian Law of Information Technology Act, 2000 provides a provision for body corporate to pay compensation if he fails to protect the data of any person [3]. This type of threat on data leads to cyberattacks, which is an ongoing debate.

A fascinating proclamation made by a renowned American cryptographer, and furthermore, one of the pioneers of public-key cryptography alongside Martin Hellman and Ralph Merkle:

> Cloud Computing is a Challenge to Security, But One That Can Be Overcome[4]

Security is based on the three most crucial components that are known as the CIA triad, (1) Confidentiality; (2) Integrity; and (3) Availability. Numerous solutions have been proposed by various researchers, scholars, and even the MeitY has issued a policy for providing a planned path for adoption of cloud services by the Government, which has been named, "MeghRaj policy." But still, issues related to cloud security remain. This chapter lists the parameters of different types of security available in the cloud; explores the cloud security issues; challenges involved; guidelines provided by the government to cloud service providers; techniques involved in cloud security; and applicability of Indian cyberlaw.

2 Literature Review

2.1 Secure: Self-Protection Approach in Cloud Resource Management [5]

In this paper, the author has talked about cloud computing and approaches taken, this paper is dependent on the guidelines and principles delivered by an existing resilience structure.

The essential theory is that in the close to prospect, cloud setups will be continuously exposed to novel assaults and different careless activities, for which standard mark-based discovering frameworks will be inauspiciously prepared and will consequently prove to be inadequate.

Cloud administrations can be isolated into three sorts dependent on the measure of control hang on by the cloud suppliers. Programming as a Service (SaaS) hangs on the most control and permits clients to get to programming usefulness on request, yet little else. Stage as a Service (PaaS) furnishes clients with a decision of execution condition, improvement devices, and so forth, yet not the capacity to deal with their own Operating System (OS).

2.2 *Cloud Computing Security: A Survey [6]*

Distributed computing is like creating an innovation model that excursions present mechanical and processing considerations into utility-like arrangements similar to vitality and water frameworks. Mists draw out a huge scope of benefits containing configurable figuring implies, budgetary saves, and administration adaptability. In any case, security and protection contemplation's square measure is demonstrated to be the primary hindrances to a decent appropriation of mists.

Cloud computing has not been sketched out in any case. The National Institute of Standards and Technology (NIST) laid out five basic qualities of distributed computing, specifically, expansive organization access, asset pooling, on-request self-administration, quick physical property or broadening, and estimated administration. Likewise, distributed computing is spoken to as a dynamic and at times stretched out stage to create clear virtualized materials to clients through the net. Distributed computing configuration comprises of three layers (1) programming bundle as an assistance (SaaS); (2) Infrastructure as a help (IaaS); and (3) Platform as an assistance (PaaS). The mists are seen as five half models that contain customers, stages, applications, foundation, and workers. The current mists square measure conveyed in one in each of the four preparing models, (a) network mists during which the physical framework is close by and overseen by a pool of associations; (b) nonpublic mists inside which the foundation is close by and overseen by a chose association (c) public mists inside which the physical foundation is close by and overseen by the specialist co-op; and (d) half and half mists which typify combos of the past three models shows cloud preparing models along the edge of their inward framework (IaaS, PaaS, and SaaS). Cloud preparing models have the same interior framework, take issue in their arrangements, and client access levels.

The acknowledgment of distributed computing worldview is continually developing. In 2010, the IT use in America to move to distributed computing arrangements was normal at $20 billion. Investigators consider that the cost drop factor in distributed computing will additionally hurry the acknowledgment of distributed computing in the public segments. With the huge development in distributed computing usage, the security included the consideration of analysts and experts yet has not recognized enough consideration.

At long last, we present and gauge the proficiency of the cutting-edge general countermeasures for cloud security assaults containing interruption discovering frameworks, self-sufficient frameworks, and united personality the board frameworks. We additionally feature the deficiencies of these frameworks that incorporate the high correspondence and calculation overhead and the introduction of productivity and inclusion.

2.3 MeghRaj: A Cloud Environment for e-Governance in India [7]

E-governance has gained a lot of importance in the last few years. In fact, much effort is being made in developing countries to gain momentum toward e-governance. India has moved way ahead and e-governance is now implemented almost everywhere. E-readiness is essential for any country to successfully implement e-governance. By readiness we mean infrastructural preparedness, data preparedness, human preparedness, and technological preparedness. E-governance has many benefits like increased transparency, reduced corruption, and more convenience for the citizens. In a country that has implemented e-governance the interaction between government and different stakeholders, that is, citizen, business, employee, and government, becomes smooth and easy. India has successfully implemented e-governance and is now moving toward the adoption of new technologies to be used in e-governance like cloud computing for still better delivery of services to the citizen.

In this paper, cloud computing has been given preference for almost all applications for the smooth, efficient, and effective working, which could take India to the Digital India path, but there is a major drawback in the form of loss of personal information, leakage of confidential information, and compromise of intellectual property, which the cloud providers will have to abide by through proper security policies and guidelines.

Appropriate measures need to be taken for the successful implementation of confidential data on the cloud and its security. Cloud computing is very much needed in e-governance as the management of the IT services on demand will be possible.

2.4 Cloud Computing: Different Approach and Security Challenge [8]

Cloud computing has made a ton of consideration and rivalry in the business and it is recognized as one of the main ten advances of 2010. It is a web grounded administration conveyance model that conveys web made administrations, processing and distribution center for clients in all market including medical care, monetary, and government.

The author has talked about the different types of cloud computing and their design. There is an organized view on various types of cloud and security challenges.

It also talks about security issues arising out of different types of cloud. It provides a contrast among dissimilar service providers on diverse cloud services like PaaS, IaaS, and SaaS. This examination shows that there are different sorts of clouds and the linked security challenges on each level.

3 Concept of Cloud Security

3.1 Meaning

In layman's terms, it means providing protection to their data stored in the cloud from illegal access and misuse of such data called Cloud Security. However, various cloud industries and security industries providing security to the cloud such as Vmware, Mcafee has defined the term "Cloud Security." The common definition of all industries is that it involves procedure and technology to save the cloud computing environment contrary to both internal and external cybersecurity intimidations within the cloud architecture.

Cloud computing—It is the delivery of IT (Information Technology) facilities over the internet, which is necessary for industries and governments seeking to expedite collaboration and remodeling. Prime practices of cloud security and security management are designed to preempt illegal approaches required to keep information and requests within the cloud secure from current and developing cybersecurity risks.

3.2 Importance of Cloud Security

As technology is developing, security risks are continuously growing and becoming more sophisticated and cloud computing is no less at risk than the on-premise environment. It is essential to work with a cloud service provider that proposes world-class security that has been specifically made for public groundwork.

There are lots of benefits offered by cloud security; some of them are as follows:

- Centralized security: Similar to cloud computing unify its application and info, cloud security also centralizes protection. Implementation of disaster recovery plans is done and actioned easily when they are coped in a single place.
- Minimize costs: Benefits of using cloud storage helps to abolish the need to invest in devoted hardware (such as external Hard Disk, USB Flash Drive). Not only does it reduce the expenditure of capital but also reduces administrative overheads.
- Reduced administration: Choosing the right cloud service provider or security platform helps to say "ciao" to manual security arrangement and almost constant to a security upgrade.
- Reliability: Using the correct cloud security procedures in place, users can carefully access data and applications in the cloud regardless of wherever they are or whatever device they are using (Fig. 1).

Cloud computing permits associations or organizations to work at scale, reducing technological expenditure and using agile systems that provide them the cut-

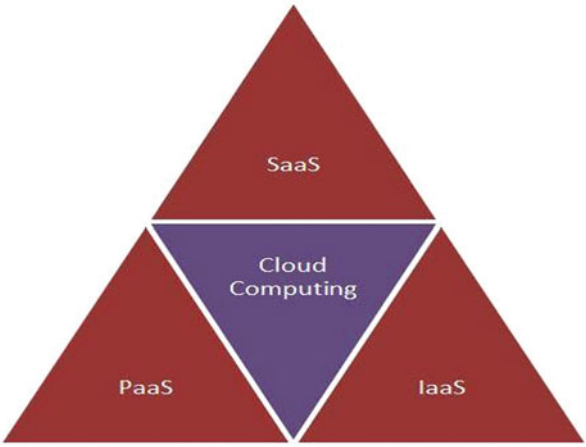

Fig. 1 SEQ Figure * ARABIC 1: key services of cloud computing

throat edge. Although it is compulsory that establishment must have total trust in their cloud computing security and all applications, information and systems are safeguarded from information theft, leakage, deletion, and corruption.

3.3 Types of Cloud Computing Services

There are three key cloud computing services (Fig. 1):

- SaaS—Software as a Service.
- IaaS—Infrastructure as a Service.
- PaaS—Platform as a Service.

1. SaaS: Full structure is Software as a Service that gives a product circulation model in which applications are presented by a seller or specialist organization notwithstanding made available to clients over an organization (web). Numerous chief undertakings, such as invoicing, bookkeeping, deals, and arranging all are frequently complete utilizing SaaS.
2. PaaS: Full structure is Platform as a Service that gives a stage and condition to allow engineers to build applications and administrations. This administration is introduced inside the cloud and got to by the clients over the web. It gives the stage to help application advancement. It incorporates programming backing and the board administrations, stockpiling, organizing, conveying, testing, teaming up, facilitating, and looking after applications.
3. IaaS: Full Form is Infrastructure as a Service that offers a basic support model of distributed computing close by PaaS. It offers admittance to figuring assets during a virtualized environment “the cloud” on the web. Additionally, gives

Fig. 2 SEQ Figure * ARABIC 2: SaaS

registering frameworks, for example, network associations, robotic worker space, load balancers, IP addresses, and transfer speed. At the end of the day, it is a total bundle for processing (Figs. 2, 3, and 4).

4 Securities, Challenges, and Architecture

4.1 Overview

Safety in cloud computing is a major alarm. Data in the cloud is necessary to be warehoused in a scrambled (encrypted) structure. However, it limits the user from accessing the public data directly. For this tenacity proxy and brokerage services are obligatory to be used. Encryption not only helps to protect transferred information in addition to the data stored in the cloud but also helps to guard data from any unapproved access, but it does not prevent data loss. Before deploying any resource to the cloud there is a necessity of security planning to examine different aspects of the resources that are as follow:

Fig. 3 SEQ Figure * ARABIC 3: PasS

- Select resources needed to transfer in the cloud and observe its sensitivity risk.
- Key models of cloud, that is, IaaS, PaaS, and SaaS – important to be considered for security at various degrees of services.
- Types of cloud, for example, public, private, network, and crossover should be thought of.
- The threat during a cloud deployment, generally, depends on the kinds of cloud and its models.

4.2 Security Challenges in Cloud Computing

As the cloud is becoming so huge and growing rapidly, many challenges are involved in various aspects of handling information. Some vital challenges in cloud computing are laid down in Fig. 5.

Fig. 4 SEQ Figure * ARABIC 4: IaaS

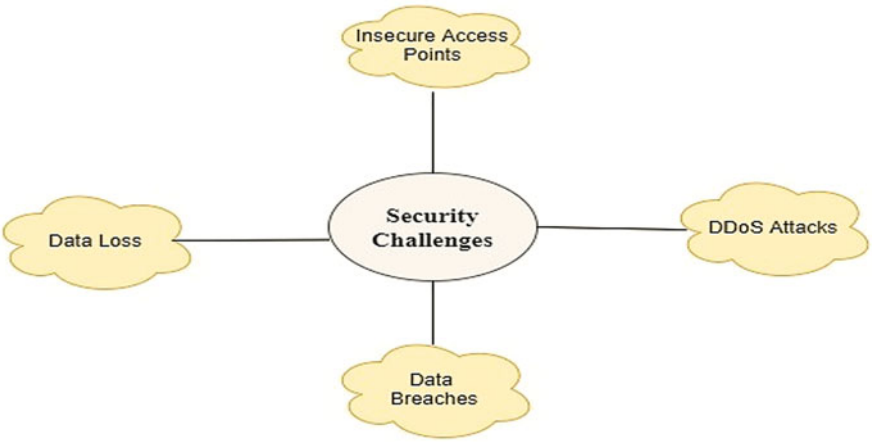

Fig. 5 SEQ Figure * ARABIC 5: vital security challenges

4.3 Degree of Security at Different Levels of Service

As we know, cloud services are divided into three phases, that is, IaaS, PaaS, and SaaS, but at each phase the responsibility of security to a degree of control is different. The control of security responsibility at different levels is discussed below:

Fig. 6 SEQ Figure * ARABIC 6: degree of security responsibility at different level

(a) IaaS: Obligation is on the supplier for essential security, although the cloud client is in-control for the entire thing they expand on the foundation. Differentiating PaaS, this spot is considerably more obligation of the customer.
(b) PaaS: Cloud Service Provider is responsible for the stage security, while the client/purchaser at risk for each and all that they execute on the stage, containing how they design any gave well-being measures.
(c) SaaS: Cloud Service Provider is subject to around all security, since the cloud client can just utilize and control their utilization of the application, and can't alter how the application functions (Fig. 6).

These parts are further troublesome when using cloud operators or different arbiters and accomplices.

The significant security thought knows precisely who is responsible for what in some random cloud venture. It is minor if a specific cloud supplier offers a chosen security controller, as long as you probably are aware of precisely what they do offer and how it functions. You can fill the holes with your controls, or pick a unique supplier if you can't close the controls' hole. Your capacity to do this is high for IaaS, and less so for SaaS.

4.4 Cloud Security Models

Cloud security models are apparatuses to help manage security choices. There are the following sorts of model:

- Conceptual models or plans incorporate originations and portrayals used to clarify cloud security observations and standards, for example, the CSA sensible model in this record.
- Controls models or structures classify and detail explicit cloud security controls or classifications of controls, for example, the CSA CCM.
- Reference structures are models for actualizing cloud security, normally broad (e.g., an IaaS security reference design). They can be very nonconcrete, verging on theoretical, or very comprehensive, directly down to correct controls and capacities
- Design structures are reusable outcomes to specific issues. In security, a model is IaaS log the board.

The lines amid these models frequently obscure and cover, remembering for the objectives of the designer of the model. In any event, gathering these out and out under the heading "model" is perhaps inaccurate, however, since we see the terms utilized so conversely across unique sources, it makes a rationale to bunch them.

4.5 How Security Changes with Cloud Networking

The absence of direct association of the first physical organization varieties normal organization rehearses for the cloud buyer and provider. The pinnacle ordinarily utilized organization security designs rely upon control of the physical correspondence ways and incorporation of security machines. This isn't feasible for cloud shoppers, since they just capacity at a virtual level.

Customary Network Intrusion Detection Systems, wherein correspondences between are reflected and examined by the virtual [9] or physical intrusion detection systems won't be held in cloud conditions; client security apparatuses got the chance to accept an in-line virtual machine or a product specialist introduced in delineations. This creates either a chokepoint or floods processor overhead, so ensure you really need that degree of seeing before satisfying. Some cloud providers may offer some degree of implicit organization observing (and you have more choices with private cloud stages), yet this isn't generally similar to snuffling a physical organization.

4.6 CSA Stack Model

CSA (Cloud Security Alliance) heap model characterizes the limits of each assistance model and shows how much disparate the utilitarian units identify with one another. It is subject to developing the fringe between the administration provider and the customer.

Key Points

- IaaS is the most fundamental level among all administrations.
- Each of the administrations gets the skills and security troubles of the model underneath.
- The structure, stage for improvement, and programming working conditions are given by IaaS, PaaS, and SaaS separately.
- The security instrument underneath the security fringe must be developed into the framework that is needed to be held by the customer.

5 Security Guidelines, Measures, and Technique

To secure the cloud, plenty of measures, rules, and techniques are mentioned by the researcher. Not only the investigators, the government has even laid down some measures to be taken into consideration by cloud service supplier to confirm security to their clients and the general public of using the cloud.

In January 2015, some guidelines for "Protection of Critical Information Infrastructure" have been published by National Critical Information Infrastructure Protection Centre (NCIIPC) under the Government of India.

NCIIPC works as a nodal agency with the aim to take all necessary measures to facilitate the protection of critical information infrastructure, from unauthorized access, use, disclosure, modification, disruption, incapacitation or destruction, through coherent coordination, synergy, and raising information security awareness among all stakeholders with a vision to facilitate safe, secure, and resilient Information Infrastructure for Critical Sectors in the country (Fig. 7).

NCIIPC introduces five families of controls:

- Planning controls: It works at the design level where all the preparation is done.
- Business continuity planning (BCP) control: It ensures minimum downtime and restoration process.

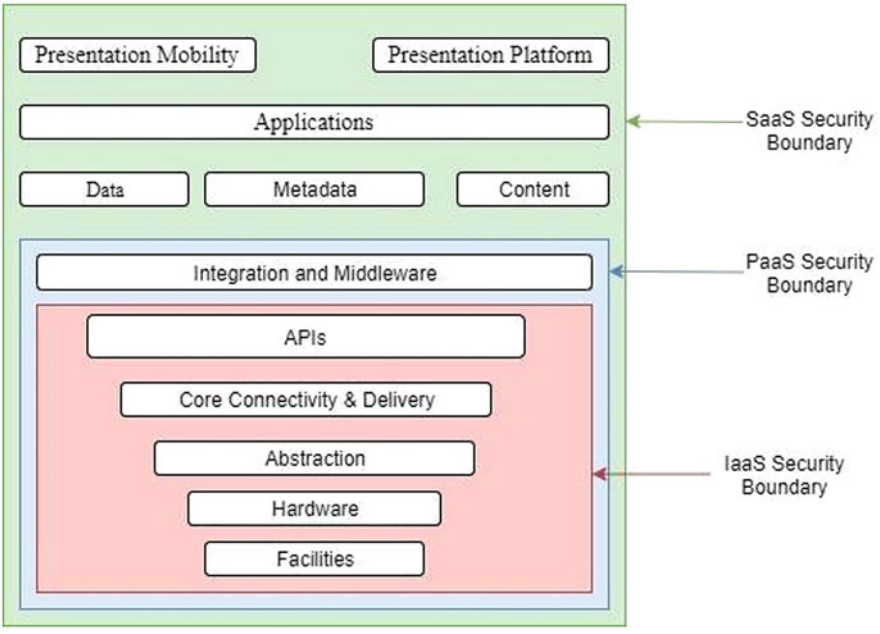

Fig. 7 SEQ Figure * ARABIC 7: CSA stack model

- Implementation controls: It translates the design into a mechanism for protecting CII.
- Reporting and accountability control: Ensuring adequate accountability by senior management and reporting to the concerned government agencies where required, enforced through compliance controls.

Certain cloud security standards that have been developed by several organizations are:

1. NIST: National Institute of Standards and Technology, discusses the threats, technology risks, and safeguards surrounding public cloud environments and their suitable defense mechanisms.
2. OGF: Open Grid Forum, concerned with technical and operational security issues in the grid and cloud environments, including authentication, authorization, privacy, confidentiality, auditing, firewalls, trust establishment, scalability and management aspects of these issues, and so on.
3. DMTF: Distributed Management Task Force, partnership with CSA to promote standards for cloud security as part of DMTF Open Cloud Standard Incubator.
4. CSA: Cloud Security Alliance, covers key issues and provides advice for both cloud computing customers and providers within various strategic domains.

GI Cloud is an initiative of the Government of India to provide guidelines and measures to be taken by cloud service providers as well as cloud service users. Under GI Cloud, MeghRaj Bill [11] has been introduced in association with USIBC, NASSCOM, and BSA. The Government of India has implemented several ICT initiatives under the National e-Governance Plan (NeGP), including the creation of ICT infrastructure both at the center and state levels. The infrastructure thus created will provide the ground for the adoption of cloud computing for the government with the objective of making optimum use of existing infrastructure, reuse of applications, efficient service delivery to the citizens, and increasing the number of e-transactions in the country, thus helping to achieve the concluding goal of NeGP. Steps were taken by GI Cloud under MeghRaj.

5.1 Enabling Activities for GI Cloud

An Empowered Committee will be formed under the chairmanship of Secretary, DeitY with representations from central/state line ministries and other government entities. It will provide strategic direction and guidance to DeitY on key matters pertaining to the functioning of GI Cloud [10–15].

Policies, standards, guidelines, and frameworks for GI Cloud will be defined at the national level and will be implemented across the country. The standards and guidelines will be developed in consultation with the industry and based on international best practices. It is proposed to create a ‘GI Cloud Expert Group’ with experts from the industry to deliberate on these standards/guidelines.

As security and privacy play an important role in cloud adoption [16], AMO will also focus on security guidelines defining the various challenges, risks, and the approach for mitigating the same. Capacity and capability-building exercises will be carried out across the country both at the national and state level for the adoption of cloud computing by the government.

5.2 Cloud Security

Security considerations remain one of the main factors inhibiting the adoption of cloud technology. It is imperative to understand and address the risks and challenges associated with the adoption of cloud. Usage of the cloud should not lead to increased risks of compromise of confidential information and intellectual property (IP), and inappropriate/unauthorized access to personal information. A robust security framework needs to be put in place to address such concerns.

DeitY shall prescribe the standards around interoperability, integration, data security, portability, operational aspects, contract management, and so on, for the cloud. Architecture Management Office (AMO), an important component of the GI Cloud institutional setup, will be responsible for defining specifications on security addressing the various challenges, risks and for prescribing the approach for mitigating the risks.

6 Conclusions and Suggestions

It is of the view that cloud has played an important role in various fields and has made work easier, but with that lots of challenges and security issues are involved. There are three major types of cloud computing services, that is, PaaS, IaaS, and SaaS. This is a three-level network that is jointly important in cloud computing. Each one of them is responsible for the control of security at different levels. The major issues involved in cloud security are Data Loss, DDoS Attacks, Data Breaches, and Insecure Access Points. The Indian Government and various national and international organizations have taken step to provide guidelines, techniques, and so on, to cloud service providers to secure cloud security. In 2015, some guidelines for "Protection of Critical Information Infrastructure" have been published by National Critical Information Infrastructure Protection Centre (NCIIPC) under the Government of India. Ministry of Electronics and Information Technology (MeitY) has announced a policy to provide strategic direction for the adoption of cloud services by the Government is "MeghRaj Policy."

But having these policies and guidelines and measures and standards have no effect unless the general public as well as the cloud service providers are made aware of it properly. And also, this policy, that is, MeghRaj Policy should have separate legislation or be combined with the existing legislations like Information Technology Act, 2000 (Amendment 2008).

References

1. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2017). Privacy and security of cloud-based internet of things (IoT). 2017 3rd international conference on computational intelligence and networks (CINE) (pp. 40–45).
2. Sulistio, A., & Reich, C. (2013, September). Towards a self-protecting cloud. In *OTM confederated international conferences on the move to meaningful internet systems* (pp. 395–402). Berlin: Springer.
3. Section 43A, The Information Technology Act, 2000.
4. Oppermann, A. (2020). Secure cloud computing in legal metrology.
5. Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Energy-aware autonomic resource scheduling framework for cloud. *International Journal of Mathematical, Engineering and Management Sciences, 4*(1), 41–55. https://doi.org/10.33889/IJMEMS.2019.4.1-004.
6. Gill, S. S., & Buyya, R. (2018). SECURE: Self-protection approach in cloud resource management. *IEEE Cloud Computing, 5*(1), 60–72.
7. Khalil, I. M., Khreishah, A., & Azeem, M. (2014). Cloud computing security: A survey. *Computers, 3*(1), 1–35.
8. Venkatadri, M., Agarwal, A., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science, 152*, 204–211. https://doi.org/10.1016/j.procs.2019.05.044.
9. Srivastava, N. (2018). MeghRaj a cloud environment for e-governance in India. *International Journal of Computer Sciences and Engineering, 6*, 759–763.
10. Sharma, M., Bansal, H., & Sharma, A. K. (2012). Cloud computing: Different approach & security challenge. *International Journal of Soft Computing and Engineering (IJSCE), 2*(1), 421–424.
11. MeghRaj Government of India, Cloud Initiative, www.meity.gov.in
12. Dewangan, M. B. K., & Shende, M. P. (2012). Survey on user behavior trust evaluation in cloud computing. *International Journal of Science, Engineering and Technology Research, 1*(5), 113.
13. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*, 1144–1148.
14. Pasricha, A., Agarwal, A., & Venkatadri, M., (2018, December). Autonomic cloud resource management. In 2018 fifth international conference on parallel, distributed and grid computing (PDGC) (pp. 138-143). IEEE. https://doi.org/10.1109/PDGC.2018.8745977.
15. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*, 583–585.
16. Choudhury, T., Agarwal, A., Pasricha, A., & Chandra Satapathy, S. (2020). Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges. *Software: Practice and Experience*. https://doi.org/10.1002/spe.2810.
17. Sharma, A., Choudhury, T., & Kumar, P. (2018). Health monitoring & management using IoT devices in a cloud based framework. 2018 international conference on advances in computing and communication engineering (ICACCE) (pp. 219–224).
18. Mittal, A., Khan, F. S., Kumar, P., & Choudhury, T. (2018). Cloud based intelligent attendance system through video streaming. Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358587.
19. Kumra, S., Choudhury, T., Nhu, N. G., & Nalwa, T. (2018). Challenges faced by cloud computing. Proceedings of the 2017 3rd international conference on applied and theoretical computing and communication technology, ICATccT 2017. https://doi.org/10.1109/ICATCCT.2017.8389105.

20. Dinote, A., Sharma, D. P., Gure, A. T., Singh, B. K., & Choudhury, T. (2020). Medication processes automation using unified green computing and communication model. *Journal of Green Engineering, 10*(9), 5763–5778.
21. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.

Elastic Security for Autonomic Computing Using Intelligent Algorithm

Amar Buchade, Rajesh Ingle, and Vidyasagar Potdar

1 Introduction

As per NIST definition [1], cloud computing enables convenient access to resources through sharing approach in elastic manner. Here elasticity means increasing and lowering the demand. The low upfront cost enables the users to use cloud computing resources, such as applications, storage, services, etc. The important characteristics of cloud environments are as follows:

- On-demand access: It indicates the provision of resources without direct interference.
- Network access: Resources are available to a wider range of devices (broad).
- Resource pooling: Resources are grouped to access it through multitenant model.
- Rapid elasticity: It indicates provision of resources on demand.
- Measured service: It indicates monitoring of resource use through pay-per-use model.

There are three basic deployment models such as private, public, hybrid, and community for provision of resources. The following service models cloud computing provided to the users [1].

- IaaS: It makes available computational resources, storage, and network. For example, Amazon web service, IBM Cloud, and google cloud platform.

A. Buchade · R. Ingle
Pune Institute of Computer Technology, Pune, India
e-mail: arbuchade@pict.edu; rbingle@pict.edu

V. Potdar (✉)
Blockchain Research and Development Laboratory, Curtin University, Perth, Australia
e-mail: Vidyasagar.Potdar@cbs.curtin.edu.au

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_13

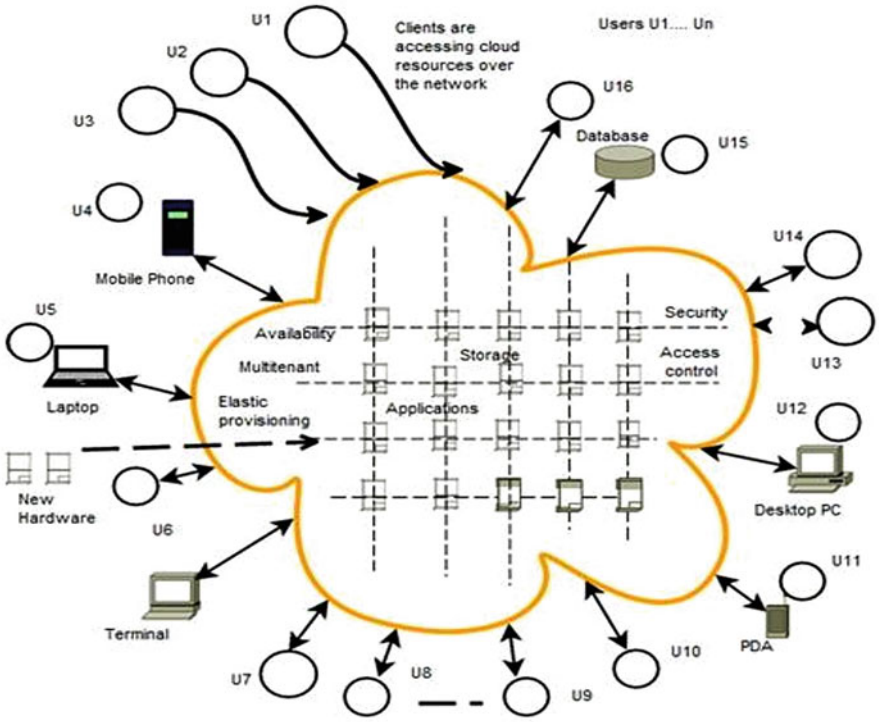

Fig. 1 Cloud computing scenario

- PaaS: It provides a tool to the user for creating applications and deployment at cloud. For example, Google app engine and Openshift.
- SaaS: It provides the applications owned by cloud provider. For example, Google Apps and Dropbox

Figure 1 shows cloud computing scenario. The users can send the information to cloud computing environment. Availability, multitenancy, elastic provisioning, and low cost are the features of cloud computing. These are shown in the figure. New hardware is added to the pool of computing resources on demand. The applications can be deployed as software or as service.

Cloud computing plays an important role as the data storage and computing resource for digital transformation in Industry 4.0. Data generated through industrial activities will be stored in Cloud Computing for digital transformation. In Industrial 4.0, the interconnectivity between machine, data, and people are important. Innovative applications on an integrated cloud platform can be deployed. Industry 4.0 will be driven by the integration of new resources and technologies. With the advent of IoT and Industry 4.0, a large amount of data is generated with speed and at high volumes. This creates a need for suitable infrastructure to manage such data more efficiently in Industry 4.0 revolution. Cloud computing offers an

environment for the users to store and process vast amount of data.Volkswagen [2] proposed automotive cloud that includes connectivity to smart home, PDA, predictive maintenance service, and media streaming. The communication platform and cloud-based storage play important role to overcome the challenges. Industrial 4.0 companies used the support of cloud for robotics.

2 Motivation for Elastic Security in Cloud Computing

Cloud computing provides resources that include the data on demand to the user. Data stored in cloud computing environment are maintained at cloud service providers (CSP) side. It is a multitenant [3] environment. Infrastructure is shared among the users. There is a risk to exploit the usage of data at cloud and data in transit due to attacks [4–7]. The information stored in cloud is required to be protected against the internal attackers, CSP. The user is not sure if the data exist in cloud and when he or she leaves that cloud environment. The main approach used to secure important information is mainly cryptography. There is important role of elastic security for cloud computing. Elastic security means dynamic change in the security level for protecting sensitive information based on the behavior in system. If there is high risk, the security level can be increased, and otherwise, security factors can be decreased. The work focuses on the design of algorithms and techniques for key management of elastic security in cloud computing. The key management can be applied for securing online transactions, multiparty computation, data sanitization, consensus, etc.

The security aspect of each service model has been identified as mentioned below.

Security at IaaS:

- An authentication of application programming interface functions invoked to the interface that manages virtual machine in hypervisor management environment.
- An authentication of virtual machine (VM) image template. Protect the interaction with instances of apps running on VMs.
- Secure the data stored in cloud.

Security at PaaS: PaaS model provides the platform and application development tools to users for the development or deployment of applications.

- Protect communication with applications.
- Protect interaction with development tool instances.
- Secure app data.

Security at SaaS: SaaS model provides access to the apps or services.

- Secure communication with an app/service.
- The ability to store data in a cryptography form.

By above mentioned security aspects for each model, elastic security is needed to protect the cloud computing resources. Other important aspects indicate the importance of the need for elastic security.

- As per 2020 cloud computing trends, state of cloud survey [8], the usage of cloud computing is increasing. Security is still a major concern for cloud computing.
- Risks of key leakages due to attacks toward cloud computing environment [2–5, 9].

3 Challenges of Elastic Security in Cloud Computing

The challenges of elastic security in cloud computing are due to the distinctness of managing resources. For example, the owner of the data is the user, but it is hosted at cloud service provider side, i.e., the under control of underlying infrastructure. The important challenges are as follows:

- The data are maintained at CSP. Thus, there is a risk to the stored data which can be manipulated by an internal employee at CSP. It does not ensure to the users' data such as theft of data.
- The data may remain in a cache of the memory space during the usage of services. This may cause the access of sensitive data to be accessible by the cloud service provider.
- The protection of keys when you maintain at cloud from the attackers.
- Cloud computing has on-demand resource access characteristics. Thus, while maintaining elastic security management, it will not cause delay to access the resources.
- The user stores important information in the storage of cloud computing environment. At present, even after the data are accessed by the user, data are still accessible. This leads to the misuse of data by attackers.

Figure 2 mentions the view of the CC and CSP for IaaS, PaaS, and SaaS. The figure differentiates the control of CC and CSP over resources in service models.

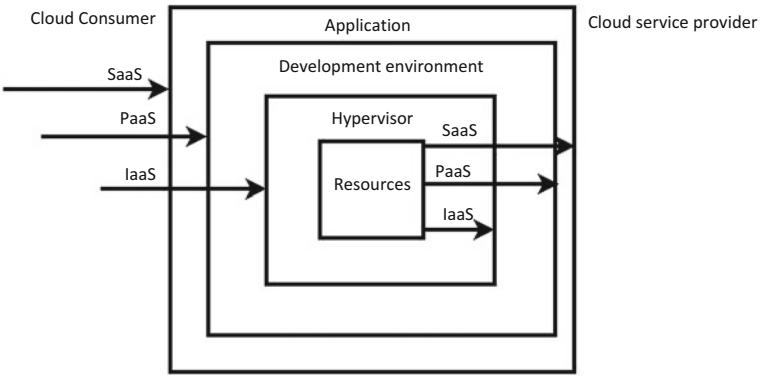

Fig. 2 CSP and cloud consumer (CC) view of cloud computing

For example, in SaaS, the cloud service provider has full control over on hypervisor, application, development environment, and resources are with CSP and control on the application layer at CC. In PaaS, the control over resources, hypervisor, and development environment is with CSP and application development environment with CC. In IaaS, the complete control over resources is with CSP and preconfigured resources with CC.

In this era, the use of social networks like Twitter, WhatsApp, Facebook, Linkedin, etc. is increasing and also threats [10]. Thus, there is a need to secure data, and only authorized users should be able to access it.

Security is a major concern due to the following reasons.

1. Possibility of data breaches: Due to attacks toward cloud environment by the attacker and internal employee for cloud service provider, it is possible of data breaches.
2. Hijacking of the accounts: Due to vulnerabilities present at cloud side, there is possibility of account hijacking. This is also because data are under the control.
3. Injection of malware: The intruder inserts malware to cause adverse effect on the data and virtual machine.
4. Abuse of cloud services: It leads by the attack to cause denial of service due to continued requests to cloud services.
5. Insecure APIs: Attackers create the API. That will directly or indirectly access the resources of cloud without using authentication. It causes potential exploitation of the system.
6. Denial of Service Attacks: The attackers, due to continuous requests called as network flood to cloud servers, are unable to provide service to actual clients.
7. Insufficient Due Diligence: Lack of maintaining security and compliance causes risk.

Due to multitenancy nature of cloud, resources are accessed in the shared manner. Due to this feature, it can lead to many attacks including cross-VM side-channel attack. The paper [11] describes attacks due to which information can be inferred.

Yinqian Zhang [11] describes cross VM side-channel attack to know information that is present in VM even under logical isolation in public cloud infrastructure. The intruder can keep its VMs alongside the victim VM and learn important information. Extraction of important information including keys extraction by cross VM attacks is possible.

Onur [12, 13] also analyzed the usage of cache to extract the keys present in the VM. If keys are stored in cloud computing, these keys can be grabbed by the attacker. Considering various scenarios following the security aspects need to be considered.

- Security at client side.
- Security of data that is in transit.
- Security at cloud server.
- Security at machine migration.

Thus, it is observed that key management in Cloud computing is important. Security of data depends on the secrecy of key. Attackers attack to get the resources from cloud computing. There is need for elastic security for managing secure access of resources. Key management through threshold cryptography can also help in managing elastic security.

In Sect. 6, we described the intelligent algorithm such as secret sharing algorithm and technique such as threshold changeability technique to achieve the elasticity. Depending on the state of cloud, the threshold is changed as per need or requirements.

4 Security and Privacy of Data Approaches

4.1 Access Control Mechanism

It is important for the end user to control the access of resources from the other users due to multitenancy nature of cloud computing. Thus, the access control policy to access resources is required. The paper discusses access control mechanisms for cloud computing [14].

4.2 Identity and Access Management (IAM)

IAM consists of authentication, authorization, and auditing. Credential management, authentication (one-time, multifactor authentication, and anonymous authentication), and authorization are essential to secure the resources in the cloud. The paper [15, 16] discusses various authentication and authorization mechanisms for identity management and secure access of cloud resources in multitenant environment.

The authorization and authentication can be inspected through monitoring to detect the security breach. One of the approaches is monitoring through log files.

4.3 Homomorphic Encryption

In cloud computing for maintaining security, the data are stored in encrypted form. But if certain operation has to be performed, the data need to be decrypted in original form and perform the operation. Homomorphic encryption involves the operation on encrypted form [17].

Table 1 Employee records

Employee name	Staying place	Expertise	Year of joining
Amar	Dadar(W)	Data management	2010
Sharad	Sion (E)	Data technology	2012
Suhas	Ghatkopar (E)	Electronics commerce	2019
Siddheshwar	Mahim (W)	Electronics and Tele.	2018

Table 2 Anonymized employee records

Employee name	Staying place	Expertise	Year of joining
Amar	Dadar	Data management	2010
Sharad	Sion	Data technology	2012
Suhas	Ghatkopar	Electronics commerce	2019
Siddheshwar	Mahim	Electronics	2018

4.4 Anonymity

Using anonymity, it is possible to publish sensitive information without disclosing identity and personal information for research. The release of patient's records at the hospital happens for research purpose for studying the features of the diseases without disclosing the names of the patient.

Institute's record may contain information of employees such as name, address, branch, and admission year. For example, Table 1 shows employee records.

Company can anonymize [18] the employee record on attributes, such as address and branch, expertise called quasi-identifiers before such information releases to CSP.

E.g., Table 2 shows anonymized information of Table 1 employee records based on attributes such as staying place and expertise.

4.5 Securing Information Through Hardware Approach

Trusted platform module (TPM) is a chip that stores important keys. It allows only intended users to access the information [19].

4.6 Threat Modeling

Threat modeling is an approach for designing secure system against attacks, such as identity, hack, data theft, denial of service, and privilege hack. Threat modeling is presented in [20].

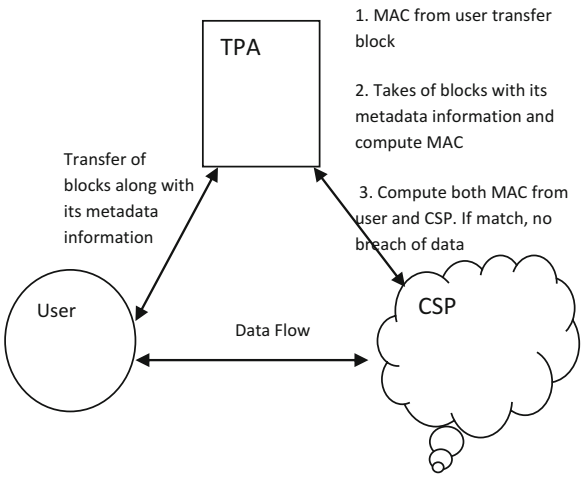

Fig. 3 User, TPA, and CSP for privacy preservation

4.7 Third-Party Auditor (TPA) for Privacy Preservation

In cloud computing, data are outsourced. Third-party auditor [TPA] that manages outsourced data for maintaining privacy of the user is mentioned at [21]. In Fig. 3, data owner gives certain number of blocks to TPA, and TPA receives certain number of blocks from CSP and computes the message authentication code. If both message authentication codes are identical, information is maintained without any modification.

4.8 Auditing of Service-Level Agreement (SLA)

The SLA is important to provide service to the user for outsourced information at cloud. The auditing of SLA is necessary. In [22], the mechanisms of SLA management are described.

4.9 Security and Privacy Issues in Virtualization

The security issues in cloud computing arise due to multitenancy nature of cloud computing. The single instance, e.g., resource, may be accessed by multiple tenants. There is a possibility that the guest operating system process can invoke code on the host OS. The paper [23] describes security aspects.

4.10 Securing Data in Cloud Through Approach of Merging

The data generated due to IoT devices is in a high-speed manner with high volume. For securing such big data, there is a need to combine various approaches for securing important data, such as sensitivity rating, cryptographic approach, session key negotiation, message authentication code, authenticated key exchange protocol, etc [24].

5 Role of Key Management for Providing Cloud Security

Key management is important for securing access of resources in cloud computing. Cloud computing provides three types of services such as IaaS, PaaS, and SaaS. On-demand self-service, broad network access, resource pooling, rapid elasticity, multitenancy, and measured service are essential characteristics. It is the remote access model for accessing the resources. Key Management system (KMS) should be designed in such a way that it should consider the characteristics of cloud computing as well as performance metrics such as storage, communication, computation cost, and protection of the key.

The following techniques utilize key management.

- User-managed public key encryption (PKE).
- Proxy re-encryption.
- Certificate less encryption.
- Convergent.
- Group key management.
- Attribute based.
- Threshold cryptography.

5.1 User-Managed Key Management

In this method, data in encrypted format are stored in the cloud computing environment. Key may be stored in the user's mobile device. The control of data and key is toward the user. Storage and processing cost may be required to manage the key as well as there is a threat of losing the mobile device. This causes data leakage by attackers. [25] proposes user-managed key management approach for securing cloud data. RSA cryptography approach is used. Private key is stored on user's mobile device. As RSA [26] has larger key size, it may not be suitable for resource-constrained devices such as mobile phones due to storage and processing capability.

5.2 Public Key Encryption (PKE)

Two types of keys can be maintained, such as public key and private key. Before sending the data to cloud storage, data may be encrypted by private key at the user and decryption by recipient public key. Thus, it indicates the authenticity of the sender. The data may be encrypted with recipient public key by the sender and stored at cloud. Public keys are generated and maintained at Public Key Infrastructure (PKI). PKI contains entities such as registration, Certificate Authority (CA), directory service, and revocation service. Registration authority gets the details of the user who wants the public key from PKI. Certificate Authority issues the digital certificate. This digital certificate may be encrypted by the CA's private key. Thus, the client confirms that this digital certificate is from particular certificate authority. Digital certificate confirms the ownership details of the given public key. PKI may be required for RSA and ECC cryptography algorithms. Directory service includes the directory of public keys. But the performance of PKI is unrealistic [27] in terms of authentication as well as scalability in case of the number of users increases. There is also a need to protect CA key due to key leakages.

5.3 Proxy Re-encryption

It is a delegation model. The third party re-encrypts the data on the behalf of other user without revealing private key. Proxy re-encryption may lead with collusion problem and involves complex key management operations [28, 29] details about proxy re-encryption schemes in cloud computing environment.

5.4 Certificate-Less Encryption

Traditional PKI incurs the high cost of key management. Digital Certificate Management in PKI is complex. It has key escrow problem and key revocation problem. Certificate-less encryption is proposed in [30, 31]. Certification of public keys is not required. Public key is generated by the user. The encryption is done by public key and user's ID. Private key is formed by partial value provided by key generation center at cloud and secret user value. Secure channel may be required to provide the partial private key. The decryption is done by the private key. Key generation center is maintained at cloud which provides the partial private key to the user. There is no mechanism mentioned in [32] about how the keys are stored in cloud storage.

5.5 Convergent Key Management

Convergent key management [32] can be used to avoid the duplication of data in cloud computing. Hash key is generated from the content. If the same hash key is generated from other contents, then the content is said to be duplicated. It can be used to reduce cloud storage costs.

5.6 Group Key Management

Group Key Management is required when members in group accesses the resources. Resource may be considered as data, CPU, VM, etc. To form the group key, Tree-based Group Diffie-Hellman (TGDH) protocol is used, [33–36] propose share-based key management scheme. [37] proposes the scheme of tree key graph design, but it has computation overhead for connection network generation. [38, 39] propose group-based sharing of files among the different users in Cloud.

5.7 Attribute-Based Key Management

Mohamed Nabeel et al. and Jinbo Xiong et al. [40, 41] proposes the approaches based on attribute-based encryption. The users can access the data if identity attributes satisfy content provider's policy. Attribute-based encryption (ABE) can be used to achieve data security and access control. ABE are of two types: Cipher text policy ABE (CP-ABE) [42, 43] and Key policy ABE (KP- ABE) [41, 44, 45]. In CP-ABE, file is encrypted under the access structure policy, and cipher text is only decrypted when attributes are matched with the access structure associated with ciphertext. In KP-ABE, the file is encrypted under the set of attributes, and decryption is possible when the set of attributes satisfy the access structure. Approaches based on ABE are not scalable and effective in case of member's revocations.

5.8 Threshold Cryptography-Based Key Management

In the threshold cryptography [46], key is split into the multiple key shares by the third party. In the existing systems, threshold cryptography is applied during the group communication among the members. The third party distributes each key share to the multiple members in the group. Each member collects threshold number of key shares from the other members. Key is generated from the threshold number of key shares. The existing approach tends to increase the communication cost as

the members increase. The third party needs to be online at each instant of time to distribute the key shares to the new member. There are various threshold schemes, such as Blackley [47], Rabin IDA [48], Shamir [49], Asmuth-Bloom [46, 50], and Mignotte [51]. The applications of threshold cryptography are mentioned in [52]. As per NIST, threshold cryptography mentioned as future key management approach [53–55].

6 Secret Sharing Algorithms

Secret sharing algorithms can be applied to autonomic computing environments such as cloud computing. It acts as an intelligent algorithm for protecting key/data for cloud computing.

A secret sharing algorithm [49] can be used to keep the secret secure. The secret sharing algorithm can be used to split the key or data into multiple splits as per user requirements, behavior of the system, and security policy. Owner of the data/key splits into multiple shares as per predefined threshold. These shares can be distributed and stored into multiple locations or virtual machines in the cloud environment. Depending on the security concern at the moment of time, the shares can be increased. Only authorized user will get access to threshold and combine all the shares to get the original data or key. The secret sharing algorithm can be used for managing the key and achieving elastic security to the cloud. Elastic security means changing security level as per need. The owner of the data/key, depending on the system behavior change, the number of shares as well as partial keys thus make the attacker to get the complete key or data.

To increase the security level, the key can be protected by splitting in multiple level depending on the user security requirement. At the first level, key is split into n number of key shares. At the second level, each key share can be split into n partial key shares, and so on. Thus, for each level, split t into n partial key shares by using threshold cryptography (t,n) scheme. Further, these partial key shares are stored at different virtual machines in cloud computing which makes the attacker difficult to collect the key shares and form the key.

Figure 4 shows how the key is split by using threshold cryptography (t,n) approach into key shares. For level 1, key is split into three key shares. For level 2, each key share is further split into three partial key shares.

7 Results and Analysis

The secret sharing algorithm can be used to split the key or data into multiple splits as per user requirements, behavior of the system, and security policy. These shares can be distributed and stored into multiple locations or virtual machines in the cloud environment. Threshold value can be changed. In the following analysis, threshold

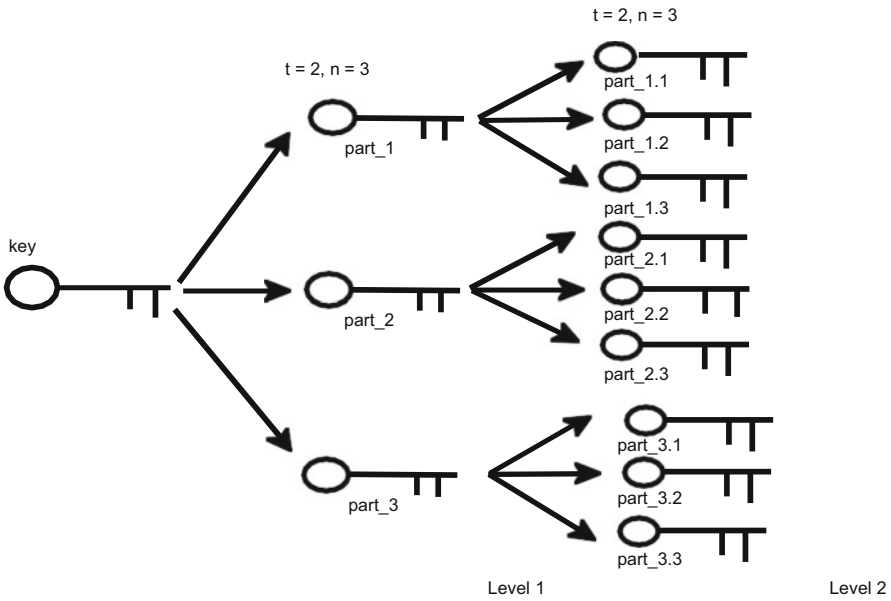

Fig. 4 Key split up approach

values for key shares are changed from 1 to 10. The computational cost required to combine shares is analyzed.

Shamir's secret sharing algorithm is applied. In Fig. 5, the total number of shares is taken 10. The k out of n threshold scheme is followed. As we increase the number of key shares, i.e., k, combining key shares time is also increases. k is varying from 1 to 10.

The key is split into multiple shares. In Fig. 6, the key is split up into 10, 20, . . . , 100 key shares. The time to key split up is also more as we increase the number of key shares. Key split-up should be done in a dynamic environment such as cloud computing. This is due to security threats to such environment.

8 Security Analysis

The key shares are placed across different VMs/locations in cloud computing. Thus, it is not possible by the attackers to get/form the key from key shares from different virtual machines. Through threshold changeability approach and multilevel key sharing approach, i.e., elastic security, it is not possible to reconstruct the complete key.

Even if the attackers get less threshold number of key shares, it is not possible to construct the key. This approach resists side-channel attack and fault.

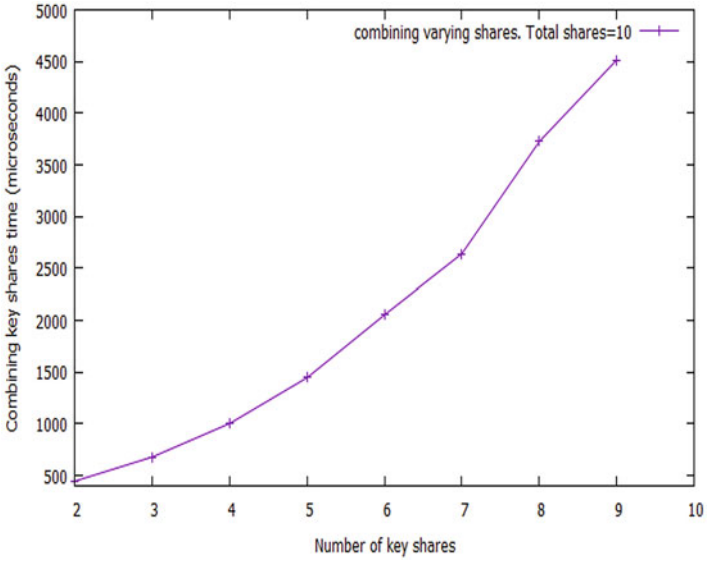

Fig. 5 Threshold key share management

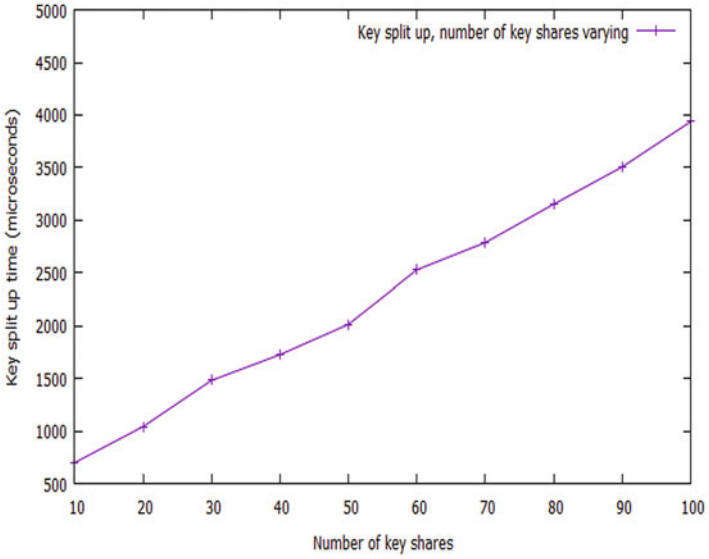

Fig. 6 Key split-up time

9 Machine Learning Algorithm to Identify Threat Patterns that Enable Security of Cloud

A cloud computing system can be exposed to several threats, including threats to the integrity, confidentiality, and availability of its resources, data, and the virtualized infrastructure. Lack of full control over the infrastructure is a major concern for the cloud services' consumers. An IDS is a software that automates the intrusion detection process and detects possible intrusions. IDS can monitor, detect, analyze, and respond to unauthorized activities. The monitored environment can be network-based, host-based, or application-based. Machine learning based: this technique has the ability of learning and improving its performance over time.

As machine learning is giving solutions to most of the problems and issues spread across various disciplines, it is natural to apply to the security of the cloud. In contrast with traditional mode, the virtual machines are dynamically added and removed. Moreover, the security requirements of each virtual machine tend to be varied. Identifying threat pattern is crucial to enable security of the cloud. Most of the intruders come from insiders. Cloud computing is shared infrastructure and virtualization technology that leads to vulnerability. Any weakness in hypervisor allows creating virtual machines and running multiple operating systems which can cause inappropriate access and control to the platform. Use of machine learning algorithm to identify threat pattern such as intrusions and attack behaviors are an important step toward solving security problems. Data-driven intrusion design with machine learning technique such as support vector machine can be used to identify features for intrusion detection.

10 Summary

The focus of this chapter was on the use of cloud computing in Industry 4.0 era. In this chapter, we covered key management roles for providing security of cloud. The chapter covers various security techniques that can be applied for securing resources in the cloud. The chapter explains how the secret sharing algorithm can be applied for managing the key and achieving elastic security to the cloud. Finally, the chapter concludes with the use of a machine-learning algorithm to identify threat patterns, which enable the security of cloud.

References

1. Jansen, W., & Grance, T. (2011). Guidelines on security and privacy in public cloud computing. *NIST Special Publication, 800*(144) 144, 10–11.
2. Hüttel, H. Volkswagen automotive cloud. https://www.volkswagenag.com/en/news/stories/2019/03/automotive-cloud-volkswagen-and-microsoft-develop-mobility-ecosy.html

3. Gözde, K. et al. Multi-tenant architectures in the cloud: A systematic mapping study. 2017 IEEE international artificial intelligence and data processing symposium.
4. Ristenpart, T., Tromer, E., Shacham, H, & Savage, S. (2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on computer and communications security (pp. 199–212). ACM.
5. Zhang, Y., Juels, A., Reiter, M. K., & Ristenpart, T. (2012). Cross-vm side channels and their use to extract private keys. In Proceedings of the 2012 ACM conference on computer and communications security (pp. 305–316). ACM.
6. Harrison, K., & Xu, S. (2007). Protecting cryptographic keys from memory disclosure attacks. In 37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07) (pp. 137–143). IEEE.
7. Qiao, R., & Seaborn, M. (2016). A new approach for rowhammer attacks. In 2016 IEEE international symposium on hardware oriented security and trust (HOST) (pp. 161–166). IEEE.
8. Flexera. (2020). Flexera: 2020 state of the cloud report. Hybrid cloud adoption ramps as cloud users and cloud providers mature. In Rightscale state of cloud report (pp. 1–68).
9. Pattuk, E., Kantarcioglu, M., Lin, Z., & Ulusoy, H. (2014). Preventing cryptographic key leakage in cloud virtual machines. 23rd USENIX security symposium (USENIX security 14) (pp. 703–718).
10. Venkatadri, G., et al. (2018). Privacy risks with Facebook's PII-based targeting: Auditing a data broker's advertising interface. 2018 IEEE symposium on security and privacy (SP). IEEE.
11. Zhang, Y., et al. (2012). Cross-VM side channels and their use to extract private keys. Proceedings of the 2012 ACM conference on computer and communications security.
12. Acıiçmez, O., Brumley, B. B., & Grabher, P. (2010). New results on instruction cache attacks. In *International workshop on cryptographic hardware and embedded systems* (pp. 110–124). Berlin: Springer.
13. Saxena, S., & Sanyal, G. (2018). Cache based side channel attack: A survey. 2018 International conference on advances in computing, communication control and networking (ICACCCN). IEEE.
14. Karataş, G., & Akbulut, A. (2018). Survey on access control mechanisms in cloud computing. *Journal of Cyber Security and Mobility, 7*(3), 1–36.
15. Indu, I., & Rubesh Anand, P. M. (2015). Identity and access management for cloud web services. 2015 IEEE recent advances in intelligent computational systems (RAICS). IEEE.
16. Indu, I., Rubesh Anand, P. M., & Bhaskar, V. (2018). Identity and access management in cloud environment: Mechanisms and challenges. *Engineering Science and Technology, an International Journal, 21*(4), 574–588.
17. Gentry, Craig, and Dan Boneh.: A fully homomorphic encryption scheme. 20 9. Stanford: Stanford University, 2009.
18. George, R. S., & Sabitha, S. (2013). Data anonymization and integrity checking in cloud computing. 2013 fourth international conference on computing, communications and networking technologies (ICCCNT). IEEE.
19. Hosseinzadeh, S., et al. (2020). Recent trends in applying TPM to cloud computing. *Security and Privacy, 3*(1), e93.
20. Amini, A., et al. (2015). Threat modeling approaches for securing cloud computing. *Journal of Applied Sciences, 15*(7), 953–967.
21. Wang, C., et al. (2011). Privacy-preserving public auditing for secure cloud storage. *IEEE Transactions on Computers, 62*(2), 362–375.
22. Marudhadevi, D., Neelaya Dhatchayani, V., & Shankar Sriram, V. S. (2015). A trust evaluation model for cloud computing using service level agreement. *The Computer Journal, 58*(10), 2225–2232.
23. Kumar, V., & Rathore, R. S. (2018). Security issues with virtualization in cloud computing. 2018 international conference on advances in computing, communication control and networking (ICACCCN). IEEE.

24. Sood, S. K. (2012). A combined approach to ensure data security in cloud computing. *Journal of Network and Computer Applications, 35*(6), 1831–1838.
25. Kao, Y.-W., Huang, K.-Y., Hui-Zhen, G., & Yuan, S.-M. (2013). UCloud: A user-centric key management scheme for cloud data protection. *IET Information Security, 7*(2), 144–154.
26. Bafandehkar, M., Yasin, S. M., Mahmod, R., & Hanapi, Z. M. (2013). Comparison of ecc and rsa algorithm in resource constrained devices. In 2013 international conference on IT convergence and security (ICITCS) (pp. 1–3). IEEE.
27. Canetti, R., Shahaf, D., & Vald, M. (2016). Universally composable authentication and key-exchange with global pki. In *IACR international workshop on public key cryptography* (pp. 265–296). Berlin: Springer.
28. Chung, P.-S., Liu, C.-W., & Hwang, M.-S. (2014). A study of attribute- based proxy re-encryption scheme in cloud environments. *International Journal Network Security, 16*(1), 1–13.
29. Tysowski, P. K., & Hasan, M. A. (2011). Re-encryption-based key management towards secure and scalable mobile applications in clouds. *IACR Cryptology EPrint Archive, 2011*, 668.
30. He, D., Chen, J., & Hu, J. (2012). A pairing-free certificateless authenticated key agreement protocol. *International Journal of Communication Systems, 25*(2), 221–230.
31. Seo, S.-H., Nabeel, M., Ding, X., & Bertino, E. (2014). An efficient certificate- less encryption for secure data sharing in public clouds. *IEEE Transactions on Knowledge and Data Engineering, 26*(9), 2107–2119.
32. Li, J., et al. (2013). Secure deduplication with efficient and reliable convergent key management. *IEEE Transactions on Parallel and Distributed Systems, 25*(6), 1615–1625.
33. Ragab Hassen, H., et al. (2007). Key management for content access control in a hierarchy. *Computer Networks, 51*(11), 3197–3219.
34. Je, D.-H., et al. (2010). Computation- and-storage-efficient key tree management protocol for secure multicast communications. *Computer Communications, 33*(2), 136–148.
35. Kim, Y., et al. (2004). Tree-based group key agreement. *ACM Transactions on Information and System Security (TISSEC), 7*(1), 60–96.
36. Aparna, R., & Amberker, B. B. (2009). Key management scheme for multiple simultaneous secure group communication. In 2009 IEEE international conference on internet multimedia services architecture and applications (IM-SAA) (pp. 1–6). IEEE.
37. Koo, H.-S., Kwon, O., Ra, S.-W., et al. (2009). A tree key graph design scheme for hierarchical multi-group access control. *IEEE Communications Letters, 13*(11), 874–876.
38. Szebeni, S., Butty'n, L., et al. (2012). Tresorium: Cryptographic file system for dynamic groups over untrusted cloud storage. In 2012 41st international conference on parallel processing workshops (ICPPW) (pp. 296–303). IEEE.
39. Xue, K., & Hong, P. (2014). A dynamic secure group sharing framework in public cloud computing. *IEEE Transactions on Cloud Computing, 2*(4), 459–470.
40. Nabeel, M., Shang, N., & Bertino, E. (2013). Privacy preserving policy-based content sharing in public clouds. *IEEE Transactions on Knowledge and Data Engineering, 25*(11), 2602–2614.
41. Xiong, J., Liu, X., et al. (2014). A secure data self-destructing scheme in cloud computing. *IEEE Transactions on Cloud Computing, 2*(4), 448–458.
42. Bethencourt, J., Sahai, A., & Waters, B. (2007). Ciphertext-policy attribute-based encryption. In IEEE symposium on security and privacy, 2007. SP'07 (pp. 321–334). IEEE.
43. Wan, Z., Liu, J.'e., & Deng, R. H. (2012). Hasbe: a hierarchical attribute-based solution for flexible and scalable access control in cloud computing. *IEEE Transactions on Information Forensics and Security, 7*(2), 743–754.
44. Li, M., Yu, S., Zheng, Y., Ren, K., & Lou, W. (2013). Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. *IEEE Transactions on Parallel and Distributed Systems, 24*(1), 131–143.
45. Ostrovsky, R., Sahai, A., & Waters, B. (2007). Attribute-based encryption with non-monotonic access structures. In Proceedings of the 14th ACM conference on Computer and communications security (pp. 195–203). ACM.

46. Kaya, K., & Selçuk, A. A. (2007). Threshold cryptography based on Asmuth-Bloom secret sharing. *Information Sciences, 177*(19), 4148–4160.
47. Blacley, G. R. (1979). Safeguarding cryptographic keys. Proceedings of AFIPS'79 Nat. Computer Conf. (Vol. 48, pp. 313–317).
48. Rabin, M. O. (1989). Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of the ACM (JACM), 36*(2), 335–348.
49. Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612–613.
50. Asmuth, C., & Bloom, J. (1983). A modular approach to key safeguarding. *IEEE Transactions on Information Theory, 29*(2), 208–210.
51. Mignotte, M. (1982). How to share a secret. In *Workshop on cryptography* (pp. 371–375). Berlin: Springer.
52. Geer, D., & Yung, M. (2003). Split-and-delegate – Threshold cryptography for the masses (pp. 220–237).
53. Chokhani, D., et al. (2010). Cryptographic key management workshop summary. NIST interagency report 7609 at computer security division, national institute of standards and technology (pp. 1–18).
54. Nahar, K., & Chakraborty, P. (2020). A modified version of Vigenere cipher using 95 × 95 table. *International Journal of Engineering and Advanced Technology (IJEAT), 9*, 1144–1148.
55. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*, 583–585.

The Architecture of Autonomic Cloud Resource Management

Poorva Shukla, Prashant Richhariya, Bhupesh Kumar Dewangan, Tanupriya Choudhury, and Jung-Sup Um

1 Introduction

Cloud computing [1] as a rising innovation has reformed the data innovation industry by versatile on-request allocation and deallocation of registering assets. Cloud computing [2, 3] is a new rising computing mechanism where the suppliers center around sharing of processing assets through a web-based, versatile serving, and "pay more only as costs rise" way to encourage client's solicitations. With thoughtfulness regarding the quick development of an enormous measure of information producing, the requirement for calculation of assets and collected information is remarkable. Cloud computing [4, 5] provides immense profitable and environmental data collection for the IT industry to decrease the cost of the industry using Industry 4.0 technologies in cloud computing [6–8]. In the cloud computing [9] world, there are services used in cloud [10] technology such as IaaS (Infrastructure as a service), PaaS (platform as a service), and SaaS (Software as a service). Due to this, a few arrangements were applied to arrive at proficient energy, vitality the board for these Datacenters, for example, improving IT foundations (Servers and stockpiles, organize gear), structuring equipment with vitality productive designs, vitality mindful occupation booking, power appropriation methods, dynamic voltage and

P. Shukla (✉) · P. Richhariya
Department of Computer Science and Engineering, IIST, Indore, India

B. K. Dewangan · T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India
e-mail: tanupriya@ddn.upes.ac.in

J.-S. Um
Department of Geography, College of Social Sciences, Kyungpook National University, Daegu, South Korea

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_14

recurrence scaling (DVFS), advanced setup and force interface strategies, power the executive procedures (union, provisioning, and virtualization), turn unused hubs to turning off mode, and different strategies [11, 12]. Industry 4.0 (the 'fourth mechanical upset') is the advancement to raise computerization, M-to-M and H-to-M correspondence, man-made reasoning, i.e., AI, and proceeded with innovative enhancements and modification in assembling.

Industry 4.0 has the following four interruptions:

- Increase in data generation.
- Analysis of generated data and connectivity.
- Development of examination and business insight capacities—for example new types of human–machine communication, for example, contact interfaces, and enlarged reality frameworks [11, 12].
- Upgrades in moving computerized directions to the physical world, for example, propelled mechanical technology and 3D printing.

The meaning of cloud computing is that data can be stored and accessed over the Internet instead of our personal home computer and network. The cloud is called a metaphor for the Internet. To decrease the upkeep cost of registering situations, the organizations are progressively incorporating their registered foundations which are along these lines overseen by explicit organizations which we call suppliers.

1.1 What is Autonomic Computing

Autonomic computing is one of the new terminologies used in the field of cloud computing, and it is a PC's capacity to oversee itself consequently through versatile advances that further figure abilities and cut down on the time required by PC experts to determine framework troubles and other upkeep, for example, programming refreshes. The concept of autonomic computing is developed by IBM. It is one of the autonomic systems that does not involve human interaction and that type of computing believes in autonomic computation without much interference from end users. IBM explains the four major fields of autonomic computing

1. Self-agreement or configuration.
2. Self-remedy or healing.
3. Self-control of autonomic resources (awareness).
4. Self-identification from intruders attacks.

There are four main fields of autonomic computing, where the AC works independently without the interference of human interaction. Now the emerging technology, autonomic computing system, works on 3 As which include Automation [13], Adaptivity, and Awareness. The mechanism of autonomic computing works like nervous system of the human body; it purely works on the human nervous system. An autonomic nervous system takes all decisions independently and reacts accordingly. In autonomic computing, the nervous system of autonomic computing

behaves according to the nature of the data, and full consciousness of input data means data are entered and what environment they require to execute. Functions of autonomic computing environment used a high level of AI (Artificial intelligence), while the rest of the data is invisible to the user.

1.2 Why Autonomic Computing Is Used

Autonomic figuring or computing is one of the structure squares of inescapable processing, a foreseen future registering model in which small—even imperceptible—PCs will be surrounding us, imparting through progressively interconnected systems prompting the idea of Internet of Everything (IoE). Numerous industry chiefs explore different parts of autonomic figuring and autonomic computing. The main reason to use autonomic computing is to minimize the cost of purchasing softwares. The main advantage of using autonomic computing is to minimize the environmental cost, continuity cost purchasing price packet delivery ratio, throughput time, delay time, and improving reliability of the IT industry system. All this is possible when we use autonomic computing in Industry 4.0. Besides, why do we use autonomic computing? The major reason to use autonomic computing is to give an opportunity to another industry or companies to run their own business very smoothly online only based on cloud. Due to this, they are able to accept other business policies without the need for any basic platform because autonomic computing fulfills the need of industry policy. They also give an update on modified applications based on the changing environment. Now, with the use of autonomic computing, we can increase reliability, storage, availability, and reduce human efforts or manpower cost to maintain large programs, software, and applications on the server.

The future of autonomic computing is very bright because autonomic processing vows to streamline the administration of figuring frameworks. Be that as it may, that ability will give the premise to substantially more powerful cloud computing. Different applications incorporate server load adjusting, process allotment, observing force gracefully, programmed refreshing of programming and drivers, pre-disappointment cautioning, memory mistake amendment, computerized framework reinforcement and recuperation, and so forth.

1.3 Role of Autonomic Cloud Resource Management in Autonomic Computing

The main role of cloud resource management in autonomic computing is the self-agreement, or the self-configurable nature of cloud resources means they are able to work independently. Cloud computing agreements are similar to traditional software

licensing agreements but often have more in common with hosting or application service provider agreements. So they can make an agreement to consult the hosting service provider as per their requirements without human interference because autonomic cloud resource management works under autonomic computing, and autonomic computing works based on the human body's nervous system. Autonomic cloud resource management system can accept the changing environment and they automatically interact with their nearest system. These kinds of cloud resources make communication with the neighboring cloud resources with an appropriate communication protocol. The cloud resource management system is much more intelligent compared to traditional because they know which resource is capable, what limitation they have, which communication protocol is used and how, and why they are communicating with the connected resources. The management of cloud resources depends on their configuration capability, which is much higher than that of traditional cloud resources because traditional cloud resources are managed by human nervous system itself, but in autonomic cloud, resource management is used in autonomic computing; so they are intelligent systems and they know where and which resource needs updated configuration and where they need service provider agreements. That is why we say that this autonomic management system is self-agreement or configuration, self-remedy or healing, self-control of autonomic resources (awareness), and self-identification from intruders attacks [14].

The autonomic systems are capable to configure and reconfigure automatically which totally depends upon changes in real-time computing environment and make the system very optimistic and deployable in an efficient computing process. The process flow of interaction with autonomic manager is shown in Fig. 1.

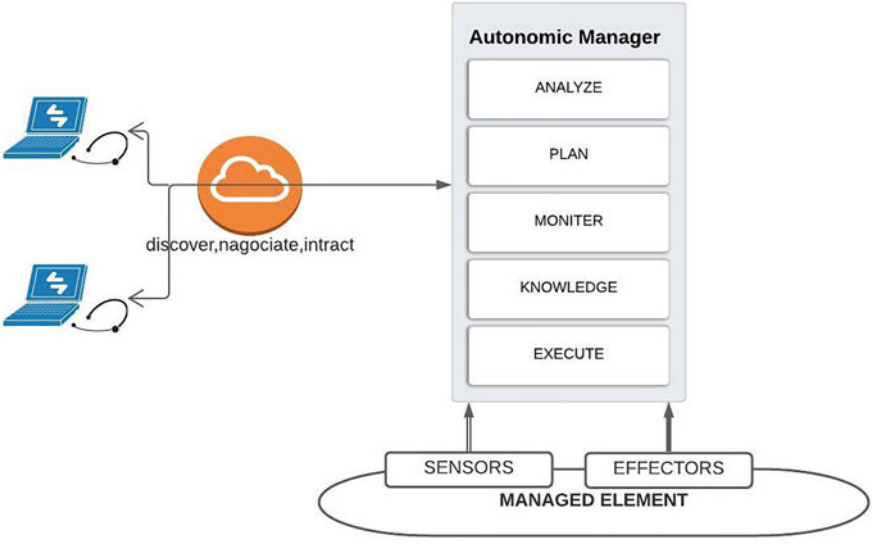

Fig. 1 Interaction with autonomic manager

2 Literature Survey

A writing survey gives the setting, advises philosophy, expands development, stays away from duplicative exploration, and guarantees that proficient principles are met. Writing audits require some investment, are iterative, and should proceed all through the examination procedure.

The main reason for a writing survey is to spot each work with regard to its commitment to understanding the examination issue being considered; portray the relationship of each work to the others viable; and recognize better approaches to decipher earlier examination.

2.1 Investigation Method

There are so many aspects where we can investigate the result. In this chapter, we read 40 research papers and Internet articles related to architecture of cloud resource management and deep study of architecture of autonomic cloud resource management, but only references 20 research papers are included in this chapter.

Dewangan et al. [15] describe autonomic cloud resource management system. In cloud resource management, autonomic computing is one of the famous models in business industry that is useful for service providers, and it gives benefits in terms of cost, and service providers can use these cloud resources, but the main thing is in what manner these resources are allocated among customers in the Industry 4.0. Various researchers researched cloud resources, and they succeeded in this field, but the need is to make efficient use of resources through which we can save our atmosphere. Now, in this article the author gives an efficient approach for using autonomic cloud resources based on classification of the current study like techniques used in cloud resource management functions. Now cloud resource management is most popularly used in the business model. Autonomic cloud resource management is highly used in Industry 4.0 which gives profit to each and all IT industry and business industry. Autonomic cloud resource management is based on on-demand and pay-per-use methods; whenever the need arises then ACRM is used to allocate adoptable resources that are compatible for the required platform. Here some concept of virtualization and virtual machine is used. ACRM is fully based on on-demand policy [16–18]. So multiple autonomic cloud resources are available. ACRM gives facility to choose compatible cloud resources as per the user demand; but without the interference of human power, they automatically understand what is the need of the user to run particular application or softwares. So free resource pools are available. The ACRM chooses compatible resource for particular user applications or softwares and allocate among customers (end users). Once the user completes their work, all the compatible resources which allocate among end users at the time of execution are deallocated and free the cloud resources in cloud. After the completion of execution, it means the allotted resources are free

if another user wants to pick the particular resource in the resource pool, they take it and work for its execution process.

The use of autonomic cloud resources directly affects the price of services that are provided by the cloud. Now the demand for cloud resources is increasing day by day. Due to this increasing demand of cloud resources, the service provider has to maintain the quality of service provided to customers' security, service-level agreement (SLA) violations, rate energy utilization by resources, price, rate of packet transfer, throughput, delivery ratio, etc. During the survey we found that energy utilization has decreased through self-improvement (optimization), SLA violation rate is reduced by self-remedy (Healing), and find defective virtual machine from cloud resource pool and make a separate resource pool of defective virtual machine. In proceeding, the working expense of assets improves, and reduced execution time has been recorded [16]. The author in autonomic cloud resource management gives an architecture, where the cloud resource management gives an architecture where the cloud users collect the data sets of workloads and submit it for clustering where all the workload data sets are divided into clusters. Depending on the working of each cluster, the resource manager is responsible for QOS and SLA of each cluster.

In self-optimization method, check whether the data sets give an optimal solution or not according to their fitness value. If it meets optimal solution, then they choose the compatible resources in the resource pool. But if the optimal solution does not meet, then data sets check their fitness value again until and unless it gives an optimal solution. During the optimization, some fault may occur and the fault prediction mechanism finds whether node failure occurs; if yes then the fault manager tries to the resolve node performance, but if node failure does not occur, then this node is replaced with another one after that resource scheduler prepares their schedule for execution. After that it is submitted to the cloud user to check QOS and SLA for satisfactory result. Some problems are trying to reduce problems in their proposed work for better performance. The problems are as follows.

- Service-level agreement violations rate.
- Execution time would be very high.
- Use of resource utilization.
- Improve fault tolerance technique.
- Price of resources is high.
- More energy consumption.

Distributed computing gives a stage where administrations are encouraging the cloud client through the web, either liberated from cost or rent base. The cloud client and requests are expanding, and because of this, a huge number of administration demands are submitted to the cloud specialist organization. The similar examination of all the studied calculations as far as various execution measurements. The perception of the overview gives some exploration holes to improve the productivity of the current asset in the board framework [17].

In this paper, the author gives the concept of STAR (Standardized Testing and Reporting). The principle reason to give STAR is to limit the SLA infringement

Table 1 The comparative analysis of different autonomic computing techniques in cloud are presented in Table 1.

Strategy given by	SLA	ACRM	PBM	IT	Industry 4.0	Method
Mehrdad Maeen et al. [6]	√	√	√	×	√	×
Lenk et al. [11]	×	√	×	√	×	√
Bahrami et al. [12]	√	√	√		√	√
Pooja Dehraj et al. [14]	×	√	√	√	√	√
Choudhury et al. [15]	√	√	√	√	√	√
Agarwal et al. [17]	√	√	√	√	√	√
Venkatadri et al. [19]	√	√	√	√	√	√
Pasricha et al. [20]	√	√	√	√	√	√
Dewangan et al. [21]	√	√	√	√	√	√
Sukhpal Singh and Inder et al. [18]	√	√	√	×	×	√
Qiu et al. [22]	×	√	√	√	×	√
Özer et al. [18]	√	×	√	×	×	×
Yumin Wang et al. [23]	√	×	√	×	√	√
Atul Vikas Lakra et al. [24]	√	√	×	√	×	×
Saraswathi and Kalaashri et al. [25]	√	√	×		×	√
Tahir M et al. [26]	√	×	√	√		√
Sukhpal Singh et al. [27]	×	√	×		√	√
Salah [28]	√	×	×	√	√	√
Salah et al. [29]	√	×	√	√	√	√
Nazir et al. [30]	×	√	√	×	×	√
Vikas Mangotra et al. [31]	×	√	√	×	×	√
Nima Jafari Navimipour et al. [32]	×	×	√	√	×	√
Anjum Mohd Aslam et al. [33]	×	×	√	√	×	√
Aarti Singh et al. [34]	√	×	√	√	×	√
Son et al. [35]	×	×	√	√	×	√
Dewangan et al. [13]	√	√	√	√	√	√

and upgrade the client fulfillment by giving their QoS as required (Table 1). In this paper, creator contemplations, the accompanying SLA like execution dependent on cost, inactivity, execution time, availability, and unwavering quality to limit the SLA infringement to fulfilled QoS. This calculation is actualized and executed in a real cloud condition at Thapar University, and the results show the exhibition for SLA infringement is better as far as an existing asset the board procedures [18]. Planning can be expressed as an occasion to occur at a specific time. There are numerous kinds of planning calculations accessible in disseminated figuring for asset booking. Numerous calculations are to be used in the circulated framework by proper validation. The motivation behind the booking calculation is to accomplish extreme throughput. For a cloud situation, the standard methodologies cannot accomplish the ideal effectiveness [17].

Resource planning for distributed computing is an apparatus, which influences the operational expense of the specialist co-op and the cloud client. Numerous

scientists have been working toward resource planning in various angles like burden adjusting, make span, remaining tasks at hand need, asset accessibility, and cost. In this paper, the author will diverse asset planning systems and their methods, and perceptions of this examination show that a large number of these structures are not completely computerized and based on a smaller number of execution target work, and the assets are booked. Because of the expanding administration request, this work likewise finds that the accommodation of outstanding tasks at hand by cloud client to booking the assets, and computerized approach is required [17, 18].

The author gives the concept of self-improved vitality of effective resource executives methodology has been considered, which gives the ideal answer for expanding the resource usage and distinguish the defective resource to protect from misdirecting of planning. The executives in cloud are basic necessity for specialist co-op and cloud client too. The resource provisioning in the cloud should be low in cost and execution. In view of these two boundaries, the accompanying perception completed: SLA infringement rate is high, execution time can be less, scope to boost the resource usage, energy utilization can be less, fault-tolerant strategies should be executed to distinguish broken VM's, and resource cost can be less. Energy utilization and some faulty resources are two key points. The resources are overseen by selecting the best VM esteem through self-improvement, and defective VM is distinguished through self-mending qualities. The quality virtual machine is distinguishing the remaining burdens to other VMs, so the virtual machine can be used for better outcomes. The rate of energy utilization and effectiveness investigation is registering and found that it is performing better [22].

The supplier's compensation has three fragments: the compensation of the favorable position, the expense of the VM occasions that regulate to the clients, and the expense of keeping whatever is left of the advantages sit out of apparatus and Proposes a model, called ABRA (Auction-Based Resource Co-Allocation) to deal with the benefit allotment issue. It powers discipline costs on unallocated resources after a closeout with a particular ultimate objective to improve the asset use [23, 24].

Autonomic cloud computing provides an environment where there are so many services available for users for where the basic need is only Internet connectivity, and the user may use this facility either cost free or pay per use. The cloud services demand is increasing day by day. Due to increasing environment, we have to scale out the existing policies. Scaling comes at the expense of substantial vitality utilization because of the incorporation of various server farms and servers. The superfluous force utilization influences the working costs, which thus, influences its clients. In this paper, the author proposes simulated cloud resource allocation conditions and figures vitality utilization for various outstanding tasks at hand amount and it expands the exhibition of various multi-goals capacities to amplify the asset usage. It contrasted and existing structures and examination results show that the proposed system performs most extreme.

The ESCORT system was introduced to advance vitality utilization, execution cost, and SLA infringement rate. This structure is actualized and reenacted in the clouds condition. The outcome may differ when it will be executed in the genuine

cloud. In this, it is introduced in a point-by-point stream of ESCORT that, how to apply the proposed system to limit the vitality utilization, SLA infringement rate, execution time, and cost to augment the exhibition for planning in the cloud. The examination of proposed work with different procedures guarantees that ESCORT performs most extreme. ESCORT recreation results limit the SLA infringement rate, execution cost, and amplify the use of the asset. The confinement of this work is that no separating strategies are applied for pernicious remaining task at hand; in future, it will be actualized to ad lib the proposed work [23]. To minimize the computing cost by which increasing demand of companies or IT sectors and IT sector elaborate their infrastructure. So, the cloud resources are in high demand.

The increased market will be maintained by the service provider or call providers. They have to deliver quality services because the overall expenditure of the market depends upon the industries or IT sectors. On-request asset allotment is one of the primary administrations that such a situation must guarantee. It must permit the portion of asset varying and asset de-allocation when they are not utilized any longer. This paper depicts various situations, which comprise guaranteeing dynamic asset allotment for a bunched J2EE application conveyed in a facilitating focus. It very well may be utilized to screen applications in a facilitating focus and at whatever point it is required, allot another machine, send the necessary programming parts on that hub and reconfigure the application to incorporate these new segments [19]. The expression "asset provisioning" has been characterized in various settings found in the literature. We saw the constant commitment has been made by International Symposium on Cluster, Cloud, and Grid Computing (CCGrid) in the field of cloud asset provisioning for head way of exploration. Ongoing exploration portrays that powerful asset provisioning systems give better asset booking. It is exceptionally hard to track down the best asset and outstanding task at hand pair for productive planning. So, it is recommended that as opposed to recognizing outstanding task at hand and asset, we ought to have the legitimate determination of asset and QoS necessities of remaining burdens for better asset the board [35].

2.2 Problem Identification and Challenges in Autonomic Cloud Resource Management

Autonomic cloud computing is very advanced technology in Industry 4.0 which makes our work easy and simple, with the basic requirement of proper Internet connectivity in the user's point of view. In autonomic cloud computing technology, autonomic cloud resource management is one of the major problems from provider's point of view. Major problems include SLA violation rate, execution time is very high, resource utilization, load stability, timing slot per job, cost, vitality improvement and accessibility, fault-tolerance problem in faulty node due to which resources are allocated to faulty node and if faulty node occupies the resources the

other resources are in waiting state, so it creates major issues in autonomic cloud resource management. Hence, we need a mechanism that finds nature of executing job and then decide which resource is compatible for that particular job (Table 2).

3 Methodology

In this architecture, autonomic cloud resources are fully used by the executing job. After the execution of job, this mechanism has to follow some mechanism. It means that this mechanism first identified at the user level. Data sets and overall workloads will be available with the end user itself, and if data set is large then it splits it into number of clusters (small packets) and work on each cluster, and if data set is smaller, it can directly check its configurability with autonomic cloud resource management in Industry 4.0. Here the mechanism automatically checks whether the node is self-configurable (optimizable) or not; if it gives optimal solution, then the resource will be allocated in the resource pool to each and every node to execute their job completely on time. Now the second part shows that if the node does not give any optimal solution, then it goes for fault-tolerance mechanism where they can identify the faulty node; if faulty node is present then this faulty node is replaced with a new node. Then it is again submitted to end users, but if no faulty node is identified by fault tolerance mechanism it is then again submitted for optimization method, where the data set is again optimized for efficient resource allocation. Autonomic computing plays an important role in autonomic cloud resource management in Industry 4.0. Industry 4.0 is revolution in the field of industry or fourth revolution which is mainly used for automation.

The main components used in flow diagram presented in Fig. 2:

USER:
User is one of the main important entity that is related to real-time action. User is the one who gives instructions to other entity in real time, and in this architecture, the user works over different types of data sets. This is also called real-time entity.

Larger data set:
Larger data set means whose size is greater than defined size. Larger data sets need to be split into different data sets which is called clustering.

Smaller data set:
Smaller data set means whose size is smaller than the defined size. It means no need to split into different data sets.

Self-configurable:
The term self-configurable in architecture of autonomic computing in resource management is any node that requires compatible resource to execute the data; they do not need to wait for any other entity or user. They provide the resource system automatically provide resource as per the requirement of resource means automatically configurable without the interference of human brain.

Table 2 Comparative analysis of different frameworks used in different technologies in autonomic cloud management

S.no	Framework	Step by step processor	Reason (parameters)	Evaluation technique	Platform
1.	Allocation scheme of resources	Resource scheduling	Based on priority	Autonomic cloud resource management	Cloud
2.	STAR	Self-maintenance	Rate of SLA violation	Price, throughput, reliability	Cloud
3.	ACRM	Self-healing (configuring) management	VM	Reduced execution time	Cloud
4.	PBM	Process scheduling	Depends on data set workload	Execution cost	Cloud
5.	Optimal resource utilization	Antlion optimization	Resource utilization (to remove faulty resources)	Qos and SLA violation	Cloud
6.	Static behavior of system	Genetic–algorithm ant colony swarm-optimization	SLA violation & Qos	–	Cloud
7.	Self-characteristics scheduling	Energy-efficient resource scheduling	Better resource utilization	Operative cost and execution time	Cloud
8.	Dynamic optimization	Self-characteristics	Without human interaction	Find energy cost and time	Cloud
9.	PC and electronic framework	Continuous administration	Innovation	Fault tolerance	Cloud
10.	Fault-tolerant management	EAR	VM based rejection of faulty machine	Energy consumption	Cloud
11.	Cloudsim toolkit		SLA violation	Operative cost, energy efficiency	Cloud
12.	Resource allocation issue	ABRA (auction-based resource allocation)	Unallocated resources	–	Cloud
13.	OCRP	OCRP algorithm	Provisioning assets	Cost for assets	Cloud
14.	Data centers	Vitality utilization	Qos and reduce energy consumption	Cloud resources in data center	Cloud

(continued)

Table 2 (continued)

S.no	Framework	Step by step processor	Reason (parameters)	Evaluation technique	Platform
15.	ESCORT	Energy consumption algorithm	SLA violation	Execution cost	Cloud
16.	Industry 4.0	ACRM	Self-configurable		Cloud
17.	KMGA	Swarn optimization and genetic algorithm	Minimized number of VMs	Reduce energy consumption of Datacenters and Qos	Cloudsim tool
18.	MyDAQ		Laboratory experiment	Reduced the cost of hardware and computation	Cloud
19.	Deadline based resource provisioning and scheduling algorithm	DBRPSA	Scheduling of resources automatically	Execution time and cost	Cloud
20.	Cluster,cloud & grid computing	CCGrid	Cloud resource provisioning	Proper Qos specification	Cloud
21	Task scheduling	Cuckoo search algorithm	Automatically resource scheduling	Scheduling process	Cloud

Fault detection:

The term fault detection in architecture of autonomic computing in resource management is error. If error occurs due to connection establishment between source to destination and in resource configuration or battery backup, then it would be automatically detected and recovered with the help of given architecture.

Resource pool:

Multiple resources are available in one place; whenever any node requires any resource, they can opt from the resource pool.

Qos:

Quality of service means providing good services without any problem.

SLA violation:

SLA (Service-Level Agreement) which gives full agreement between the communicating parties means if both the sending and receiving parties have SLA then its fine, but if they don't have agreement, then this comes under SLA violation and means this parties unauthorized access.

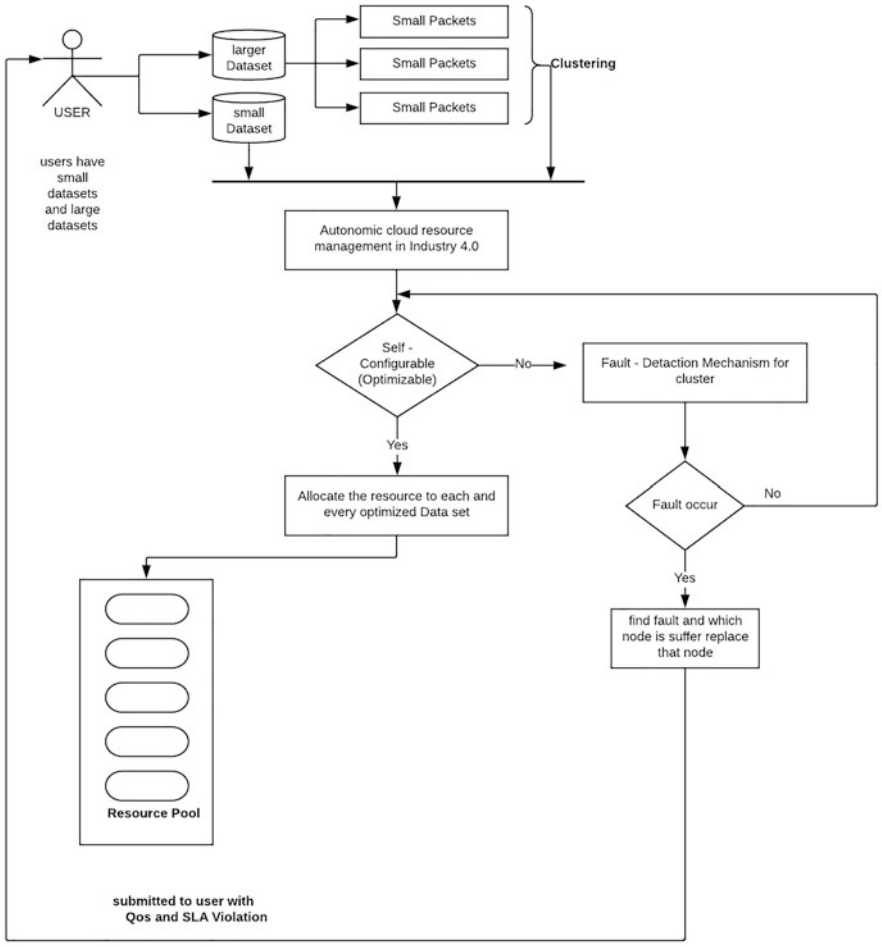

Fig. 2 Architecture for autonomic cloud resource management

4 Conclusion and Future Research Scope

Industry 4.0 means the fourth revolution in the field of industry or fourth revolution which is mainly used for automation M to M, H to M communication, digitization, etc., and exchanging of data in different technology. This is very beneficial for cloud computing automatic technologies and IOT. Autonomic cloud computing is an advanced technology in Industry 4.0 which makes our work easy and simple with only basic requirement of proper Internet connectivity from the user's point of view. In autonomic cloud computing technology, autonomic cloud resource management is one of the major problems from the provider's point of view. Autonomic cloud computing provides an environment where there are many services available for

the users and the basic need is only Internet connectivity, and the user may use this facility either cost free or pay per use. Nowadays, this cloud services demand is increasing day by day. Due to increasing environment, we have to scale out the existing policies. The scaling comes at the expense of substantial vitality utilization because of the incorporation of various server farms and servers. The superfluous force utilization influences the working costs, which, thus, influences its clients. In this chapter, the architecture of autonomic cloud resources is fully used by the executing job. After the execution of job, this mechanism has to follow some mechanism and resolve some problem issue using autonomic cloud resource management in Industry 4.0.

Appendix

Qos	Quality of service
ACRM	Autonomic computing and resource management
VM	Virtual machine
GA	Genetic algorithm
DVM	Digital variable multisystem
PBM	Profit base management
SLA	Service level agreement
IT	Infrastructure technology
RPS	Resource provisioning strategy
OCRP	Optimal cloud resource provisioning
PSO	Particle swarm optimization
MHOD	Markov host overload
RPM	Resource provisioning mechanism
FM	Fault tolerance mechanism
RIC	Resource information center
CPU	Central processing unit
CCGrid	Cluster cloud grid algorithm
STAR	Standardized testing and reporting
PC	Personal computer
EAR	Energy-aware resource
ABRA	Auction based resource allocation
ESCORT	Energy consumption
KMGA	k-means genetic algorithm
MYDAQ	Data acquisition (DAQ)
DBRPSA	Dead line-based resource provisioning and scheduling
IOE	Internet of everything

References

1. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2017). Privacy and security of cloud-based internet of things (IoT). 2017 3rd international conference on computational intelligence and networks (CINE) (pp. 40–45).
2. Sharma, A., Choudhury, T., & Kumar, P. (2018). Health monitoring & management using IoT devices in a cloud based framework. 2018 international conference on advances in computing and communication engineering (ICACCE) (pp. 219–224).
3. Mittal, A., Khan, F. S., Kumar, P., & Choudhury, T. (2018). Cloud based intelligent attendance system through video streaming. Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358587.
4. Kumra, S., Choudhury, T., Nhu, N. G., & Nalwa, T. (2018). Challenges faced by cloud computing. Proceedings of the 2017 3rd international conference on applied and theoretical computing and communication technology, ICATccT 2017. https://doi.org/10.1109/ICATCCT.2017.8389105.
5. Dinote, A., Sharma, D. P., Gure, A. T., Singh, B. K., & Choudhury, T. (2020). Medication processes automation using unified green computing and communication model. *Journal of Green Engineering, 10*(9), 5763–5778.
6. Maeen, M., Haghparast, M, & Askarizad, M. (2018). An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing iaas platforms. © Springer Science+Business Media, LLC, part of Springer Nature.
7. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-Based Outsourcing Framework for Efficient IT Project Management Practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
8. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. In Data engineering and intelligent computing.
9. Yadav, A. K., Tomar, R., Kumar, D., & Gupta, H. (2012). Security and privacy concerns in cloud computing. In Computer science and software engineering.
10. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
11. Lenk, A., Klems, M., Nimis, J., Tai, S., Sandholm, T. (2009). What's inside the cloud? An architectural map of the cloud landscape. *IEEE Computer Society*.
12. Bahrami, M., & Singhal, M. (2015). The role of cloud computing architecture in big data. In *Information granularity, big data, and computational intelligence*. Cham: Springer. https://doi.org/10.1007/978-3-319-08254-7_13.
13. Faruqui, N., Yousuf, M. A., Chakraborty, P., & Hossain, M. S. (2020). Innovative automation algorithm in micro-multinational data-entry industry. In Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325 LNICST, pp. 680–692). Springer. https://doi.org/10.1007/978-3-030-52856-0_54
14. Dehraj, P., & Sharma, A. (2020). A review on architecture and models for autonomic software systems. Springer.
15. Choudhury, T., Agarwal, A., Pasricha, A., & Chandra Satapathy, S. (2020). "Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges. *Software: Practice and Experience*. https://doi.org/10.33889/IJMEMS.2020.5.4.060
16. Sukhpal S., Inderveer, C., & Buyya, R. (2016). STAR: SLA-aware autonomic management of cloud resources. IEEE transactions on cloud computing (pp. 1–14).

17. Agarwal, A., Venkatadri, M., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science, 152*, 204–211. https://doi.org/10.1016/j.procs.2019.05.044.
18. Özer, A. H., & Özturan, C., (2009, September). An auction based mathematical model and heuristics for resource co-allocation problem in grids and clouds. In Soft computing, computing with words, and perceptions in system analysis, decision and control, 2009. ICSCCW 2009. Fifth international conference on (pp. 1–4). IEEE.
19. Venkatadri, M., Agarwal, A., & Pasricha, A. (2018, December). Autonomic cloud resource management. In 2018 fifth international, conference on parallel, distributed and grid computing (PDGC) (pp. 138–143). IEEE. https://doi.org/10.1109/PDGC.2018.8745977.
20. Pasricha, A., Agarwal, A., & Venkatadri, M. (2019). Energy-aware autonomic resource scheduling framework for cloud. *International Journal of Mathematical, Engineering and Management Sciences, 4*(1), 41–55. https://doi.org/10.33889/IJMEMS.2019.4.1-004.
21. Dewangan, M. B. K., & Shende, M. P. (2012). Survey on user behavior trust evaluation in cloud computing. *International Journal of Science, Engineering and Technology Research, 1*(5), 113.
22. Qiu, X., Dai, Y., Xiang, Y., & Xing, L. (2017). Correlation modeling and resource optimization for cloud service with fault recovery. *IEEE Transactions on Cloud Computing, 5*(1), 1–13.
23. Yumin, W., Li, J., & Haoxiang Wang, H. (2017). Cluster and cloud computing framework for scientific metrology in flow control. Springer.
24. Lakra, A. V., & Yadav, D. K. (2015). Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. In *International conference on intelligent computing, communication & convergence* (pp. 107–113). Bhubaneswar: Elsevier.
25. Saraswathi, A. T., Kalaashri, Y. R. A., & Padmavathi, S. (2015). Dynamic resource allocation scheme in cloud computing. *Procedia Computer Science, 47*, 30–36.
26. Tahir, M., Ashraf, Q. M., & Dabbagh, M. (2019). Towards enabling autonomic computing in IoT ecosystem. In 2019 IEEE international conference on dependable, autonomic and secure computing, international conference on pervasive intelligence and computing, international conference on cloud and big data. computing, international conference on cyber science and technology congress (DASC/PiCom/CBDCom/CyberSciTech) (pp. 646–651). IEEE.
27. Sukhpal, S., & Inderveer, C. (2016). Cloud resource provisioning: survey, status and future research directions © Springer-Verlag London.
28. Salah, K. (2013). A queueing model to achieve proper elasticity for cloud cluster jobs. In 2013 IEEE sixth international conference on CLOUD computing (CLOUD). IEEE.
29. Salah, K., Calero, J. M. A., Zeadally, S., Al-Mulla, S., & Alzaabi, M. (2013). Using cloud computing to implement a security overlay network. *IEEE Security and Privacy, 11*(1), 44–53.
30. Nazir, S., Patel, S., & Patel, D. (2017). Autonomic computing meets SCADA security. In Proceedings of 2017 IEEE 16th international conference on cognitive informatics and cognitive computing, ICCI* CC 2017. London South Bank University (pp. 498–502).
31. Vikas, M., & Richa, D. (2018). Cloud reliability enhancement mechanism: A survey. *International Journal on Scientific Research in Computer Science and Engineering, 6*(3), 31–34.
32. Nima, J., & Navimipour, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. *International Journal of Modeling and Optimization, 5*(1), 44–47.
33. Aslam, A. M., & Jaur, M. (2018). A review on energy efficient technique in green cloud: Open research challenges and issues. *International Journal on Scientific Research in Computer Science and Engineering, 6*(3), 44–50.
34. Aarti, S., & Dimple, J. (2015). Autonomous agent-based load balancing algorithm in cloud computing. International conference on advanced computing technologies and applications (pp. 832–841).
35. Son, S., & Jun, S. C. (2013). Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing. Paper presented at Proceedings of the 2013 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid) (pp. 168–171). IEEE.

Towards Industry 4.0 Through Cloud Resource Management

Minakshi Sharma, Rajneesh Kumar, Anurag Jain, Bhupesh Kumar Dewangan, Jung-Sup Um, and Tanupriya Choudhury

1 Introduction

The emergence of various paradigms in computing, such as distributed computing, cluster computing, and grid computing, led to the growth of cloud computing [1, 2]. It is termed as "Cloud" [3, 4], as the information accessed by the user is found in a virtual space or remotely. Industries' work in the field enables the user to save applications used and work files on these virtual resources remotely and accessing them via the Internet. It facilitates its users by providing the hardware and software applications and a development platform along with various tools as resources. Such resources are delivered as a service to a cloud [5, 6] user. These aforementioned types of services are Infrastructure-as-a-Service (IaaS), and latter two are Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS), respectively.

M. Sharma · R. Kumar
Department of Computer Science and Engineering, MMEC, Maharishi Markandeshwar Deemed to be University, Ambala, India
e-mail: drrajneeshgujral@mmumullana.org

A. Jain (✉)
Virtualization Department, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India
e-mail: anurag.jain@ddn.upes.ac.in

B. K. Dewangan · T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

J.-S. Um
Department of Geography, College of Social Sciences, Kyungpook National University, Daegu, South Korea

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_15

For providing the services to its users, it relies upon a service-oriented architecture (SOA) to share the resources pooled together in one place, and they can use these services via a network by configuring handheld devices. The resources provided to users to serve their requests are created using virtualization. Each physical host is capable of accommodating multiple virtual machines based on user requests depending on its hardware configuration. Virtualization is the key feature of cloud computing [7, 8] that aids in providing virtual resources to serve user requests [9]. These resources are released to a free pool of resource eventually after serving user requests. So, management of these resources one of the key aspects in a cloud environment which is considered as a challenge in the field due to the variability of load heterogeneity and unpredictability of resource types [10].

The process of resource management in the cloud environment can be defined as the allocation of resources, such as storage, computing, networking provided to a set of applications to meet the performance metrics to satisfy user demands and service providers' quality of service(QoS) parameter collectively. Resources are allocated to users within specified constraints and restrictions imposed by the service provider to furnish the services to give the consented benefit to the user based on service-level agreement. Resource provisioning and its management is a complex task to handle in the cloud system due to the sharing of resources and unpredictable demand patterns.

Cloud computing also provides an environment for businesses to adapt the contemporary technologies by providing various types of resources. It also aids in processing their applications seamlessly by optimizing their business processes. In this way, the cloud enables new business innovations and processes. Integrating cloud with a production process to produce customized products by digitizing components and elements involved in manufacturing is a step toward a new revolution Industry 4.0 [5]. Thus, managing resources efficiently in the cloud is a step toward the Industry 4.0.

To lay the foundation of our study based on resource management, the cloud environs chapter has been divided into various sections. Section 1 discusses the need for resource management, Sect. 2 explains about resource management policies and mechanism, and a general model for resource management has been discussed in this section including factors that affects the resource management process. Section 3 represents a taxonomy based on surveyed resource management techniques. Section 4 discusses the various performance metrics, issues, and challenges in the field.

1.1 Resource Management Is Essential

In cloud environs, the services delivered to users depend upon the heterogeneous and dynamic nature of resources that process user tasks obligated by service-level agreements(SLAs). The SLA is an agreed-upon document between user and cloud service provider that guarantees the QoS requirements provided to the

user. Therefore, fulfilling these QoS requirements, heterogeneity of resources, the variability of workload, and uncertainty of resources brings challenges in the field, and traditional resource management techniques are unsuitable for such a system [11].

Also, new market-oriented techniques devised for the environment have not incorporated the solutions to determine the global state of the system, which is not feasible if the system is built of a huge number of servers distributed over a broad geographical network. Therefore, the state of the system that includes the dispersion of resources with unpredictable load changes very rapidly [12]. Thus, the need for autonomic resource management based on self-management principles arises in the cloud environment.

1.2 Research Scope and Motivation

In cloud computing, resource management is an important concept, as it directly affects the three significant parameters, such as performance, system functionality, and cost. An inefficient approach for the management of resources can directly affect the system performance and related cost that can have an adverse impact on system functionality. With a large infrastructure including public, private, community, and hybrid cloud, companies surely need to consider the resource management concepts during strategic planning for cloud computing [13]. Consequently, this study emphasizes the different resource management techniques.

The objective of the presented research work is to understand the need for resource management in the field. To uncover the concept, various aspects related to resource management have been presented. A comprehensive survey has been conducted to cover various studies conducted for resource management techniques in cloud environs. Taxonomy has been designed based on the diversity of techniques in resource management.

2 Classification of Resource Management(RM) Policies

Cloud computing constitutes a complex environment with multiplexing of resources that are shared subject to process fluctuating load and uncertainty of these resources due to external events make the system difficult to regulate. A strategic formation for the aforementioned system is extremely complex. Therefore, to achieve a multi-objective quality of service(QoS)-based resource management, complex policies and decision variables are required. These policies of resource management can be loosely divided into five different categories based on the functionality it performs.

Admission control
Capacity allocation
Load balancing
Energy optimization
Guarantee of Quality of services (QoS).

Admission control: In this, external workloads are not accepted if system high-level policies get violated. For example, system hindrance to accepting new workload to complete the job already in the execution state.

Capacity allocation: These policies are based on resource allocation for individual activation of service or instance. Capacity allocation is done for an instance based on the demand.

Load balancing: These policies are based on equal distribution of load among all the nodes or servers.

Energy optimization: The policies based on load balancing and energy optimization are interrelated. These policies are used for the optimization of energy and reduction in CO_2 emission.

Guarantee of quality of service (QoS): These types of policies are based on SLAs and satisfy response time and other constraints specified in it.

These policies can be implemented by several mechanisms that include control theory, utility-based, machine learning, and market-oriented mechanisms [11].

2.1 Resource Management (RM) Mechanisms

A mechanism refers to means by which a policy can be implemented. In cloud computing techniques, resource allocation should be based on the disciplined approach instead of using ad hoc methods [9]. There are four basic mechanisms in which these RM policies can be implemented, and these are as follows:

Control theory: The control theory mechanism is used to predict the local behavior of the system instead of global. It guarantees the stability of the system by using the feedback control, and it can also predict the transient behavior [9].

Machine learning: To apply these techniques, there is no need for any system performance model, and these can be applied to coordinate various automatic system managers of the system.

Utility-based approaches: These approaches require a performance model and a mechanism to correlate cost with a performance at the user level.

Market-oriented approaches: Resource management based on market-oriented architecture has limited support provided by cloud computing technologies [2]. These approaches need not require any performance model for the system. The market-related metrics for maximizing the welfare is the sum of the service provider, and consumer surplus cannot be monitored in the real environment.

3 A General Model for Resource Management Mechanism

The infrastructure of cloud computing consists of three layers: application layer, platform layer, and infrastructure layer [14]. The application layer can be used by the users for sending request for their services and for receiving the outcomes. The platform layer provides an environment for application creation and to deploy them. On the other hand, the infrastructure layer provides the consumer with a set of virtualized resources for their services that are present within the datacenter as shown in Fig. 1.

The utility-based approach applied to such infrastructure uses the utility function to relate the cost to provide a service with the benefits of service [9]. For example, revenue to provide service could be a benefit, and power consumption could be considered as a cost. In the case of web services, SLAs specify the benefits along with penalties associated with QoS requirements. In SLAs-based policies, each user request for execution in the cloud is directed to the nearby data center by the service provider. The submitted request is interpreted by the SLA monitor to determine the QoS requirements shown in Fig. 2. Besides that, it also monitors the process of the submitted request.

These requests are processed on networked physical servers possessing a unique identity. Each physical server contained different virtual machines that share different computing resources of the physical servers with the help of hypervisor. These VMs contained within the sever have the same hardware and software capabilities as that of a physical server. The user requests are directed to these VMs within the server for processing with the help of the scheduler if different resources are present there to compete based on a deadline [15]. After completion of the task,

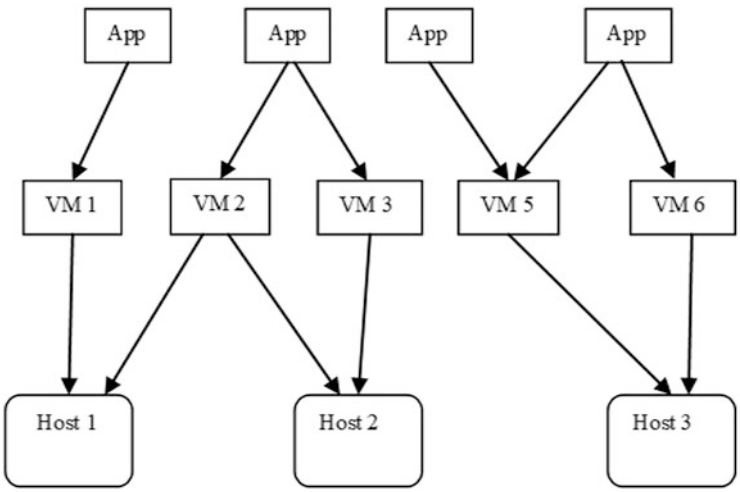

Fig. 1 A layered representation of cloud computing infrastructure

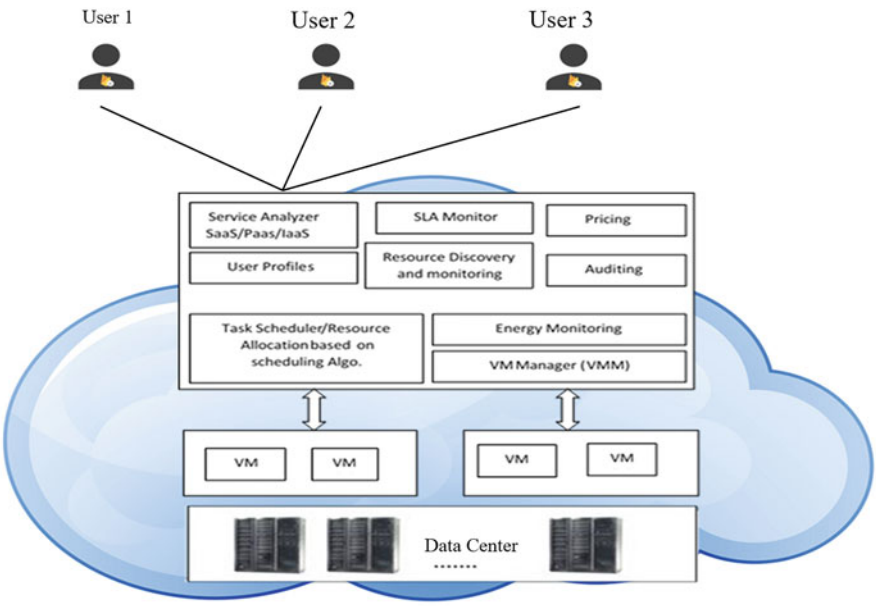

Fig. 2 A general model for resource management in cloud computing

these resources are released and available for the creation of new VMs for serving new user requests.

Figure 2 represents the different phases of management of resources that include resource provisioning, resource scheduling, and resource monitoring. Based on the user request and QoS parameters, adequate resources are identified by the consumer of the cloud for the provisioning of resources. The process includes the discovery and selection of resources based on user requests. For the discovery phase, the cloud consumer connects with a cloud portal for submission of workload after authenticating their QoS requirements. This information related to all the resources in the resource pool is collated at one place known as Resource Information Center(RIC), and the required information is retrieved based on workload specifications [16]. Resource provisioning agent (RPA) uses the workload and information retrieved by RIC for the availability of resources. After checking, the resource availability workloads are mapped to the appropriate resource by using resource scheduler based on the scheduling algorithm. Thus, the process of finding the available resources form a pool of resources is known as resource detection. Also, the process of selection of the most appropriate resource from the list generated during resource detection is known as resource selection. Resource monitoring is done to achieve better optimization results based on QoS [17].

3.1 Role of Essential Factors in Resource Management

The following factors affect resource management decisions.

Different types of constraints: Constraints can be from both sides, that is, the consumer of cloud services and the service provider.

- Constraints the consumer side may be related to a deadline or any budgetary control.
- Constraints can be enforced from the cloud service provider side regarding maximizing their resource usage to achieve maximum benefits.

Optimization criterion: This is different from the constraints, as constraints are based on the estimated values such as the number of available resources in cloud infrastructure. Besides, in this case of optimization criteria, the limits are specified by using the words minimum or maximum such as optimization criterion from the user side can be that task should be completed in a minimum amount of time. An objective function is used to express the optimization criterion, which helps in comparing the computed results with other tested results in the same field.

The resource scheduler's main objective is to get an optimal solution for the formulated objective function based on the optimization criteria prescribed by consumers or service providers in the cloud environs.

Quality of service (QoS): In cloud computing, QoS is used to represent the degree to which a set of inherent characteristics meet the requirements. These QoS requirements rely upon various performance metrics that are responsible for providing the security, scalability, and reliability offered by an application and by the infrastructure or platform that hosts it [18]. Invoking a QoS mechanism in the cloud computing environment allows the consumer to specify the requests, such as performance, advanced reservation of resources, completion before a given deadline. Such requests are formally established between the consumer and the service provider of cloud in a document called service-level agreement (SLA), which is a formal description of the guaranteed service.

4 Taxonomy of Resource Management(RM) Policies

Resource management is an important aspect of cloud computing, as it directly affects the performance and cost in the cloud environment. An inefficient approach adversely affects the functionality of the system and has a negative impact on performance. A cloud environment is based on virtualization that invokes the sharing of computing resources between multiple VMs. These VMs are easily manageable but prone to performance-related issues that bring various challenges in the field. These challenges are related to balancing the load on the system, energy optimization, SLA violations, and limiting the workload based on admission control. In extant various research that have been done in the field, and some which

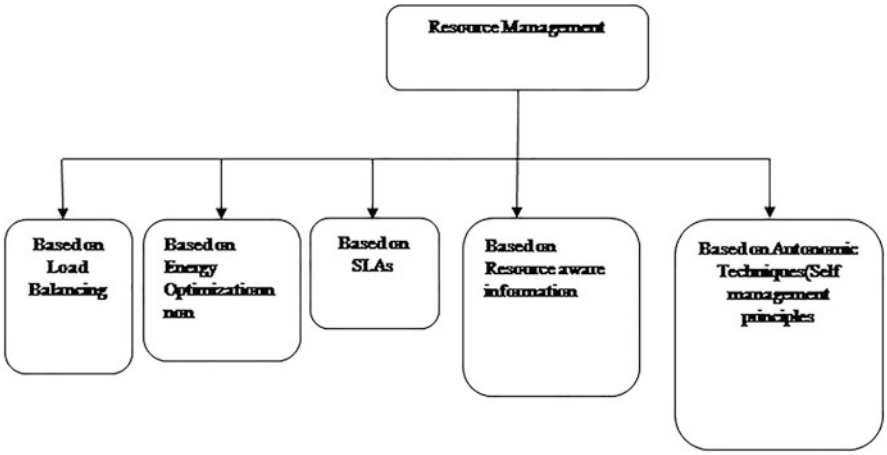

Fig. 3 Taxonomy of resource management policies

are persistently growing to overcome these challenges, there is a need to upgrade the literature to cover new aspects and to evaluate and upgrade the existing research techniques. Taxonomy has been designed to represent the related challenges in the field.

4.1 Related Surveys

Singh and Channa provide a taxonomy on resource management and define that resource management constitutes of three functions that include resource provisioning, resource scheduling, and resource allocation [16]. Saad Mustafa et al. provide a taxonomy that covers various performance metrics and research challenges in the field [19]. Jennings and Stadler also explain various resource types, metrics, the scope of resource management, and resource management functions in their study on resource management [20]. Swapnil M. Parikh et al. also provide a classification of RM techniques. The survey includes various resource types and a comparison of various RM techniques [21]. Based on the aforementioned survey of papers, challenges in the field of taxonomy has been designed for RM (Fig. 3).

4.2 RM Techniques Based on Load Balancing

To balance the load among available resources in the system is desirable in the cloud environment, as it ensures the stability and reliability of the system. It is related to balancing the load between the physical machines. Load balancing algorithms can

be integrated at two levels, at the application level and other at the VM level. At application level, the tasks can be migrated to balance the load and at VM level, the migration of virtual machines is done from one host server to another to balance the load among the physical machines [22]. Several research techniques have been devised on this challenging issue; moreover, there is a need to propose some new methods for sharing of workloads, as these techniques are multi-objective if used efficiently.

In 2015, Babu and Samuel [23] devised a technique to balance the load based on the foraging nature of honey bees. In their proposed technique, the task with the lowest priority has the higher chances of being migrated from an overwhelmed virtual machine(VM) to the underwhelm virtual machine. Their algorithm relies on the priority of tasks in a waiting queue for VM. Simulated results demonstrate tasks that are limited in number makespan reduces, and there is a reduction in the total number of migrations needed for operation, which shows that the proposed algorithm has low scalability.

In 2013, Babu and Krishna [24] proposed an algorithm for load balancing inspired by the foraging nature of honey bee that has an objective to achieve load balancing across VMs to have a high degree of throughput and reduction in waiting period of tasks in system queue by considering the priority of tasks. Tasks are migrated from an overload virtual machine to the underloaded virtual machine to reduce the makespan of the system. Tasks that are to be scheduled are non-preemptive and independent. When these tasks are migrated, they act as honey bees and the underloaded virtual machine acts as a food source for honey bees. The migrated task to the underloaded virtual machine globally updates the load information for deciding on the deployment of tasks for further processing. This exchange in the information process relates to the waggle dance of honey bees. The experimental results show that their approach led to a reduction in makespan for a given task set and an increase in throughput.

In 2011 Kokilvani et al. [25] proposed a Min–Min and Max–Min algorithm. At the first stage, Min–Min algorithm is applied to tasks on different servers, and it identifies the task and resources with minimum execution time. After this task allocation process, tasks are removed from the unassigned list of tasks and remaining tasks with calculated execution time are updated on the particular server that hosts it. Besides that, a Max–Min approach is applied in the same way, but it selects a task with maximum execution time.

In 2016, Mittal and Katal [26] have introduced an algorithm based on task scheduling optimization. The algorithm uses the total execution time taken by individual resources to process all tasks. It is calculated by determining the total execution time overall available resources to process all tasks. Afterward, the algorithm calculates the two optimized values one for the fastest resource and one for the slowest resource based on calculation performed over the execution time of the fastest resource and slowest resource. They have applied the algorithms Max–Min, and Min–Min based on these optimized values. This algorithm aims to reduce the makespan.

Although load balancing approaches are beneficial for increasing the performance and balancing the load on the system, one must be careful during the implementation of a load balancing approach, as it can increase the traffic on the network during task migration or VM migration.

4.3 RM Techniques Based on Energy Optimization

Minimization of energy consumption is another important parameter that is a recent research challenge in the cloud computing environment. The complexity of the cloud computing ecosystem is persistently increasing with the proliferation of cloud services and an increase in the number of heterogeneous resources [11]. With this growth, energy consumption of service providers is also increasing that further raises various challenges related to CO_2 emission. Several research techniques that have been devised for minimization of energy consumption are presented here.

In 2019, Moges et al. [27] devised an energy-efficient algorithm that is based on Modified Best Fit Decreasing (MBFD) technique used for VM placement. Although MBFD is a well-renowned technique best known as the VM placement algorithm of OpenStack Neat, it increases the SLA violations and less energy efficient that led to VM migrations. To overcome the aforementioned issue, Moges proposed an approach based on bin-packing heuristics. Also, another bin-packing approach has been introduced known as Medium-Fit that is used to reduce SLA violations. Experimental results demonstrate the reduction in energy consumption and SLA violation.

In 2015, Carli et al. [28] devised a technique based on bin-packing problem to model the energy consumption of private clouds. The algorithm has an objective to optimize energy consumption by balancing the load. It is an effort to model the packing cost that is proportional to energy utilization. Two algorithms have been proposed one offline and another for online mode. Experimental results show that packing cost decreases with an increase in the number of bins when implemented in the offline mode. The proposed algorithm may also reduce the number of bins needed for packing the given items by doing the fragmentation of items under some constraints.

In 2013, Gao et al. [29] devised a multicriteria VM placement algorithm by using an ant colony optimization technique. It is used to achieve a Pareto efficient solution that simultaneously minimizes power consumption and resource wastage. In this approach, all parameters are initialized in the beginning, and pheromone trails are initialized to 0. The process of execution starts with a VM request received by an ant which is to be mapped to the proper host. Mapping is based on the concentration of pheromones which act as a heuristic for mapping to VM host. The pheromones are updated at the local level after every assignment. After completion of the pheromone updating process by ants, the pheromones are updated at a global level. Simulation results show the optimum resource utilization by all resources and reduction in energy utilization in comparison to multi-objective grouping genetic algorithm.

Both load balancing and energy optimization are interrelated concepts, and an efficiently balanced system always leads to energy efficiency by using server consolidation and switching off the underloaded server.

4.4 RM Techniques Based on Cost

In a cloud computing environment, the provisioning of resources can be broadly classified into two ways, based on demand and reservation plans. Both these scenarios of resource provisioning directly affect the cost. The cost for the on-demand plan is charged based on pay-per-usage basis and the cost for the reservation plan is charged one time. The cost for the on-demand plan remains high, as the consumer has a facility to dynamically provision resources that are needed to adapt in an environment of unpredictable demands. Various resource management techniques are devised for reducing the cost and maximizing the profit during resource provisioning.

In 2010, Liu et al. [6] proposed a scheduling algorithm by negotiating time and cost for the instance intensive workflows. The results of the simulation demonstrate that mean completion cost and mean completion time is reduced by 15% and 20%, respectively.

In 2012, Chaisiri et al. [7] proposed a resource provisioning algorithm to overcome the problem of advanced reservation that arises due to uncertainty in future demands and resource cost by cloud supplier. The algorithm considers the cost and demand uncertainty during the resource provisioning. Numerical analysis of results proves that total cost can be reduced in cloud environs.

In 2014, Zho et al. [8] proposed an algorithm to minimize the completion time cost for processing all the tasks. A PSO-based algorithm has been used by considering the heterogeneous resources in the cloud environment. A fitness function has been used to determine resource usage and processing time cost.

4.5 RM Techniques Based on SLA Violations

SLAs are a legalized document that represents the consented benefits given to the consumer of services. Moreover, it also includes the penalties enforced in case of service violations. Researchers have presented various approaches using SLA-aware scheduling algorithms. It guarantees the quality of services(QoS) given to the user of the cloud.

In 2019, Yong Wang et al. [30] proposed an algorithm for efficiently using cloud storage by using an SLA-aware approach. The proposed approach is based on the Cinnder weight algorithm in OpenStack. The algorithm uses the I/O throughput

and utilization of space in resources like information for efficient cloud storage. The proposed approach has one drawback that it cannot arrange the cloud nodes dynamically.

In 2014, Alrokayan et al. [31] proposed a task scheduling algorithm particularly used for Big Data analytic jobs. It is an SLA-aware policy guaranteed to meet the services within budget and deadline. A compare-cost-aware algorithm has been used for scheduling the analytic jobs by provisioning the resources.

In 2012, Ardagna et al. [32] proposed an algorithm based on a capacity allocation that can be operated on distributed sites to control the fluctuating workload. It is an SLA-aware technique that guarantees the service provided within constraints specified in SLA. The algorithm assures the capability to regulate different controllers distributed over different cloud sites. It also uses the mechanism to redirect the load over distributed sites when load fluctuation is there.

4.6 RM Techniques Based on Resource-Aware Scheduling

The algorithms used to enhance the usage of resources in an optimal way are called resource-aware algorithms. These resources may include computing power, energy, network bandwidth, etc. Operations in these algorithms are based on the monitoring and observing parameters that belong to resources to determine the current status, such as resource usage percentile, degree of imbalance in resource capacity, consumption of energy, network load to analyze the need, and availability of resources. Bin packing and dynamic clustering to balance workload are some techniques that can be used for load balancing.

In 2014, Ramezani et al. [33] proposed a load balancing technique based on Particle swarm optimization. This technique was used to overcome the problems that occurred during live migration of VM for balancing load in the cloud computing system. Although during live migration of VM downtime of this is insignificant, it occupies significant space in memory and takes time. It also has the risk of losing recent consumer activities. To deal with such issues, algorithms used a task migration approach using particle swarm optimization. The proposed algorithm deals with the migration of tasks instead of VM migration to another host that reduces the network load. This algorithm first used to find a set of tasks that is to be migrated from overloaded VMs to the underloaded VMs. The proposed algorithm uses a particle swarm optimization technique for task migration from overload VM to another appropriate VM by keeping the value of the objective function minimum to reduce the migration time and execution time. Experimental results showed that there is an improvement in downtime and QoS.

In 2013, Tziritas et al. [34] proposed an algorithm to solve the network load and energy consumption in the cloud environment. The algorithm is based on a lazy approach for doing the VM migration, only the most beneficial VM migrations are

done. The algorithm initiates by using Low Perturbation Bin Packing Algorithm (LPBP) and maintains a list in descending order based on energy consumptions. All the VMs or a set of VMs of a server present in the top of the list are migrated to an underloaded server. No migration is performed if the server cannot be fully offloaded.

In 2016, Sheikhalishahi et al. [35] devised a resource scheduling technique based on capacity allocation using the bin packing approach. The algorithm estimates the job score based on the requirement of resources for the particular job and availability of the resources on the available physical machines. This calculated job score determines the host's feasibility to schedule the job at a particular scheduling time. Results of simulation show that the aforementioned approach is beneficial in improving performance and resource utilization.

4.7 Autonomic Resource Management

Autonomic RM techniques are based on self-management principles that include self-management properties, which include self-healing, self-protecting, self-optimizing, and self-configuring [37].

In 2019, Gill and Buyya [36] proposed a resource provisioning mechanism that abides the QoS based on SLA. The resource provisioning framework named SCOOTER is an autonomic technique that automatically schedules resources to observe service behavior, and it satisfies the QoS requirements by adjusting the system dynamically.

In 2014, Sah and Joshi [38] proposed a technique based on self-management principles for distributing the workload dynamically named AVM(Autonomic Workload Manager) to solve the problem for a flat separable queuing network model. A Distributed Provisioning and Scaling Decision Making System (DPSMS) has been used for workload distribution to available resources to satisfy the QoS requirements. The proposed technique categorizes the resources into two ways, coarse-grained and fine-grained. AVM works in three phases: (1) In the first phase allocation of resources is done to the incoming workloads; (2) execution time is minimized; and (3) execution time is checked. (AVM demands more resources if execution time exceeds the time and budget).

In 2015, Sheikhalishahi et al. [39] proposed a contention-aware resource scheduling that is an autonomic technique. The algorithm has been to reduce resource contention for a distributed system and named it Autonomic Resource Contention Scheduling (ARCS). The ARCS works in four modules. (1) Jobs are admitted and queued based on admission control. (2) Backfilling algorithm is used for job scheduling. (3) The information about the scheduler can be retrieved from the information service. (4) Resources are allocated to jobs.

5 Performance Metrics to Evaluate RM Techniques

The implemented policies for managing resources are evaluated on the scale of performance metrics. To determine the performance corresponding to the respective parameter, the observed value is compared with the expected result. It is an important step to evaluate the performance of the implemented technique, as it ensures the stability of the new techniques in the environment. Table 1 represents different performance metrics used by RM techniques studied in the literature. The following are some of the parameters used by the various researchers in the field to evaluate the performance.

Resource utilization: Resource utilization is one of the important metrics that directly affect the profits of the cloud service provider. It is directly related to energy consumption; a system with high resource utilization uses the energy optimally by minimizing the use of resources. The overall utilization of resources can be evaluated by the following equation [2]

Table 1 RM techniques with different performance metrics

Paper refs.	Performance metrics considered	Platform used
Babu and Simuel [23]	Makespan, VM migration	CloudSim
Babu and Krishna [24]	Throughput, Makespan	CloudSim
Kokilvani et al. [25]	Resource utilization, Makespan	C++
Mittal and Katal [26]	Makespan	Java 7
Moges et al. [27]	SLA violations, energy consumption	Openstack neat, CloudSim
Carli et al. [28]	Energy consumption	Real platform
Gao et al. [29]	Resource utilization, energy Consomption	□
Yong wang et al. [30]	SLA violations	Virtual cinder block storage (Rackspae)
Alrokayan et al. [31]	SLA violations	CloudSim
Ardagna et al. [32]	Cost, SLA violations	VMWare virtual machine based on Ubuntu 9.10
Ramezani et al. [33]	Network load, VM migration	CloudSim
Tziritas et al. [34]	Network load, energy consumption	□
Sheikhalishahi et al. [35]	Resource utilization	SDSC blue horizon traces used as workload, simulated environment
Gill and Buyya [36]	Execution time, execution cost, resource utilization, energy consumption	CloudSim, JADE platform
Sah and Joshi [38]	Execution time	CloudSim
Sheikhalishahi et al. [39]	Completion time, resource contention	Haizea, SDSC blue horizon traces

$$U_x\,(F,t) = \sum_{k=1}^{v} f_{xk} * \frac{Req - CPU_x(t)}{CPU_x}$$

Here, $U_x(F,t)$ represents the resource utilization of server S_x at a particular time and F indicates the placements of VMs on the same. Also, the notation f_{xk} represents the placement of a VM V_k hosted on the server or not. The value of f_{xk} is 1 if it is hosted on the server S_x or 0 otherwise. CPU_xRepresents the total computing power of the server S_x and $Req - CPU_x\ (t)$ represents the extent of power required by VM V_k at a particular time t.

Response time: A minimum response time is always desirable by a cloud customer for the execution of a request. It is the time lag between the submission of a request by a customer and its outcome after processing the request. To keep it the minimum resources should be balanced, as requests migrated to an overloaded VM may increase the response time. It can be calculated by the following formula:

$$\text{Response Time}(t) = F(t)\text{–Sub}(t)$$

Here, $F(t)$ denotes the finish time of processing a request, and Sub(t) represents the submission time of a request.

Makespan: User request is decomposed into task units for processing. So, makespan is the time when the latest task finishes related to a user request. It is independent of any particular execution order of tasks [4]. It can be defined as:

$$\min_{s_i \in \text{Sched}} \left\{ F_j \right\}$$

Here F_j notation represents the time when the task j finalizes, set of all schedules represented by Sched and jobs represent the set of all jobs that are to be scheduled.

Throughput: The throughput is another performance metric to determine the capability of the system to execute tasks in terms of time. It determines the number of tasks executed by a system in a unit time. A system with high throughput means that tasks are executed in less time and service provided to the customer with a rapid response [25]. The following formula is used to calculate the throughput of the system (if all the tasks executed of the same length).

$$\text{THP} = \frac{(\text{Task lenght} * \text{number of task})}{\text{Response Time}}$$

VM migration time: Researchers devised various RM techniques to increase performance in a cloud environment. Moreover, they use the VM migration technique for several reasons, such as to balance the load, to optimize the energy consumption, to reduce network load and other communication overheads that may arise in between two or more VMs during processing interdependent tasks. For an efficient RM technique, it should be minimized. The following formula can be used to calculate VM migration [3].

$$T_{mj} = \frac{M_j}{B_j}$$

Here, T_{mj} denotes the migration time, M_j represents the memory used by the migrated VM_j and B_j denotes the bandwidth.

Number of VM migrations: This is another important parameter to consider while evaluating an RM technique. The number of migrations should be minimum to reduce the network load and downtime of a virtual machine. The following formula can be used to calculate the number of VMs migration [19].

$$\text{migrations}\,(F, t1, t2) = \sum_{x=1}^{v} \int_{t_1}^{t_2} \text{Mig}_x(F)$$

Here, F denotes the current placement of VMs, and the number of migrations for server S_x is denoted by Mig_x in the time interval for placement F.

Energy consumption: In cloud computing system, energy consumption [49–51] is the total amount of energy absorbed by all the Information and Communication Technology (ICT) devices interconnected in the system. These include personal terminals, network components, and local servers. Proper resource management lowers energy consumption. In a cloud environment, each virtual resource has two states, idle and active. An idle state of virtual resource consumes 60% of the energy consumption of the active state that virtual resource [22]. The total energy consumption is the sum of the energy consumed during the active and idle state.

6 Research Issues and Challenges in RM

Several techniques have been devised by the researchers for resource management but still, there are some issues and challenges in this field need to provide a solution. These are the following.

Virtual machine migration: Virtualization [40, 41] is known as an enabling technology that allows a physical machine to emulate its behavior into one or more virtual machines. During resource management, there is a need to migrate a virtual machine image from one host server to another. Challenge is during the dynamically distributing the load for balancing the resources by migrating the virtual machine image from one host to another.

Automated service provisioning: Elasticity is the main characteristic of cloud computing liable for automatically provisioning and releasing of resources. In a dynamic cloud, environment the latest information when available at best is obsolete [9]. It is hard to manage the resources in such a dynamic environment. The RM techniques are needed to be devised on self-management principles to cope up with persistently changing demands.

Managing geographically distributed data centers: With the proliferation of services in the cloud environment resources are increasing in abundance. The communications may lead to a high network load during large operations. RM techniques [42–45] need to be designed over large, distributed networks to handle the communication delay. Some intelligent techniques for resource allocation need to be designed like placing the subtasks that are interdependent on the same cluster.

Managing information: It is difficult to collect information about resources in such a huge network. By collecting the information from various distributed servers could affect the performance due to network load and communication overheads. RM techniques need to be designed for collecting and analyzing the information over distributed cluster managers rather than using only a central manager to reduce the communication overheads. For proper management, the central manager needs to decide the host cluster basis information.

Algorithm complexity: A complex RM technique may lead to complex implementing process that requires more computations and more information which may drop the performance. RM techniques need to be devised having less complexity in terms of several operations for execution providing the maximum throughput and optimum resource utilization.

RM techniques to avoid security threats: Users of competitor organizations could also share the resources in the cloud environment. Certain cases may occur with the aforementioned issue in which users can use data or algorithms of other organizations. The lack of robustness in RM led to a data breach in the information; therefore, some RM technique is required to overcome the problem. There is some information security model like Nash and Brewer in which users are separated into different categories based on their conflict of interest, in this security is ensured by permitting the users for sharing the server with the same conflict of interest.

7 Conclusion

This chapter presents a review of cloud resource management techniques within Industry 4.0. In a cloud environment, resource consumption is based on the request that enables industry 4.0. The chapter discusses the need for RM, different policies, and mechanisms in RM. It also discusses different issues and challenges in the field. After an extensive study on RM techniques, it can be concluded these techniques have one common goal that is to achieve the performance in terms of QoS. Some of these techniques are interrelated such as RM based on balancing the load and energy-efficient management of resources, so one should be very attentive while choosing these QoS metrics as one can adversely affect the other.

It is also concluded that these RM techniques at best can be implemented at a local level; to implement these policies at a large scale most updated information is required at the global state of the system. To determine such a piece of information over large distributed networks constitute thousands of servers is not feasible as the

state of the system changes very rapidly. So, it is expected that such a complex system can work better on self-management principles with the proactive security mechanism to avoid unauthenticated logging and auditing in the system.

References

1. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2017). Privacy and security of cloud-based internet of things (IoT). 2017 3rd international conference on computational intelligence and networks (CINE) (pp. 40–45).
2. Sharma, A., Choudhury, T., & Kumar, P. (2018). Health monitoring & management using IoT devices in a cloud based framework. 2018 international conference on advances in computing and communication engineering (ICACCE) (pp. 219–224).
3. Mittal, A., Khan, F. S., Kumar, P., & Choudhury, T. (2018). Cloud based intelligent attendance system through video streaming. Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358587.
4. Kumra, S., Choudhury, T., Nhu, N. G., & Nalwa, T. (2018). Challenges faced by cloud computing. Proceedings of the 2017 3rd international conference on applied and theoretical computing and communication technology, ICATccT 2017. https://doi.org/10.1109/ICATCCT.2017.8389105.
5. Dinote, A., Sharma, D. P., Gure, A. T., Singh, B. K., & Choudhury, T. (2020). Medication processes automation using unified green computing and communication model. *Journal of Green Engineering, 10*(9), 5763–5778.
6. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *(IJACSA) International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
7. Tomar, R., Khanna, A., Bansal, A., & Fore, V. (2018). An architectural view towards autonomic cloud computing. In Data engineering and intelligent computing.
8. Choudhary, V., Kacker, S., Choudhury, T., & Vashisht, V. (2012). An approach to improve task scheduling in a decentralized cloud computing environment. *International Journal of Computer Technology and Applications, 3*(1), 312–316.
9. Marinescu, D. C. (2017). *Cloud computing: Theory and practice*. Burlington, MA: Morgan Kaufmann.
10. Singh, S., & Chana, I. (2015). QoS-aware autonomic resource management in cloud computing: a systematic review. *ACM Computing Surveys (CSUR), 48*(3), 1–46.
11. Marinescu, D. C., Paya, A., Morrison, J. P., & Olariu, S. (2017). An approach for scaling cloud resource management. *Cluster Computing, 20*(1), 909–924.
12. Singh, S., & Chana, I. (2015). QoS-aware autonomic resource management in cloud computing: A systematic review. *ACM Computing Surveys (CSUR), 48*(3), 1–46.
13. Sosinsky, B. (2010). *Cloud computing bible* (Vol. 762). Hoboken: John Wiley & Sons.
14. Haidri, R. A., Katti, C. P., & Saxena, P. C. (2014). A load balancing strategy for Cloud Computing environment. In The proceedings of 2014 international conference on in signal propagation and computer technology (ICSPCT) (pp. 636–641). IEEE.
15. Sharma, M., Kumar, R., & Jain, A. (2020). Load balancing in cloud computing environment: A broad perspective. In Proceedings of ICSAD international conference on sentimental analysis and deep learning. Springer. India.
16. Singh, S., & Chana, I. (2016). Cloud resource provisioning: survey, status and future research directions. *Knowledge and Information Systems, 49*(3), 1005–1069.

17. Gonzalez, N. M., de Brito Carvalho, T. C. M., & Miers, C. C. (2017). Cloud resource management: Towards efficient execution of large-scale scientific applications and workflows on complex infrastructures. *Journal of Cloud Computing, 6*(1).
18. Sharma, M., Kumar, R., & Jain, A. (2020). A proficient approach for load balancing in cloud computing-join minimum loaded queue: Join minimum loaded queue. *International Journal of Information System Modeling and Design (IJISMD), 11*(1), 12–36.
19. Mustafa, S., Nazir, B., Hayat, A., & Madani, S. A. (2015). Resource management in cloud computing: Taxonomy, prospects, and challenges. *Computers and Electrical Engineering, 47*, 186–203.
20. Jennings, B., & Stadler, R. (2015). Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management, 23*(3), 567–619.
21. Parikh, S. M., Patel, N. M., & Prajapati, H. B. (2017). Resource management in cloud computing: Classification and taxonomy. arXiv preprint arXiv:1703.00374.
22. Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: A big picture. *Journal of King Saud University Computer and Information Sciences, 32*, 149–158.
23. Babu, K. R., & Samuel, P. (2015). Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. *Innovations in Bio-Inspired Computing and Applications, 424*, 67–78.
24. Dinesh Babu, L. D., & Krishna, P. V. (2013). Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing, 13*(5), 2292–2303.
25. Xhafa, F., & Abraham, A. (2008). Meta-heuristics for grid scheduling problems. In *Metaheuristics for scheduling in distributed computing environments* (pp. 1–37). Berlin: Springer.
26. Rana, K., & Zandu, V. (2016). Resource-aware load balancing scheme using multi-objective optimization in cloud computing. *International Journal of Computer Science, 8*(9), 345–353.
27. Mittal, S., & Katal, A. (2016). An optimized task scheduling algorithm in cloud computing. In The proceedings of 6th international conference on advanced computing (IACC), Bhimavaram, India (pp. 197–202).
28. Moges, F. F., & Abebe, S. L. (2019). Energy-aware VM placement algorithms for the OpenStack neat consolidation framework. *Journal of Cloud Computing, 8*(1).
29. Carli, T., Henriot, S., Cohen, J., & Tomasik, J. (2016). A packing problem approach to EnergyAware load distribution in clouds. *Sustainable Computing: Informatics and System, 9*, 20–32.
30. Gao, Y., Guan, H., Qi, Z., Hou, Y., & Liu, L. (2013). A multi-objective ant Colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences, 79*(8), 1230–1242.
31. Wang, Y., Tao, X., Zhao, F., Tian, B., & Sai, A. M. V. V. (2020). SLA-aware resource scheduling algorithm for cloud storage. *EURASIP Journal on Wireless Communications and Networking, 1*, 1–10(2019).
32. Alrokayan, M., Dastjerdi, A. V., & Buyya, R. (2019). Sla-aware provisioning and scheduling of cloud resources for big data analytics. In 2014 IEEE international conference on cloud computing in emerging markets (CCEM) (pp. 1–8). IEEE.
33. Ardagna, D., Casolari, S., Colajanni, M., & Panicucci, B. (2012). Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems. *Journal of Parallel and Distributed Computing, 72*(6), 796–808.
34. Ramezani, F., Lu, J., & Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization. *International Journal of Parallel Programming, 42*(5), 739–754.
35. Tziritas, N., Xu, C. Z., Loukopoulos, T., Khan, S. U., & Yu, Z. (2012). Application-aware workload consolidation to minimize both energy consumption and network load in cloud environments. In 2013 42nd international conference on parallel processing (pp. 449–457). IEEE.
36. Sheikhalishahi, M., Wallace, R. M., Grandinetti, L., Vazquez-Poletti, J. L., & Guerriero, F. (2016). A multi-dimensional job scheduling. *Future Generation Computer Systems, 54*, 123–131.

37. Kokilavani, T., & Amalarethinam, D. G. (2011). Load balanced min-min algorithm for static meta-task scheduling in grid computing. *International Journal of Computer Applications, 20*(2), 43–49.
38. Gill, S. S., & Buyya, R. (2019). Resource provisioning based scheduling framework for execution of heterogeneous and clustered workloads in clouds: From fundamental to autonomic offering. *Journal of Grid Computing, 17*(3), 385–417.
39. Sah, S. K., & Joshi, S. R. (2014). Scalability of efficient and dynamic workload distribution in autonomic cloud computing. In 2014 international conference on issues and challenges in intelligent computing techniques (ICICT) (pp. 12–18). IEEE.
40. Agarwal, A., Venkatadri, M., & Pasricha, A. (2018, December). Autonomic cloud resource management. In 2018 fifth international conference on parallel, distributed and grid computing (PDGC) (pp. 138–143). IEEE https://doi.org/10.1109/PDGC.2018.8745977.
41. Dewangan, B. K., Agarwal, A., Pasricha, A., Choudhury, T., & Chandra Satapathy, S. (2020). Extensive review of cloud resource management techniques in industry 4.0: Issue and challenges. *Software: Practice and Experience*. https://doi.org/10.1002/spe.2810.
42. Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
43. Venkatadri, M., Agarwal, A., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science, 152*, 204–211. https://doi.org/10.1016/j.procs.2019.05.044.
44. Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2018). Resource scheduling in cloud: A comparative study. *International Journal of Computer Sciences and Engineering, 6*(8), 168–173.
45. Faruqui, N., Yousuf, M. A., Chakraborty, P., & Hossain, M. S. (2020). Innovative automation algorithm in micro-multinational data-entry industry. In Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325 LNICST, pp. 680–692). Springer. https://doi.org/10.1007/978-3-030-52856-0_54.

A Walkthrough in Live Migration Strategies for Energy-Aware Resource Management in Cloud

Neha Gupta, Kamali Gupta, Meenu Khurana, Deepali Gupta, Anurag Jain, and Bhupesh Kumar Dewangan

1 Introduction of Cloud Computing

Cloud computing is going to reshape the IT industry as a revolution very soon [1]. Cloud has reduced the cost of using computing resources to a greater extent that now people do not hesitate to shift on cloud for getting the resources on pay-per-use basis.

Five essential components of cloud computing are:

1. *On-demand self-service*: Consumers can get the resources on a click from the cloud service provider according to the requirement.
2. *Broad network access*: Users can access resources across the globe. Actual location of the resource does not affect its mode of accessibility and time duration between request and response.
3. *Resource pool*: There is a pool of computing resources with latest updates. Cloud service providers provide these resources to heterogeneous users without service interruption.

N. Gupta · K. Gupta (✉) · M. Khurana · D. Gupta
Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, India
e-mail: neha.gupta@chitkara.edu.in; kamali.singla@chitkara.edu.in; meenu.khurana@chitkara.edu.in; deepali.gupta@chitkara.edu.in

A. Jain
Department of Virtualization, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, India
e-mail: anurag.jain@ddn.upes.ac.in

B. K. Dewangan
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India
e-mail: b.dewangan@ddn.upes.ac.in.

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_16

4. *Rapid elasticity*: The request for more resources during runtime can be easily served in an automatic manner.
5. *Measured service*: Cloud service provider can measure the services used by users accurately, despite its multitenancy feature. Cloud customers need to pay only for what they use.

Deployment Model in cloud tells the access rules for users. There are four types of deployment models—Private, Community, Public, and Hybrid Cloud [2]. Service model tells about the description of services given by cloud providers. There are three types of service models—Infrastructure as a Service, Software as a Service, and Platform as a Service [3].

1.1 Challenges of Cloud Computing

Although the cloud service providers are providing enormous resources at a very cheap rate, still they are facing some challenges which degrade the performance of cloud system and increase the cost payable by customers.

(a) *Data management and resource allocation*: Managing data of unlimited customers with efficient resource allocation is the primary challenge of cloud computing. Resources can be network resources, processing resources, development tools, etc. Resource allocation not only minimizes power consumption by data centers but also reduces the number of active virtual machines on a physical machine.
(b) *Security and privacy*: User authentication and access control process are important points in security on cloud. There are various methods adopted by cloud to secure sensitive data like the user does not know the actual location of storage. Firewalls are installed on data centers to check the incoming information, but still new security techniques should be adopted by cloud in the future.
(c) *Load balancing*: As the number of customers for cloud is increasing at a very fast rate, overloading of physical machines can happen on cloud, so more algorithm should be designed to split the load from overloaded physical machines to underload physical machines.
(d) *Scalability and availability*: To fulfill dynamic changing requirements of the user, cloud should provide good scalability of resources without disturbing the work of the user. Cloud service provider needs to ensure all time availability of the resources.
(e) *Migration to cloud and compatibility*: Cloud is gaining interest of the customer in best possible way. When any user wants to migrate complete existing setup of system to cloud, a lot of compatibility issues arise. The mechanism of adaptation of standard existing IT components against cloud run time environments should be supported by cloud service provider.

(f) *Energy consumption*: Cloud data centers need to be functional all time to maintain data of their customers, so they are consuming large amount of energy 24/7. This energy requirement is so high that it becomes major challenge for cloud service providers as it is not only increases the cost of maintaining data centers but also affects the environment by continuously emitting carbon footprints.

According to the statistics, electricity consumption in data centers is 2% of the total energy consumption globally, and it can rise to 8% by 2030 [4]. Carbon emission by data centers is about 0.3% of total carbon emission.

Therefore, the research study embarks on energy management which can also be mitigated by adopting efficacious resource scheduling strategies; therefore, resource management, its elements, and scheduling models are discussed in Sect. 2. Section 3 explicates on energy management and live migration strategies in cloud computing. Section 4 has elucidated discussion points for future research work. Section 5 presents conclusion.

2 Overview of Resource Management

Cloud computing is famous among various organizations because of its pool of computing resources that are shared among unlimited number of users. Managing these resources is a major job of cloud service providers. The resource management with its large scope is divided into three parts, Resource Monitoring, Resource Allocation, and Resource Discovery, as shown in Fig. 1.

Resource monitoring [6] is a primary task that handles supervision of resources like network resources, computing resources, etc. It collects the data to help in taking decisions regarding feasible resource allocation. Besides, it saves the resource

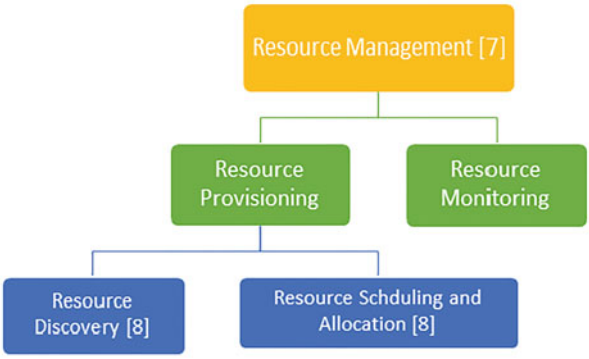

Fig. 1 Resource management taxonomy in cloud computing [5]

status in case of any failure. Monitoring of resources is important for optimizing the use of resources.

As mentioned in [7], resource discovery is the process to identify the physical machine that can efficiently provide VM with required resources. The role of resource scheduling is to identify the feasible physical machine from a pool of matched physical machines. After this decision of scheduling, all requests are given to the available resources and are divided into two categories based on requests by users: task allocation and resource allocation. Task allocation is performed at PaaS level and is assigning the submitted requests to the virtual machine instances. Resource allocation is performed at IaaS level and is a technique of allotting the available virtual machine resources to the requests submitted.

2.1 Resource Allocation Strategies: An Elemental Walkthrough

From cloud service provider's point of view, it is impractical to predict the dynamic and heterogenous demand for different level of users. Cloud users aspire to obtain the bequeathed service request before deadline with nominal charges [8]. Hence, efficient resource allocation techniques are required that have the ability to establish a match between both perspectives. The system should be able to deal with finite resources, heterogeneous demands by heterogeneous users, and geographical distribution of customers and resources. Some of the existing strategies for resource allocation in a Cloud computing are mentioned in Fig. 2.

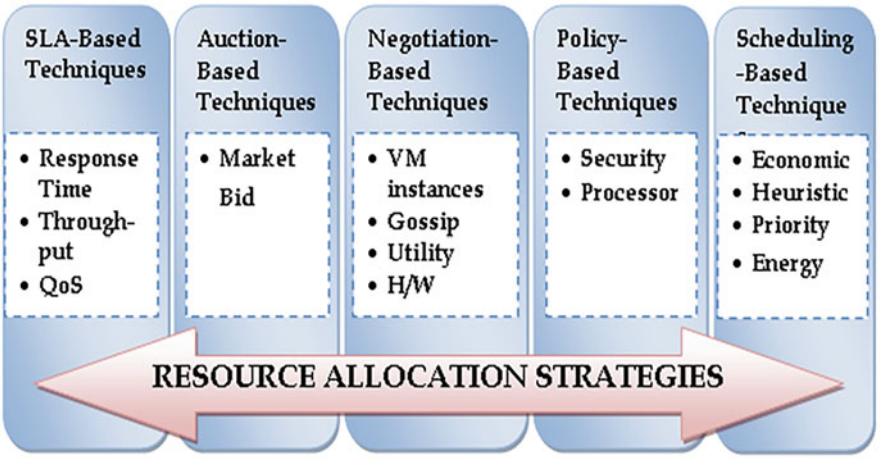

Fig. 2 Elements of resource allocation strategies [5]

SLA-based cloud scheduling: Resource allocation ensures to maintain SLA agreement while submission of user's request to physical machines. The parameters important for users are QoS, response time, and deadline, but the parameters for providers are proper resource utilization.

Auction-based technique: Here allocation of resources is a market-based strategy that trade in resources after the auctions on the basis of their bids and resource availability.

Negotiation-based technique: This technique is mainly divided into four categories, namely, VM instances, gossip, utility, and hardware resource dependency. These techniques complete the services of customer by negotiating on type of request made by them.

Policy-based approach: Resource allocation is dependent on the requests made by the users. Important parameter in some requests is security while in other requests it is processor (storage or computing services).

Scheduling-based approach: Resource allocation is dependent on scheduling decisions and is explained in the next section.

2.2 Existing Resource Scheduling Models in Cloud Computing

The scheduling techniques are classified as economic schedulers, heuristic models, priority schedulers, and energy-aware schedulers [9] as shown in Fig. 3.

Economic schedulers use auction-based method to regulate the matching of demand and supply of computing resources. This scheme is beneficial for user as well as cloud service provider.

Heuristic-based scheduling is achieved by comparing all feasible solutions with threshold solution to deduce whether the objective has been fulfilled. It takes lesser time.

Priority schedulers checks the priority type (fixed or dynamic) to map the user's request and available resource.

Energy-aware schedulers use bin-packing techniques to match the request with resources. Live migration of VM is conducted to pack VM instances on minimum number of physical servers (bins).

Round-robin schedulers allot CPU to the user's request for a time quantum and if request does not complete in the given time slot, it has to wait for next chance for execution.

Considering the hazardous impact of enormous energy release due to uneffi-cacious resource scheduling/increasing traffic on cloud data centers, it becomes imperative to discuss the strategies that can contribute to reducing energy consumption on cloud data centers. Therefore, this research study presents extensive survey of existing strategies in this direction in the next section.

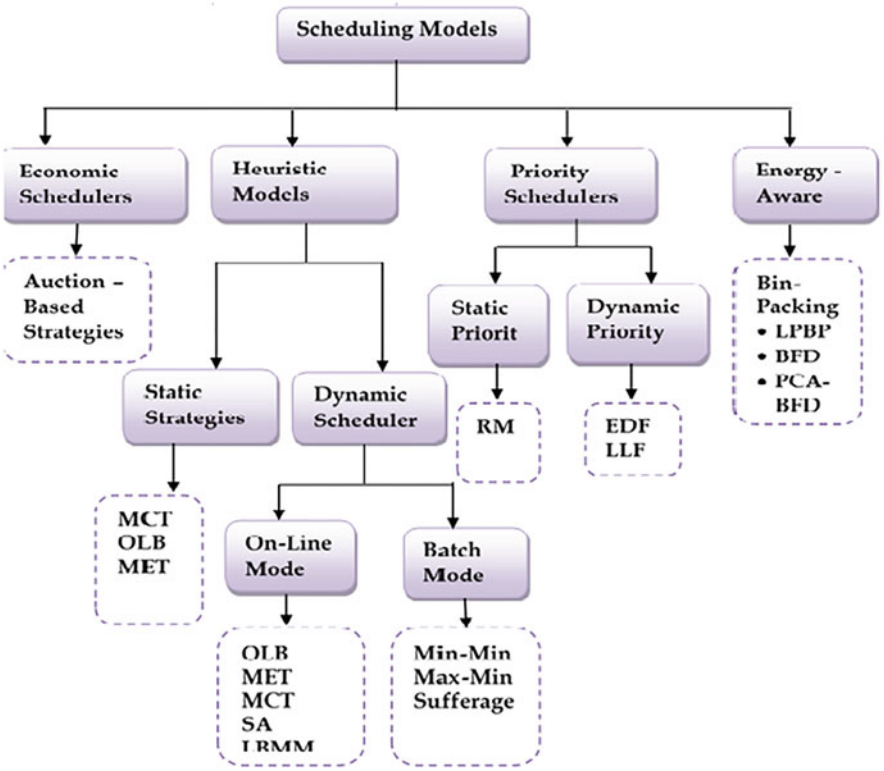

Fig. 3 Taxonomy of resource scheduling [5]

3 Energy Management in Cloud

3.1 Motivation of Research

As we know cloud computing is one of the buzzwords, nowadays, in the current scenario of digitization. Its benefits and challenges are going to expand in the near future which should be worked upon. Hence, cloud computing is chosen as research domain to contribute to the area.

Cloud service providers are facing a lot of challenges in providing services such as load balancing, resource management, energy consumption, etc. Among all challenges, energy consumption has the maximum weightage as it not only increases the cost on customers but also effecting the environment by increasing the carbon footprints.

Number of algorithms have been proposed to reduce the energy of data centers, but effective and long-term solution is yet to come. The research study is envisaged to help researchers in coming up with the solution, as a deep survey has been carried out.

3.2 Existing Energy Management Scheduling Strategies

Increasing data on cloud is increasing energy consumption exponentially. Many researchers are working in this direction to come up with a solution that can minimize the energy consumption in cloud data centers. The existing scheduling strategies are discussed briefly in Table 1.

Cloud computing has been proved as a boon for companies dealing with huge amount of data, as it provides cheap resources with high reliability and security but still is facing many challenges that affect the performance of cloud service providers. Among all challenges, resource management and energy consumption are major issues. They depend on number of working servers at a time, so by migrating VM, maximum servers can be made idle to decrease total energy requirement of the system.

Migration of VM can be live or non-Live migration. Live migration is migration of VM while in running state and with minimum service interruption, whereas non-Live migration is migration of VM in suspended mode and with maximum service interruption.

3.3 Contribution of Live Migration in Energy Management Scheduling Strategies

Considering the benefits of inculcating the practice of live migration into scheduling strategies, energy consumption in cloud data centers can be reduced. The existing pragmatic implementation has been discussed below to get a know-how on the research work which is already done in the area.

Sun et al. in [33] have studied the live migration of multiple VMs. Two strategies—"modified serial migration strategy" and "m-mixed migration strategy"—are proposed in this direction. The modified serial migration can take lesser migration time and downtime. The m mixed migration strategy uses the combination of both modified serial migration and the parallel migration strategy. It uses pre-copy migration with serial migration and post-copy with parallel migration strategy.

Sharkh et al. in [38] have considered the situations, where live migrations of VMs inversely affect the performance of system and hence proposed a technique "Smart VM Over Provision (SVOP)" using GreenCloud. This technique depends on "dynamic idleness prediction (DIP)" that uses prediction about future demand of VMs. The process used in SVOP resembles host-switching method in symbiotic organisms.

Abdullah et al. in [42] have proposed the integration of "fast best-fit decreasing (FBFD)" algorithm and "dynamic utilization rate (DUR)" algorithm using CloudSim. The difference between MBFD and new algorithm FBFD is that the decreasing list of VMs and increasing list of host list are dynamic. DUR suggests

Table 1 Existing energy scheduling strategies

Year	Author/s	Source	Summary
2008	Torres et al.	[10]	Combined effect of memory compression and request discrimination to decrease energy requirement of data center is studied.
2009	Cardosa et al.	[11]	Algorithm "PowerExpandMinMax" is proposed where min, max, and shares features inherent in virtualization technologies are used.
2010	Csorba et al.	[12]	A new method for deployment mapping is proposed which is based on "Ant Colony optimization" for multiple VM concurrently.
2010	Lee et al.	[13]	Two energy-conscious task consolidation heuristics "ECTC and MaxUtil" are proposed to maximize resource utilization
2010	Beloglazov et al.	[14]	Heuristics for dynamic reallocation of VMs is proposed
2011	Murtazaev et al.	[15]	Focused on number of VM migrations along with reducing number of active servers so proposed algorithm "Sercon" which is a server consolidation algorithm
2011	Garg et al.	[16]	Algorithm "carbon-efficient green policy(CEGP)" is proposed which schedules user application requests with given deadline on cloud datacenters
2012	Mazzucco et al.	[17]	Energy-aware allocation policies are introduced and evaluated which are based on the dynamic powering servers on and off
2013	Boru et al.	[18]	Data replication is studied which brings data closer to customers
2013	Tian et al.	[19]	Algorithm "dynamic bipartition-FirstFit (BFF)" is proposed where some parameters like start-time, an end-time, a processing time, and a capacity demand of VM on host physical machine (PM) are considered.
2013	Moreno et al.	[20]	Algorithm "server selection algorithm" is proposed which is based on workload heterogeneity
2013	Jeong et al.	[21]	Two live migration schemes are proposed which significantly reduce power consumption and to resolve the power overshooting issue.
2014	Singh et al.	[22]	Algorithm "energy-based efficient resource scheduling algorithm" is proposed which schedules the resources on the basis of energy as a QoS parameter
2014	Zhao et al.	[23]	A model "DREAM-CEP" is provided which provides an easy way to examine energy consumption of multicores operating with varying frequencies
2014	Horri et al.	[24]	A novel QoS-aware VMs consolidation approach is proposed that constructed technique based on resource utilization history of virtual machines on host machine

(continued)

Table 1 (continued)

Year	Author/s	Source	Summary
2014	Quang-Hung et al.	[25]	Two algorithms "MinDFT-ST and MinDFT-FT" are proposed which reduces the total completion times of all physical machines instead of reducing number of working physical machines
2015	Sami et al.	[26]	Resource allocation algorithm and server consolidation algorithm are proposed
2015	Selvi et al.	[27]	Virtual batching with memory management is proposed to manage power for computational and storage units
2015	Dong et al.	[28]	Greedy task scheduler "Most-efficient-server first scheme" is proposed which schedules tasks to the most energy-efficient servers of a data center.
2016	Gupta et al.	[29]	Enhanced bin-packing algorithm is proposed
2016	Alismail	[30]	An algorithm "VM scheduler" is proposed which aims to reduce number of working and heterogenous servers by switching off idle machines. This algorithm can improve resource utilization and energy consumption.
2016	Deng et al.	[31]	"Minimum average utilization difference (MAUD)" VM placement policy is proposed
2016	Sharma et al.	[32]	Overview of existing techniques is presented considering reliability and energy consumption of system. The classifications are done on the basis of resource failures, fault tolerance procedure, and energy management techniques.
2016	Sun et al.	[33]	For live migration of multiple VMs, two strategies are proposed—serial migration and the mixed migration. Mixed migration uses combination of serial migration and parallel migration strategies
2017	Science et al.	[34]	Several adaptive heuristic algorithms are proposed that are based on analyzing the historical data to optimize the VMs dynamic consolidation and to optimize the performance and energy consumption
2017	Bala et al.	[35]	Flowchart for power saving algorithm using max–min scheduling algorithm and DVFS is proposed
2017	Han et al.	[36]	Algorithm "resource-utilization-aware energy-efficient server consolidation algorithm(RUAEE)" is proposed to improve resource utilization while reducing the number of virtual machine live migrations
2017	Gupta et al.	[37]	Architectural model for enhanced bin-packing technique is proposed
2017	Sharkh et al.	[38]	"Smart VM over provision (SVOP)" algorithm is proposed based on "dynamic idleness prediction technique (DIP)" using prediction of future VM requirements. This algorithm can be implemented where live VM migration is inefficient

(continued)

Table 1 (continued)

Year	Author/s	Source	Summary
2017	Bermejo et al.	[39]	Study of resource management and resource allocation techniques
2017	Khoshkholghi et al.	[40]	Modified algorithms for overloaded host detection, underloaded host detection, VM selection, and VM placement are proposed which can provide high-quality service to customers along with decreasing the energy consumption of cloud system under SLA constraints.
2017	Chaabouni et al.	[41]	Modified policy for load detection, VM selection, and VM placement is proposed which aims to reduce energy, number of migrations, and number of host switching(on/off).
2017	Abdullah et al.	[42]	The "fast best-fit decreasing (FBFD)" algorithm for intelligent VMs allocating into hosts and "dynamic utilization rate (DUR)" algorithm for migrating VM migration is suggested.
2018	Diouani et al.	[43]	Model for dynamic resource management method which focuses on energy-performance trade-off
2018	Karuppasamy et al.	[44]	Algorithm "energy saving algorithm" is proposed to fulfill the requirement of users with least number of resources for best results
2018	Yadav et al.	[45]	Two flexible heuristic algorithms, "least medial square regression" for overloaded host detection and "minimum utilization prediction" for VM selection from overloaded hosts are proposed
2018	Rehman et al.	[46]	Two VM selection policies, "threshold-based selection (TBS)" and "capacity-based selection (CBS)" are proposed. For TBS, threshold value of resource utilization is the prime focus and for CBS, capacity of servers in resource utilization is important
2018	Kaur et al.	[47]	Summary of live migration techniques is given with parameters like pre- and post-copy methods used
2018	Mishra et al.	[48]	A VM selection technique is proposed using dynamic voltage frequency scaling (DVFS) to reduce energy consumption and makespan
2019	Sayadnavard et al.	[49]	A novel approach is proposed that gives priority to the reliability of all PMs along with shrinking the number of working PMs
2019	Jeba	[50]	Three algorithms-"Power_reduction", "VM_migration" and "sequential or random or maximum fairness" are proposed which implements dynamic scheduling of servers for efficient resource utilization
2019	Panda et al.	[51]	Algorithm "energy-efficient task scheduling algorithm (ETSA)" is proposed to measure energy efficiency and makespan in the heterogeneous environment

(continued)

Table 1 (continued)

Year	Author/s	Source	Summary
2019	Ali et al.	[52]	Research is done on various energy-efficient techniques and concluded that most researchers used VM consolidation and VM scheduling technique to reduce energy consumption
2019	Zakarya et al.	[53]	Trade-off between overall energy consumption and performance for heterogeneous workload is discussed
2019	Hao et al.	[54]	By using varying system pressures and network failure situations, the reliability of VM live migration and number of VM migrations are tested
2019	Mazrekaj et al.	[55]	Overview of some live migration techniques is proposed by providing information about hypervisor used, parameter metrics, and benefits
2020	Mandal et al.	[56]	Prediction policy is proposed for resource utilization based on linear regression
2020	John et al.	[57]	Review of some virtual machine techniques are discussed with some parameters like merit/demerit, resources used, and research gap
2020	Wang et al.	[58]	A VM consolidation mechanism using bio-inspired heuristics is proposed. By using host-switching techniques in symbiotic organisms, some heuristic functions are proposed which included both the "host utilization levels" and "resource utilization correlations" among co-located VMs

keeping some extra space on host machine for VM expansion instead of providing exact space for VM.

Rehman et al. in [46] have given new VM selection policies–

- TBS (Threshold-Based Selection) which focuses on putting over-loaded hosts below the given threshold value.
- CBS (Capacity-Based Threshold) which focuses on putting overloaded hosts below the given capacity value of host.

Mishra et al. in [48] used DVFS technique in VM selection process to reduce energy consumption and total makespan. An algorithm "EEDTSA (Energy-Efficient DVFS based Task Scheduling Algorithm)" is proposed which has better results than FCFS. According to the study, DVFS utilizes the dynamic voltage and frequency adjustments for a heterogenous cloud.

Sayadnavard et al. in [49] addressed the negative effects of more and frequent VM migration on the reliability of data centers. VM consolidation is implemented after checking the reliability of physical machine using Markov model. Results are validated on Cloudsim.

Ali et al. in [52] presented taxonomy of various energy-efficient techniques like "VM Selection Method," "VM Migration Method," "VM Placement Method," "DVFS-Aware Consolidation Method," "VM Scheduling Method," and "VM Allo-

cation Method." This paper concluded that more efficient algorithms are needed for heterogeneous cloud environment.

Zakarya et al. in [53] focused on performance of data centers under dynamic workload. Using CloudSim, a heuristic "Energy-Performance-Cost (Epc)" is proposed to achieve minimum energy consumption. The heuristic uses host performance differences during migration procedure.

Hao et al. in [54] investigated the reliability of VM live migration for dynamic system pressures and network failures. This paper concludes that following steps can reduce the number of migrations to increase reliability:

- Giving priority to VM that uses less memory.
- Selecting VM with less disk write requests.
- Packet delay time.
- Number of packet loss.
- Number of duplicate packets.
- Total corrupted packet rate.

Mandal et al. in [56] used live VM migration for optimizing load balancing in cloud data centers. Based on the requirement, this paper proposed "UPLRegA (Utility Prediction Linear Regression Analysis Algorithm)," where future prediction about resource utilization can be done using past resource utilization patterns using linear regression using CloudAnalyst.

Wang et al. in [58] considered total host's "CPU utilization levels" and "Resource Utilization Correlation (RUC)" among VMs on same physical machine during live migration. Based on these factors, this paper proposed two heuristic functions that behaved like host-switching techniques in symbiotic organisms (i.e., parasites and hosts). These functions are host susceptibility to measure host condition and symbiotic coefficient to measure correlations among VMs. Finally, this paper proposed VM migration algorithm for optimizing VM reallocations using CloudSim.

John et al. in [57] presented a review of VMC techniques taking parameters like merit/demerit, considered resources, research gap. The paper also discussed that migration of VM can minimize the number of active servers, but it can also degrade the quality of service of physical machines. Hence, migration decisions should be taken with great caution to make VM migration more productive.

Mazrekaj et al. In [55] presented a review of VMC techniques based on the type of live migration used—precopy or postcopy. The main focus of the review is some performance metrics like migration time, downtime, etc., which influence VM live migration procedure.

Kaur et al. in [47] gave detail of some live migration techniques along with the details of the precopy and postcopy methods used in them. It also explains benefits and weaknesses with respect to security of cloud along with other issues like Pre-copy Transfer Rate, Pre-copy page resend problem, and Post-copy missing page problem.

4 Discussion for Future Work

Resource Management and Energy Management in cloud computing are interrelated and affecting the performance of cloud service providers. VM Live migration plays an important role in both strategies. The available statistics [59] explicates that electricity consumption will hike from 632 to 1963 Billion Kilo Watt Hours by the end of 2020 and CO_2 emission would be ~1034 megatons which are awakening point before the cloud user entities to handle the issue. The observations that can contribute to future research work are presented below:

- The strategies proposed are more provider driven, and little focus has been given on fulfilling user requirements of parameters like deadline and success count [60].
- An additional stepping of providers performance evaluation should be added before resource scheduling for increasing the trust of user on service paradigm [61].
- A third layer of provider feedback should be added for taking feedback from customers on provider services with available infrastructure.
- Some factors should be considered for live migration of VM [62] like preparation time, resume time, downtime, number of pages transferred, and dirty pages in memory after transfer as without considering these parameters, their contribution to energy [63] management has posed extra overhead.
- The bin packing should not be considered as the only means of energy reduction and overall system performance [62, 63] should be tracked that can lead to better resource utilization.
- The notion of multitenancy has increased resource utilization but has escalated problem of energy management [3]. So, addressing the issue can be a major contributor in problem resolving.
- Researchers can work on challenges of VM live migration [4] to use the facility to take maximum advantage of VM live migration.

The discussion points gathered here can give possible future directions to the researchers for exploration, and the conclusion to the research study is presented in the subsequent section.

5 Conclusion

Reaping the benefits of cloud has exaggerated the problem of resource management and has captured the focus of service providers in discovering solutions that can mitigate the effect. The increasing traffic over cloud has made it difficult to extend the services with the available resources. Energy management has further widened the apprehensions of providers in terms of cost and hazardous effects on our surroundings. The consequences of such notions have somehow affected the trust

level of service extractors which has further necessitated to discover solutions in the stated directions. A deep conviction is to be made on available solutions that can let the user enjoy the technology-driven services without any major aftereffects. The study conducted here presents an extensive survey of the terms resource management, energy management, and live migration from different dimensions to assist researchers in the future work.

This paper presented a detailed survey of resource management methods and how it is related to energy consumption of the total system.

References

1. Dillon, T., Wu, C., & Chang, E. (2010). Cloud computing: Issues and challenges. Proceedings - international conference on advanced information networking and applications (pp. 27–33). AINA. https://doi.org/10.1109/AINA.2010.187.
2. Savu, L. (2011). Cloud computing: Deployment models, delivery models, risks and research challanges. 2011 international conference on computer and management, CAMAN 2011. https://doi.org/10.1109/CAMAN.2011.5778816.
3. Hari Krishna, B., Kiran, S., Murali, G., & Pradeep Kumar Reddy, R. (2016). Security issues in service model of cloud computing environment. *Procedia Computer Science, 87*, 246–251. https://doi.org/10.1016/j.procs.2016.05.156.
4. https://fortune.com/2019/09/18/internet-cloud-server-data-center-energy-consumption-renewable-coal/
5. Gupta K, Katiyar V. Survey of resource provisioning heuristics in cloud and their parameters International Journal of Computational Intelligence Research 13, 5 (2017), pp. 1283–1300. ISSN 0973–1873.
6. Aceto, G., Botta, A., De Donato, W., & Pescapè, A. (2013). Cloud monitoring: A survey. *Computer Networks, 57*(9), 2093–2115.
7. Krauter, K., Buyya, R., & Maheswaran, M. (2002). A taxonomy and survey of grid resource management systems for distributed computing. *Software: Practice and Experience, 32*(2), 135–164.
8. Rajasekar, B., & Manigandan, S. K. (2015). Efficient resource allocation strategies in cloud computing. *International Journal of Innovative Research in Computer and Communication Engineering., 3*(2), 1239–1244.
9. https://shodhganga.inflibnet.ac.in/bitstream/10603/219788/12/12_chapter3.pdf
10. Torres, J., Carrera, D., Hogan, K., Gavaldà, R., Beltran, V., & Poggi, N. (2008). Reducing wasted resources to help achieve green data centers. IPDPS Miami 2008 – proceedings of the 22nd IEEE international parallel and distributed processing symposium, program and CD-ROM. https://doi.org/10.1109/IPDPS.2008.4536219.
11. Cardosa, M., Korupolu, M. R., & Singh, A. (2009). Shares and utilities based power consolidation in virtualized server environments. 2009 IFIP/IEEE international symposium on integrated network management, IM 2009 (pp. 327–334). https://doi.org/10.1109/INM.2009.5188832.
12. Csorba, M. J., Meling, H., & Heegaard, P. E. (2010). Ant system for service deployment in private and public clouds. Proceeding of the 2nd workshop on bio-inspired algorithms for distributed systems, BADS '10 (pp. 19–28). https://doi.org/10.1145/1809018.1809024.
13. Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *Journal of Supercomputing, 60*(2), 268–280. https://doi.org/10.1007/s11227-010-0421-3.
14. Beloglazov, A., &Buyya, R. (2010). Energy efficient allocation of virtual machines in cloud data centers. CCGrid 2010 – 10th IEEE/ACM international conference on cluster, cloud, and grid computing (pp. 577–578). https://doi.org/10.1109/ccgrid.2010.45.

15. Murtazaev, A., & Oh, S. (2011). Sercon: Server consolidation algorithm using live migration of virtual machines for green computing. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India), 28*(3), 212–231. https://doi.org/10.4103/0256-4602.81230.
16. Garg, S. K., Yeo, C. S., &Buyya, R. (2012). Green cloud framework for improving carbon (pp. 491–502).
17. Mazzucco, M., & Dyachuk, D. (2012). Optimizing cloud providers revenues via energy efficient server allocation. *Sustainable Computing: Informatics and Systems, 2*(1), 1–12. https://doi.org/10.1016/j.suscom.2011.11.001.
18. Boru, D., Kliazovich, D., Granelli, F., Bouvry, P., &Zomaya, A. Y. (2013). Energy-efficient data replication in cloud computing datacenters (pp. 446–451).
19. Tian, W., Xiong, Q., & Cao, J. (2013). An online parallel scheduling method with application to energy-efficiency in cloud computing. 2006. https://doi.org/10.1007/s11227-013-0974-z.
20. Moreno, I. S., Yang, R., Xu, J., & Wo, T. (2013). Improved energy-efficiency in cloud datacenters with interference-aware virtual machine placement.
21. Jeong, J., Kim, S. H., Kim, H., Lee, J., & Seo, E. (2013). Analysis of virtual machine live-migration as a method for power-capping. *Journal of Supercomputing, 66*(3), 1629–1655. https://doi.org/10.1007/s11227-013-0956-1.
22. Singh, S., & Chana, I. (2014). Energy based Efficient Resource Scheduling: A Step Towards Green Computing. International Journal of Energy, Information and Communications 5(2), 35–52.
23. Zhao, X., & Jamali, N. (2014). Energy-aware resource allocation for multicores with per-core frequency scaling. *Journal of Internet Services and Applications, 5*, 1–15.
24. Horri, A., &Sadegh, M. (2014). Novel resource allocation algorithms to performance and energy efficiency in cloud computing. https://doi.org/10.1007/s11227-014-1224-8.
25. Quang-Hung, N., Le, D. K., Thoai, N., & Son, N. T. (2014). Heuristics for energy-aware VM allocation in HPC clouds. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8860*, 248–261. https://doi.org/10.1007/978-3-319-12778-1_19.
26. Sami, M., Haggag, M., & Salem, D. (2015). Resource allocation and server consolidation algorithms for green computing. *International Journal of Scientific & Engineering Research, 6*(12), 313–316.
27. Selvi, R., &Anitha, V. K. (2015). Energy constrained resource scheduling for cloud environment. 3(2):417–421.
28. Dong, Z., Liu, N., & Rojas-cessa, R. (2015). Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers. https://doi.org/10.1186/s13677-015-0031-y.
29. Gupta, K., &Katiyar, V. (2016). Energy aware virtual machine migration techniques for cloud environment. *Journal of Grid Computing, 141*(2).
30. Alismail, S. M. (2016). Green algorithm to reduce the energy consumption in cloud computing data centres (pp. 557–561).
31. Deng, D., He, K., & Chen, Y. (2016). Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers.
32. Sharma, Y., Javadi, B., Si, W., & Sun, D. (2016). Reliability and energy efficiency in cloud computing systems: Survey and taxonomy. *Journal of Network and Computer Applications, 74*, 66–85. https://doi.org/10.1016/j.jnca.2016.08.010.
33. Sun, G., Liao, D., Anand, V., Zhao, D., & Yu, H. (2016). A new technique for efficient live migration of multiple virtual machines. *Future Generation Computer Systems, 55*(February), 74–86. https://doi.org/10.1016/j.future.2015.09.005.
34. Science, C., Sciences, I., Science, C., & Sciences, I. (2017). Heuristic algorithms for energy and performance dynamic optimization in cloud computing. *Yifei Zhang Shuguang Zhao., 36*, 1335–1360. https://doi.org/10.4149/cai.

35. Bala, R., & Mann, E. J. (2017). A research paper on green computing using energy efficient task allocation strategy in cloud environment. *International Journals of Advanced Research in Computer Science and Software Engineering, 6*, 186–191. https://doi.org/10.23956/ijarcsse/V7I6/0248.
36. Han, G., Que, W., Jia, G., & Zhang, W. (2017). Author' s accepted manuscript resource-utilization-aware energy efficient server consolidation algorithm for green computing in IIOT. *Journal of Network and Computer Applications*. https://doi.org/10.1016/j.jnca.2017.07.011.
37. Gupta, K., & Katiyar, V. (2017). Energy-aware scheduling framework for resource allocation in a virtualized cloud data centre. *International Journal of Engineering and Technology, 9*(2), 558–563. https://doi.org/10.21817/ijet/2017/v9i2/170902032.
38. Sharkh, M. A., &Shami, A. (2017). An Evergreen Cloud: Optimizing Energy Efficiency in Heterogeneous Cloud Computing Architectures. *Vehicular Communications*, February. https://doi.org/10.1016/j.vehcom.2017.02.004.
39. Bermejo, B., Filiposka, S., Juiz, C., Gómez, B., & Guerrero, C. (n.d.). Improving the energy efficiency in cloud computing data centres through resource allocation techniques (pp. 211–236). https://doi.org/10.1007/978-981-10-5026-8.
40. Khoshkholghi, M.A., Derahman, M.N., Abdullah, A., Subramaniam, S., Othman, M. (2017). Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers (p. 1). https://doi.org/10.1109/ACCESS.2017.2711043
41. Chaabouni, T., & Khemakhem, M. (2018). Energy management strategy in cloud computing: A perspective study. *Journal of Supercomputing, 74*(12), 6569–6597. https://doi.org/10.1007/s11227-017-2154-z.
42. Abdullah, M., Lu, K., Wieder, P., & Yahyapour, R. (2017). A heuristic-based approach for dynamic VMs consolidation in cloud data centers. *Arabian Journal for Science and Engineering, 42*(8), 3535–3549. https://doi.org/10.1007/s13369-017-2580-5.
43. Diouani, S., &Medromi, H. (2018). Green cloud computing: Efficient energy-aware and dynamic resources management in data centers. July, 10–14. https://doi.org/10.14569/IJACSA.2018.090717
44. Karuppasamy, M., & Balakannan, S. P. (2018). Energy saving from cloud resources for a sustainable green cloud computing environment. *Journal of Cyber Security and Mobility, 7*, 95–108. https://doi.org/10.13052/jcsm2245-1439.718.
45. Yadav, R., Zhang, W., Li, K., Liu, C., Shafiq, M., & Karn, N. K. (2020). An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center. *Wireless Networks, 26*(3), 1905–1919. https://doi.org/10.1007/s11276-018-1874-1.
46. Rehman, Q. H. U., & Shu, G. (2019). Efficient VM selection heuristics for dynamic VM consolidation in cloud datacenters. Proceedings - 16th IEEE international symposium on parallel and distributed processing with applications, 17th IEEE international conference on ubiquitous computing and communications, 8th IEEE international conference on big data and cloud computing, 11th IEEE international conference on social computing and networking and 8th IEEE international conference on sustainable computing and communications, ISPA/IUCC/BDCloud/SocialCom/SustainCom 2018 (pp. 832–839). https://doi.org/10.1109/BDCloud.2018.00124
47. Kaur, J., & Chana, I. (2018). Review of live virtual machine migration techniques in cloud computing. 2018 international conference on circuits and systems in digital enterprise technology, ICCSDET 2018 (pp. 1–6). https://doi.org/10.1109/ICCSDET.2018.8821170.
48. Mishra, S. K., Mishra, S., Bharti, S. K., Sahoo, B., Puthal, D., & Kumar, M. (2018). VM selection using DVFS technique to minimize energy consumption in cloud system. Proceedings – 2018 international conference on information technology, ICIT 2018, December (pp. 284–289). https://doi.org/10.1109/ICIT.2018.00064.
49. Sayadnavard, M. H., ToroghiHaghighat, A., & Rahmani, A. M. (2019). A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers. *Journal of Supercomputing, 75*(4), 2126–2147. https://doi.org/10.1007/s11227-018-2709-7.

50. Jeba, J. A. (2019). Towards green cloud computing an algorithmic approach for energy minimization in cloud data centers. *International Journal of Cloud Applications and Computing, 9*(1). https://doi.org/10.4018/IJCAC.2019010105.
51. Panda, S. K., & Jana, P. K. (2019). An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. *Cluster Computing, 22*(2), 509–527. https://doi.org/10.1007/s10586-018-2858-8.
52. Ali, S. A., Affan, M., &Alam, M. (2019). A study of efficient energy management techniques for cloud computing environment. Proceedings of the 9th international conference on cloud computing, data science and engineering, confluence 2019 (pp. 13–18). https://doi.org/10.1109/CONFLUENCE.2019.8776977.
53. Zakarya, M., & Gillam, L. (2019). Managing energy, performance and cost in large scale heterogeneous datacenters using migrations. *Future Generation Computer Systems, 93*, 529–547. https://doi.org/10.1016/j.future.2018.10.044.
54. Hao, J. et al. (2019). Live migration of virtual machines in OpenStack: A perspective from reliability.
55. Mazrekaj, A., Nuza, S., Zatriqi, M., & Alimehaj, V. (2019). An overview of virtual machine live migration techniques. *International Journal of Electrical and Computer Engineering, 9*(5), 4433–4440. https://doi.org/10.11591/ijece.v9i5.pp4433-4440.
56. Mandal, G., Dam, S., Dasgupta, K., & Dutta, P. (2020). A linear regression-based resource utilization prediction policy for live migration in cloud computing. *Studies in Computational Intelligence, 870*, 109–128. https://doi.org/10.1007/978-981-15-1041-0_7.
57. John, N. P., & Bindu, R. B. V. (2020). A review on dynamic consolidation of virtual machines for effective energy management and resource utilization in data centres of cloud computing. Proceedings of the 4th international conference on computing methodologies and communication, ICCMC 2020, ICCMC (pp. 614–619). https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000114.
58. Wang, J. V., Ganganath, N., Cheng, C. T., & Tse, C. K. (2020). Bio-inspired heuristics for VM consolidation in cloud data centers. *IEEE Systems Journal, 14*(1), 152–163. https://doi.org/10.1109/JSYST.2019.2900671.
59. https://www.theguardian.com/environment/2010/apr/30/cloud-computing-carbon-emissions, 2020.
60. Venkatadri, M., Agarwal, A., & Pasricha, A. (2019). Self-characteristics based energy-efficient resource scheduling for cloud. *Procedia Computer Science, 152*, 204–211.
61. Faruqui, N., Yousuf, M. A., Chakraborty, P., & Hossain, M. S. (2020). Innovative automation algorithm in micro-multinational data-entry industry. In Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, LNICST (Vol. 325 LNICST, pp. 680–692). Springer. https://doi.org/10.1007/978-3-030-52856-0_54.
62. Khatun, M., Islam, M.I., Chakraborty, P., Ahmed, T., Sarker, A., and Shamim-Ul-Islam, M. (2020). Secrecy Capacity via Cooperative Transmitting under Rayleigh and Nakagami-m Fading Channel. *Institute of Electrical and Electronics Engineers (IEEE)*, 82–85.
63. Dewangan, B. K., Agarwal, A., Venkatadri, M., & Pasricha, A. (2018, December). Autonomic cloud resource management. In 2018 fifth international conference on parallel, distributed and grid computing (PDGC) (pp. 138–143). IEEE.

Virtual Machine Scaling in Autonomic Cloud Resource Management

Avita Katal, Vitesh Sethi, and Saksham Lamba

1 Introduction to Virtualization and Virtual Machine Scaling

The concept of virtualization was first introduced by IBM during the late 1960s and early 1970s in the era of mainframe technology, when it was developing time sharing solutions [1]. Time sharing solutions share a group of resources between the users to increase the utilization of the resources. Virtualization is achieved using a software called Hypervisor or Virtual Machine Monitor (VMM) that creates a simulated computer environment; virtual machine. Guest operating systems are installed on a single physical hardware system whose resources are multiplexed to guest machines with the help of hypervisor. Thus, ensuring full utilization of the resources and cutting down the cost expenditure.

Hypervisor or VMM can be categorized as:

- *Type 1 Hypervisor* also called as Native or Bare Metal Hypervisor operates directly on top of the system's hardware so as to keep a control on the hardware and monitor the virtual machines. Examples: Oracle VM, VMWare ESX, etc. Fig. 1 shows the architecture of the type 1 hypervisor.
- *Type 2 Hypervisor* also called as the Hosted Hypervisor runs on top of the guest's OS. Example: Oracle VM VirtualBox, KVM, QEMU, etc. Fig. 2 shows the architecture of the type 2 hypervisor.

  Virtualization treats the physical hardware system as a combination of resources like storage, memory, CPU, servers, etc. rather than a discrete system and distributes the resources according to the need. It solves the challenges of cloud computing by providing the following advantages:

A. Katal (✉) · V. Sethi · S. Lamba
Department of Virtualization, University of Petroleum and Energy Studies, Dehradun, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_17

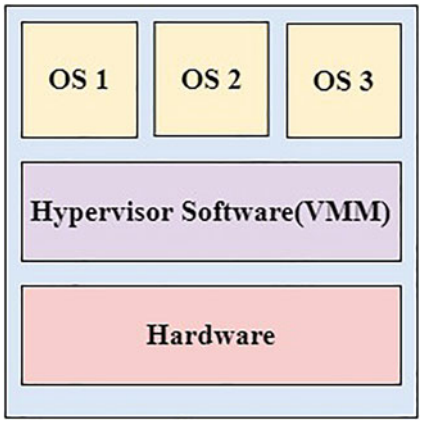

Fig. 1 Type 1 hypervisor

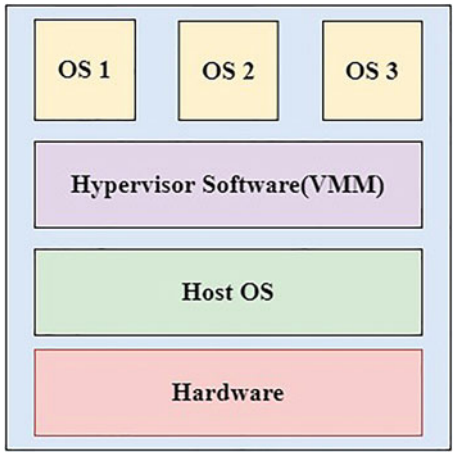

Fig. 2 Type 2 hypervisor

- Virtualization helps in encapsulation and migration of the workload to the systems that are idle. Thus, it helps in better usage of the existing resources.
- Virtualization cuts down the cost of buying more resources and ensures a great throughput from the system through storage and server consolidation.
- Since, virtualization consolidates the resources, the total energy consumption reduces leading to lower costs.

Expansion of the data centers can pose many problems like higher costs of building and cooling the servers. Since, virtualization allows many virtual machines to run over a single hardware system, it prevents the problem of Server Sprawling. Sometimes, the resources required by the applications running in the virtual machines may not be sufficient to process all of the data. To overcome this problem of providing on demand resources, virtual machine scaling comes into picture which provides resources to the virtual machines when there is a sudden spike in the workload.

Virtualization and Scalability are the two important concepts/mechanisms in Cloud Computing. Virtualization is a technique that enables the utilization of the existing resources fully by allowing to share a single physical instance of a resource or an application among multiple customers and organizations. Scalability means to quickly and easily decrease or increase the resources, performance, or functionalities according to the user's need. Scaling can be done manually or automatically. Auto-scaling techniques are the ones that run on the cloud service providers (CSP) side in order to automate the scaling according to the load conditions. It reduces the human efforts and is responsible for providing quick service. The choice of auto-scaling has reduced due to many limitations like response time, SLA violations, cost of service and quality of service (QoS).

Many virtual machines are hosted within each server acting as an independent physical computer. But sometimes, a case may arise that all the workload is being transferred to a single virtual machine leading to overloading of that virtual machine while the other virtual machines may remain idle, thus leading to underutilization of those virtual machines. This problem can be solved by implementing load balancing. As the name suggests, a load balancer checks which virtual machines are idle and which are overloaded and then accordingly distributes the workload. Thus, maintaining the system's efficiency and preventing downtime. Also, the resources that are provided to all the virtual machines may get exhausted in a single server too. In this case, all the virtual machines get overloaded and the load balancer cannot transfer the load due to the unavailability of the resources on the same server. To solve this problem, the migration of the virtual machine is done from the source server to the destination server. This process is called virtual machine migration. Since, now the VM has been migrated to the destination server having sufficient resources, load balancing can again be incorporated and the transfer of workload does not stop. This whole process ensures that the system is scalable and that it does not encounter any downtime.

2 Need of Virtual Machine Scaling

Scaling in cloud computing helps the user process the data easily by providing features like improved performance, low costs, high availability, low power consumption and increased resource capacity.

a. Performance: VM scaling helps in the increasing throughput of the system, whatever may be the type of workload running, the resources are provided accordingly. The performance of the system can be increased by applying the two of the below mentioned methods. The first method is the creation of the replica of the virtual machine and then distributing this workload among these replicas. This process of the creation of the replicas can be achieved through horizontal scaling. This technique initiates the creation of a replica of the virtual machine in which the processing of the applications can occur. Even though this technique

increases the performance of the virtual machine, it consumes a lot of CPU time and also uses the network bandwidth when a virtual machine is transferred from one machine to another using live VM migration. To overcome this problem, another type of scaling called vertical scaling is implemented, which increases the efficiency by scaling up the resources of a single VM, thus reducing CPU time and preventing unnecessary consumption of bandwidth. The detailed explanation about the two types of scaling would be covered in the next sections.

b. *Cost*: One of the major benefits of VM scaling is cost optimization, that is, there is no need to buy additional servers or resources. Users need not to worry when there is sudden hike in the workload as VM scaling provides the resources on demand. The users need to pay only for the resources that their application is using. The cost of maintenance and cooling of the servers is also eliminated. Since, there is no need for the users to buy additional servers, the power consumption is also reduced to a great extent.
c. *Increased capacity*: VM scaling is very helpful in increasing the capacity of the virtual machines. The resources of the virtual machine are increased through virtual scaling so as to prevent downtime.
d. *Energy*: The servers in the data centers consume a lot of energy. The main problem is that even when there is not too much of workload, the servers would still be operating and consuming a lot of power. VM scaling prevents this unnecessary energy utilization by providing the resources of the applications as per the demand, thus reducing the need to buy and install additional servers.
e. *Availability*: The most important feature of VM scaling is to provide the required CPU power, memory, and storage on demand to the applications. This helps in faster processing of data because the resources are always available when needed.

3 Methods to Implement Scaling

Scaling can be implemented by the following methods: vertical scaling and horizontal scaling.

3.1 Vertical Scaling

Vertical scaling (shown in Fig. 3) increases the allocated resources of a single virtual machine so as to cater the needs of the workload at runtime. Vertical scaling provides flexibility to the system so as to deal with different kinds of workload. Some of the works [2, 3] deal with only scaling of the CPU, while the others [4, 5] focus on resizing the memory. Some of the techniques that are used for memory scaling are exponential moving average (EMA), page faults, ballooning [6]. While some techniques are used both for the CPU scaling and memory scaling as in [7, 8].

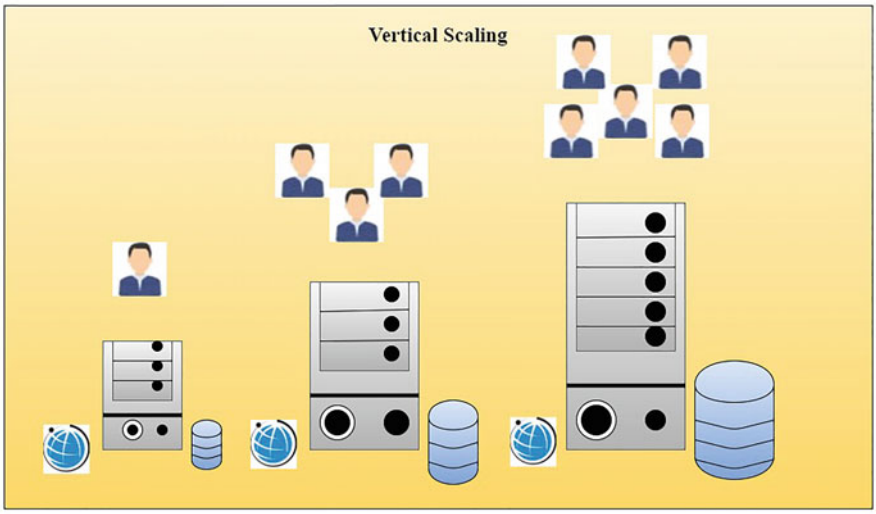

Fig. 3 Vertical scaling (resources are increased in a single machine)

The authors in [9] have also proposed a process for CPU and memory scaling. Two of the cloud vendors that provide these features to their users are: ProfitBricks and RightScale.

For further allowing vertical scaling when there are not much resources on the host server, migration can also be considered as a solution [10]. Migration can also be further classified into two types: live migration and non-Live migration as shown in Figs. 4 and 5, respectively. Live migration has two main categories: pre-copy and post-copy. In pre-copy technique the transfer of the memory pages occurs while the virtual machine is still running on the host server [11]. If some of the pages (also known as dirty pages) are changed at the time of the transfer of the memory pages, these will be copied again till the number of the copied pages is more than the dirty pages, otherwise the host virtual machine will be terminated. After that copying of the remaining dirty pages to the destination virtual machine would occur. While in the case of post-copy technique, the VM is first suspended from the host and the minimum processor memory is copied to the destination host and the VM is again started. It then starts fetching memory pages from the source.

3.1.1 Memory Scaling Algorithm

In the memory scaling algorithms [12, 13], the memory utilization of each VM is studied and compared with the maximum memory threshold value. If the memory utilization is equal to or more than the maximum memory threshold value, then the counter for the maximum memory utilization is increased and the counter of minimum memory utilization is reset, where R_{mu} represents Memory maximum

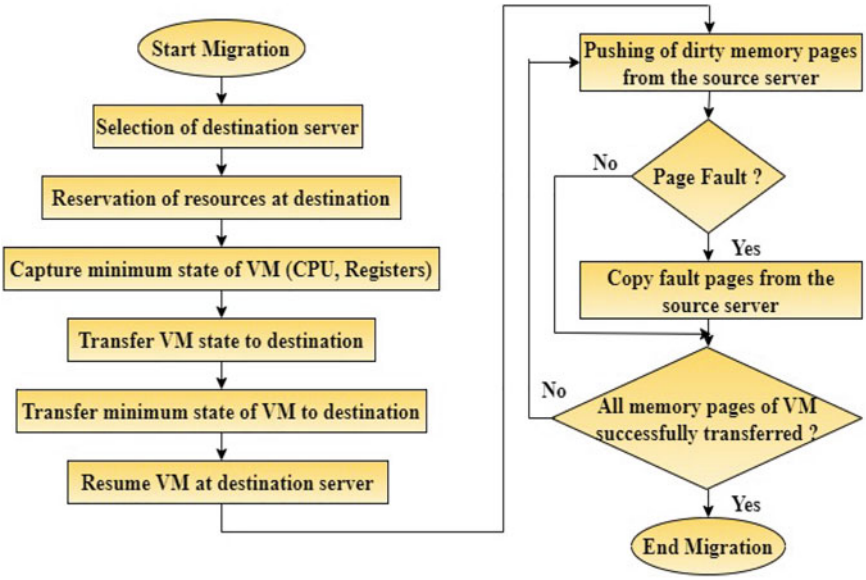

Fig. 4 Virtual machine live migration

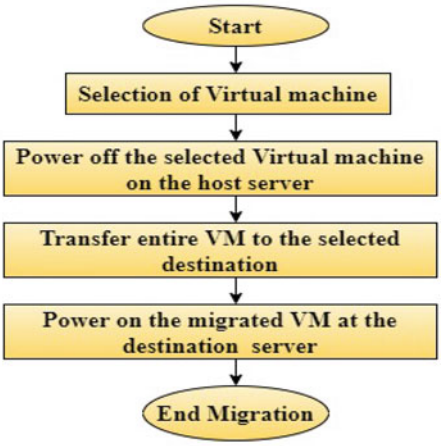

Fig. 5 Virtual machine non-live migration

threshold defined, R_{mn} represents Memory minimum threshold defined, TTR_t represents Time threshold counter defined for Memory, T represents Time interval defined to read the Memory utilization of VMs, RT_u represents Max memory iteration count, RT_m represents Min memory iteration count, and R_u represents Memory utilization of VM_x.

Algorithm 1: Memory scaling algorithm

Read memory usage R_u for each VM_u initialization;
if $R_u \geqslant R_{mu}$ **then**
  Increment the RT_u for theVM_u;
  Reset RT_m for the VM_u;
else
  if $R_u \leqslant R_{mn}$;
  Increase the RT_m for the VM_u ;
  Reset RT_u for the VM_u ;
end
if $RT_u > TTR_t$ *and if there is the availability of unused memory* **then**
  start the virtual machine up-scaling for memory;
  go to step 5;
else
  if $RT_m > TTR_t$;
  start the virtual machine down scaling for memory;
  move to step 5;
end
Move to Step 1 ;
Reset the RT_u and RT_m ;
Move to Step 1 ;

3.1.2 CPU Scaling Algorithm

In the CPU scaling algorithm [12], the CPU usage of each VM is studied and compared with the maximum CPU threshold value. If the CPU usage is equal to or

Algorithm 2: CPU scaling algorithm

Read CPU usage P_u for each VM_u initialization;
if $P_u \geqslant P_{mu}$ **then**
  Increase the PT_u for the VM_u;
  Reset PT_m for the VM_u;
else
  if $P_u \leqslant P_{mn}$;
  Increment the PT_m for the VM_u ;
  Reset PT_u for the VM_u ;
end
if $PT_u > TTP_t$ *and if computing resources available* **then**
  start the Virtual Machine up-scaling for CPU;
  go to step 5;
else
  if $PT_m > TTP_t$;
  start the Virtual Machine down scaling for CPU;
  move to step 5;
end
Move to Step 1 ;
Reset the PT_u and PT_m ;
Move to Step 1 ;

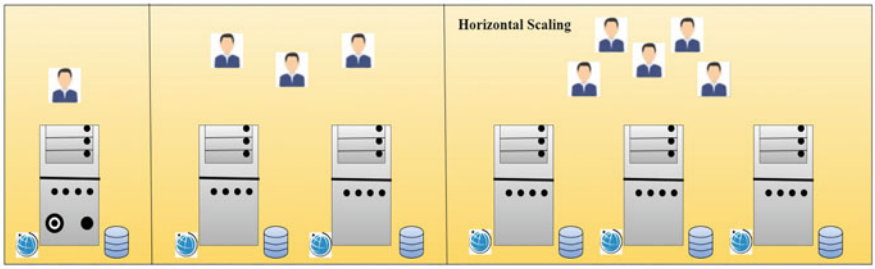

Fig. 6 Horizontal scaling

more than the maximum CPU threshold value, then the counter for the maximum CPU utilization is increased and the counter of minimum CPU utilization is reset, where P_{mu} represents CPU maximum threshold defined, P_{mn} represents CPU minimum threshold defined, TTP_t represents Time threshold counter defined for CPU, T represents Time interval defined to read the CPU usage of VMs, PT_u represents Max CPU iteration count, PT_m represents Min CPU iteration count, and P_u represents CPU utilization of VM_x.

3.2 Horizontal Scaling

Horizontal scaling (shown in Fig. 6) is the technique in which the addition or removal of instances occurs on different physical machines which are located at different locations. Load balancer plays an important role in this technique by distributing the workload between various instances. This method is commonly used and mostly cloud providers like Amazon and AzureWatch use this technique to distribute the workload among different instances.

4 Modes or Policies

So as to perform actions related to scaling, there is a need for interactions or manners which are known as policies or modes. Generally, they are categorized as:*automatic mode and programmable mode*. In automatic mode as the name suggests, all scaling actions are performed without anyone's intervention but in programmable mode which is quite similar to manual mode, the actions are done using application programming interface (API) calls.

4.1 Automatic Mode

In this policy, all of the actions are done automatically without anyone's intervention. Automatic mode can further be classified as reactive and proactive, where proactive is much more cost effective as it predicts the future requirements for the resources and triggers the action on the basis of this forecasting.

4.1.1 Reactive Mode

In this mode, scaling occurs on the basis of a threshold value. So as to process any type of application, the system first reacts to the type of load provided and triggers the scaling actions accordingly.

Thresholds here can be divided into two categories static thresholds and dynamic thresholds.

(a) Static thresholds: The scaling processes are triggered to scale up or down the resources only when the certain conditions are met. This type of policy is dependent on certain thresholds or the service level agreement (SLA) requirements and the actions are taken on the basis of measurement of metrics like memory usage, latency, CPU usage, etc. At least two thresholds are used for each of the performance metrics. For example, if the usage of the CPU is more than 75% and if this condition lasts for more than 8 min, scaling of the resources would be done.

One such example of work done is by the authors in [14] where they have presented an elastic scaling technique that utilizes the cost-aware factor so as to find and examine the bottlenecks in multi-tier applications based on the cloud. The authors implemented an adaptive scaling algorithm to lower the costs acquired by the customers of cloud services used by them. Thus, helping them scale their applications at bottleneck tiers, and introduce a platform to automate scaling of the applications. The platform named Imperial Smart Scaling engine (iSSe) was introduced to handle elastic scaling of multi-tier cloud applications. It acts as middleware between the consumers and cloud vendors. The authors gave a short description of the CAS algorithm and introduced two cost-aware benchmarks to advise scaling down and up of applications. Two algorithms based on these two criteria are the Cost-Aware-Capacity-Estimation (CACE)-For-Scaling-Up and CACE-For-Scaling-Down. The CAS algorithm is initiated after the deployment of the application and keeps executing till the application of the processing is completed. Whenever a change in the amount of input workload is detected, an estimation based on capacity, either CACE-For-Scaling-Up algorithm or CACE-For-Scaling-Down, gets triggered by CAS algorithm. Using the capacity estimation, the addition or removal of servers is done. An automatic reactive scaling technique is applied by the CAS algorithm like RightScale and Amazon WS. The two criteria on which Cost-Aware-Capacity-Estimation (CACE)-For-Scaling-Up and CACE-For-Scaling-Down are

based are: consumed cost/decreased response time (CC/DRT) ratio for CACE-for-Scaling-Up and Saved Cost/Increased Response Time (SC/IRT) for CACE-for-Scaling-Down. This algorithm reduces the latency, while at the same time keeping the deployment cost low.

(b) Dynamic thresholds: Static thresholds cannot be changed and are dependent on the user-defined value which is fixed, while the dynamic thresholds scale the resources at run time according to the type of workload of the applications.

One such work is done by the authors in [15] where they have designed a technique for the dynamic unification of the VMs which is based on adaptive usage thresholds and makes sure that the service level agreement (SLA) requirements are met. The authors validated a high throughput of the technique across various types of workloads using the workload traces.

The operations of the system can be categorized into two parts: (1) the selection of the virtual machines that need to be transferred so as to enhance the allocation; and (2) the placement of the chosen virtual machines for migration and the new virtual machines that are asked by the users on the physical nodes.

For the selection of the virtual machines to migrate, the authors have proposed four heuristics in their previous work [16]. The first heuristic known as single threshold (ST) sets an upper utilization threshold for the hosts and the placement of the virtual machines while keeping the total utilization of the CPU lower than this threshold, while the other three heuristics work by setting an upper and lower utilization thresholds for hosts and keeping the total usage of the CPU by all the virtual machines among these thresholds. The authors proposed policies for the selection of virtual machines that need to be transferred from an overloaded host:

- Minimization of Migrations (MM)—To lower the migration overhead, the number of virtual machines that is migrated is reduced.
- Highest Potential Growth (HPG)—migrate only those virtual machines that have less CPU utilization, thus minimizing the total increase of utilization and service level agreement violation.
- Random choice—randomly choose the unnecessary VMs.

Since constant values of thresholds are not relevant for a dynamic environment, a system should exist that automatically scales the resources up or down depending upon the type of workload exhibited by the applications. The authors here proposed a method for auto adjustment of the resources. This method was based on a statistical analysis of the data that is gathered during the lifetime of the virtual machine.

One *Reactive Mode Technique* applied for VM Scaling is Light Weight Scaling Algorithm (LS)[17]. If there is a y-tier application that requires scaling through the LS algorithm, this application has m server parts which means that its server set Q contains m servers where s_i belongs to Q and i = 1,. . . . . ,m. After the application is deployed, the LS algorithm is started and is made to run till the whole of the application ends. When the observed response time is more than the higher bound of the desired response time, the light weight scaling up algorithm is triggered by the LS algorithm. Similarly, the light weight scaling down algorithm is triggered when

the observed response time is less than the lower bound of the desired response time. The pseudo code for LS algorithm is given below:

Algorithm 3: LS algorithm

```
Input: Q,(U_l ,U_b) initialization;
while the processing of application is not finished do
    Monitor U_o once every few minutes;
    if U_o > U_b then
        LSU (Q, (U_l, U_b), U_o);
    else
        if U_o < U_l;
        LSD (Q, (U_l , U_b), U_o);
    end
end
```

The LS algorithm works in two steps. First, fine-grained scaling is conducted by changing the configuration of each virtual machine resource. This makes sure that the algorithm consumes very less amount of computing resources for the management of the resources. This scaling is finished in a few milliseconds, which provides high response time to the consumers. The second step involves a reactive scaling process that is autonomous. This process is somewhat identical to the process that is implemented by Rightscale and EC2. This type of scaling does not require any previous information about the application. Whenever an application is scaled, the requests that are coming are automatically distributed by the load balancing servers.

Light weight scaling up algorithm (LSU) reduces the latency of the application below the upper bound, and at that time minimizes the increase of the number of virtual machine instances.

Three types of scaling techniques are implemented by the LSU algorithm according to the level of priorities. First, the application response time is reduced by self-healing and resource level scalings by increasing the allocated resources to the virtual machine. But these two scalings have certain drawbacks. For example, a 32-bit OS can maximum be allocated a memory of 4 GB. If the response time target is met by one of the abovementioned scalings, the LSU is achieved. If the above requirements are not met, a new VM instance is needed to be added in a new physical machine. *Self-healing scaling* is the type of scaling in which any two of the virtual machines of the application server are operated on the same source machine, the resources of these virtual machines may complement each other. In *resource level scaling*, if a host machine that operates an application's server has unused resources, then these resources can be utilized in the scaling process without affecting the other applications hosted in these physical machines. One method to make the resources available is to preserve some of these resources in advance. Since a physical machine can generate a large number of CPUs theoretically, the availability of these CPU resources can be monitored in accordance with the CPU utilization of the physical machine. VM scaling up technique uses the complement

of the resource and self-healing scalings. VM level scaling up has the lowest chance to get triggered.
Light weight scaling down algorithm removes all the unnecessary VMs and resources from an application, while at the same time maintaining the response time of the system.

4.1.2 Proactive Mode

This mode predicts the future requirements for the resources and triggers the action on the basis of this forecasting.

4.1.3 Time Series Analysis

Time series analysis is a succession of estimations that are considered at constant interims [18]. Time series analysis predicts the future requirements and needs by keeping a check on the repetitive patterns in the workload. The scaling is performed on the basis of this prediction. This proactive scaling technique has two major aims: first to predict the values of the time series on the basis of the last recorded results. The second objective of this technique is to find the repeating patterns and then use these results to forecast the future values. The generation of the future values is done in the following manner: the latest history of the usage of the resource is the input. The technique then estimates the future points on the basis of these input values. To achieve the first aim of the time series analysis, the following techniques are used: *Auto-Regression, Auto-Regression Moving Average (ARMA)*, etc. To achieve the second objective, the repetitive patterns are observed by the following techniques: *pattern matching, auto correlation*, etc.

(A) Moving Average: The moving average method irons out the variations of the data by taking the mean of it. This method eliminates the fluctuations and thus estimates the trend. The simplest of all means is to take the arithmetic mean for the measurement of the trend. The moving average of any period n is a series of consecutive arithmetic means of the n terms at a time. The average is computed by using first, second, third etc values considering n data values at a time. Thus, the first average taken is the mean of the first n terms. The second arithmetic mean is the average of the n terms that starts from the second data till the $(n + 1)$th term.
The major drawback of this technique is to identify the extent of moving average which eliminates the oscillatory variations. This technique expects that the pattern is linear; however, it is not the situation generally and cannot be used for predicting the future trends which is the major objective of the time series analysis.
The authors in [19] have proposed prediction model based on double exponential smoothing which takes into consideration not only the present condition of

the resources but also records of the past. So as to simplify model building, two presumptions about the prediction are presented: (1) the performance characteristic of each of the resource can be calculated and can be explained by some measurements. (2) The measurement of the performance can be calculated and collected non-intrusively. The monitoring and collection of the information uses the operating system; the load of which requires no consideration.

The authors introduced two components of performance, memory, and CPU. The formulas that calculate the CPU and memory utilization are shown below:

$$C_{util} = [C_{user} + C_{sys}/C_{total}] \times 100 \tag{1}$$

$$M_{util} = [(M_{user} - M_{cache}/M_{total}] \times 100 \tag{2}$$

where *util* represents the rate of usage of a resource, *user* represents rate of usage by a user, *sys* represents the rate of usage of a system, *cache* represents the rate of usage of cache memory, and *total* represents the maximum available usage rate. To get the efficiency of the prediction model, summary of the squared errors (SSE) or mean of the squared errors (MSE) can be used. The authors have used SSE that can be computed by using the below formula

$$SSE = \sum_{i=1}^{n} (S_i - y_i)^2 \tag{3}$$

Here, S_i represents the forecasted value of time i-period and y_i denotes the real value of time i-period.

The authors have introduced a resource management log (RML) that keeps information of the resource that the customers have used earlier. As soon as a job scheduling is completed by a customer, the resources (CPU and Memory) used are added to RML. After which, the smoothing factor alpha and the initial exponential smoothing factor can be obtained using the time series. The resource needed by the user can be forecasted the next time and the user will only need to reserve it.

(B) Auto-Regression: The auto-regression model (AR) is another type of time series analysis model that uses the outcomes from the previous observations as input to a regression equation so as to forecast a value at the consecutive step. This method of predicting the next value is very accurate. This model is very flexible and can handle a wide range of problems. Through this mode it becomes clear that the output variable is directly proportional to the previous observations.

(C) Auto-regression Moving Average: The auto-regression moving average model (ARMA) is a tool to understand and forecast the future points. There are two parts of this model: the AR part is responsible for variable regression on its own previous value. The MA part is responsible for error modeling term as a linear combination of these error terms that occur contemporaneously more

than one time in the past. This model is known as the ARMA (r,s) model where r represents the order of the AR part and s represents the order of the MA part. The authors in [20] have proposed an auto-scaling system that exploits the heterogeneous resources by mentioning various levels of QoS requirements. This proposed technique chooses a resource scaling plan on the basis of requirements of both workload and customer. The auto-scaling system presented automatically scales a web application as the throughput changes at fixed intervals. The intervals are kept to 10 min. This system works along with the services that are arranged by an open-source runtime environment called ConPaaS, so as to host applications in cloud infrastructure. This can also be merged with any other Platform as a Service (PaaS) as this system is dependent on the below mentioned services provided by the platform: the monitoring engine and the resource manager. The former is used to track the workload of the application and the resources of the system. The architecture of this system consists of the following components: Profiler, Predictor, Dynamic load balancer, and Scaler.

(D) Holt Winters: Holt Winters model forecasts the behavior of a pattern of values. It is considered as the most important model for time series. Holt Winters model is ubiquitous in some of the applications like monitoring where it is made to be used for anomaly detection and capacity planning. This model is based on time series. Predicting future workload or output requires a model and Holt Winters is a path for the modeling of three aspects of the time series: a cyclical repeating pattern (seasonality), a slope (trend) over time, and a typical value (average). This model utilizes exponential smoothing to encode large historical values and use these values to forecast results for the current time and the future. Holt Winters model uses various parameters: one for each smoothing (alpha, beta, gamma), the length of a season, and the number of periods in a season.

The authors in [21] discuss the self-configuration approaches that can be implemented in the infrastructure of the cloud; the first approach works on a predictive model which is based on past data and the second approach utilizes the agents that monitor the efficiency of the virtual machines' resources in real time. The aim of the input workload requests is to carry out computational stress analysis, and thus find the behavior of the prediction and self-configuration algorithms. The architecture of this approach has five different modules: elastic adapter, admission control, actuator agent, monitor agent, and load prediction policies. There is also the methodology for financial control.

(E) Machine Learning: Machine learning is a part of artificial intelligence that creates systems that can learn from the data patterns without human intervention and provide predictions based on those data patterns. ML with the integration of cloud enhances the ML applications. This integration of ML with cloud is known as intelligent cloud. Since cloud is mainly used for networking, storage, and computing, the use of cloud machine learning increases the efficiency of both ML and cloud algorithms.

The authors in [22] presented the SmartScale scaling framework that is autonomous and uses both vertical and horizontal scaling so as to enhance

the resource usage and the cost of reconfiguration which is generated due to scaling. This methodology is based on a proactive model. The authors proposed an application aware scaling model to capture the merits of application scaling by modifying the total number of virtual machines versus changing the assigned resources to the VM. The authors present an effective mechanism that efficiently builds the scaling model. The smart scale algorithm is implemented through two phases to solve the scaling problem. In the first phase, an assumption is made that the demand of additional resources is very high and that each virtual machine runs efficiently. In the second phase, minimum number of instances are found such that the throughput equals ρ^1, that is, min_n, s.t. $\Sigma_{i=1}^{n} \rho_i >= \rho^1$. n is the optimal number if the throughput was increased by $\Sigma_{i=1}^{n} \rho_i$, whereas $n - 1$ is the optimal number of extra instances if the desired throughput increases $\Sigma_{i=1}^{n} \rho_i$. Since $\Sigma_{i=1}^{n} \rho_i >= \rho^1 > \Sigma_{i=1}^{n-1} \rho_i$, evaluation of both the options is done and the option with the least total cost is chosen.

(F) Pattern Matching: The pattern matching approach takes two inputs: the first is the cloud client usage of the past and the second is the current usage patterns. The historical data used for pattern matching is taken from the same application domain as the application that is trying to predict its own usage. Therefore, it is recommended to isolate the previous data on the basis of application domain before usages. The current usage pattern of the cloud user can be utilized to rectify patterns in the previous set which are similar to the present pattern itself. The patterns that are rectified should be independent on their scale, just on the relation among the elements of the rectified pattern and the pattern that is being searched for. The pattern which is very close will be interpolated by weight interpolation (the pattern that has been found similar to the current pattern will have a greater weight) and will have an estimation of the values which would follow after the current pattern. In a nutshell, the utilization of the cloud user can be forecasted by identifying usage patterns in the past or in other usage traces that are similar to the current usage pattern.

4.1.4 Model Solving Mechanisms

The basis of these approaches is modeling frameworks or the probabilistic model checking so as to know about the system's diverse behaviors and, hence, predict its future states such as probabilistic timed learning and Markov decision processes (MDPs). MDP presents a mathematical framework for the modeling of decision making in some conditions where the results are partially random and partially controlled by the decision maker. MDPs are used for the study of the optimization problems; the solution of which is provided by dynamic programming and reinforcement learning.

The authors in [23] proposed an extensible approach for the enforcement of elasticity by dynamic instantiation and online quantitative verification of Markov decision processes (MDP) by the use of probabilistic model checking. The decision-

making process uses the probabilistic models to analyze, drive, and specify cloud resource elasticity. The authors have used the probabilistic models to capture any kind of uncertainty in elasticity of the system. The input workload and the system state are being monitored. The decision policy is activated periodically and it is known as an elasticity step. The decision-making frequency is either equal to or less than the monitoring frequency. The elasticity step is divided into three subparts: dynamically instantiate the model on the basis of the input load and the log measurements; the model is being verified to reach elasticity decisions; elasticity actions are taken. The execution of the next elasticity step is suspended till the system stabilizes.

4.1.5 Combination of Both Reactive and Proactive

Some of the mechanisms use the features of both reactive and proactive approaches. Some of these mechanisms are as follows.

4.1.6 Control Theory

Control theory was introduced to automate web server system management, management of data centers/server clusters, storage in systems, and various other systems. Control systems are generally reactive. There are some proactive approximations, like the model predictive control, and even the combination of control systems with the predictive model. Figure 7 shows the block diagram of a feedback control system.

- *Open-loop controllers:* These controllers are also considered as non-feedback controllers. They are responsible to manage and process the inputs to the system. As the name suggests, these controllers do not have a feedback mechanism to check the efficiency of the system.
- *Feedback controllers:* These controllers have a feedback mechanism that monitors and rectifies any deviation of the outputs from the desired result and thus maintains the efficiency of the system.

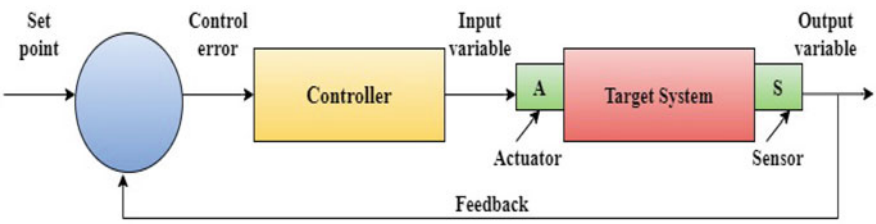

Fig. 7 Block diagram of a feedback control system

- *Feedback and Feedback-forward controllers:* These controllers anticipate the output errors in advance, predict the behavior of the system, and then accordingly take the actions before the occurrence of the errors.

The authors in [24] presented an online, software-only version of feedback-controlled virtual machine called self-tuning VM. The resource provisioning abstraction that is used in this model is a virtualized resource that can be configured by the users or the vendors. The VM reconfiguration means modifying different resources that are linked with a virtual machine, and throttling means reconfiguring on a provided share of the computation unit to a VM. After the deployment of an active VM on a host the feedback controller requests its share of the core of the system. The controller sustains the progress of the task that has been mentioned by the customer so as to meet the deadline. To accomplish an objective, three phases are performed that are application modeling, controller design, and actual control.

When a task executes (1) the sensors input in the task and informs the progress to the model estimator, (2) the model estimator runs the system by throttling to changing levels and (3) model estimation that relates the utilization of the resource to calculated progress. The control tuner utilizes the model for designing parameters for PI controller. After the completion of the tuning phase (4) the setting of the control parameters is done in the PI controller, which in uniform time intervals throttles to (5) measure the reference progress derived to achieve the deadline of the task. It uses the (6) error (reference measurement) in previous cycles to determine the (7) throttling at the next cycle. (8) A moving average filter is placed in between the controller and the job being sensed to smooth out the measurement noise.

4.1.7 Queuing Theory

To model the traditional servers and Internet applications, queuing theory is mostly used. This theory helps in the estimation of performance metrics like the queue length (the average waiting time for the request). Queuing theory makes reference to the mathematical study of queues, or waiting lines. In this model, the request of the client reaches the system at a mean arrival rate lambda and is queued until their processing is done. Many servers are available in this model that processes the request at mean service rate x.

The authors in [25] presented a technique based on dynamic provisioning for the multi-tier Internet applications that makes use of the queuing model to find the amount of resources that should be provided to each and every tier of the application and also uses a combination of both reactive and proactive methods to find when the provisioning of these resources would take place. The model presented takes input as the incoming request state and demand of the customer and then calculates the number of servers that are required at each tier so as to handle the request of the customers.

The authors modeled a multi-tier application as a network of queues where each queue resembles an application tier, and the queues from a tier feed into the

consecutive tier. Each of the allocated server is represented by queue. Queues that represent one tier feed into the queues of the consecutive tier. The initial step to solve the model is to find the individual server's capacity on the basis of the request rate. The next step calculates the number of servers that are needed at a tier to process a peak session rate. The author's modeled each server as G/G/1 queuing system. In this system, the request arrives at each of the server in a manner that the inter arrival times are derived from a constant, known distribution. Each of the requests brings certain amount of work that is to be done by the server. The time taken by the server to process the requests is called service time of the requests.

4.1.8 Reinforcement Learning (RL)

The reinforcement learning automatically makes the decisions for auto-scaling. This approach understands and automates decision-making and goal-oriented learning. This approach learns by direct interacting between the agent and its environment. The agent is the decision maker that learns from the experience to execute the best action for the environment. The agent is responsible for the addition or the removal of the resources with the application. This depends on the input workload, efficiency, or the other set of variables so as to minimize the application response time.

In [26] the authors proposed an approach based on reinforcement learning, namely VCONF, to make the VM configuration process autonomous. VCONF makes use of RL algorithms that are model based so as to address the issues related to adaptability and scalability in applying RL for the management of the system.

The design of VCONF is based on standalone daemon residing in the driver domain. It makes use of the control interface which is given by dom0 (it is a driver domain or privileged VM which manages other VMs and executes policies related to resource allocation) so as to control the configuration of each virtual machine. VCONF is responsible to manage the configurations of the virtual machine by monitoring the feedbacks that are performance based from each VM. Actions related to reconfigurations take place on a regular basis in time intervals that are predefined. VCONF queries the driver domain for the present state and processes valid actions. On the basis of the RL algorithm, VCONF chooses an action and sends it to dom0 for the reconfigurations of the VMs. At the end of each step, VCONF is responsible to collect the feedback based on the performance of each VM and then calculates the reward. The new sample of the immediate reward is calculated by RL algorithm and the update of configuration policies is done by VCONF accordingly.

4.1.9 Comparison Between Various Techniques of Reactive and Proactive Mode

This section compares the various techniques of reactive and proactive mode on the basis of different parameters. Table 1 shows the comparison between various techniques of reactive and proactive mode.

Table 1 Comparison between various techniques of reactive and proactive mode

Technique	Type	Approach used	Pricing	SLA (metrics used)
Time series analysis	Proactive	Scaling done by making predictions at fixed intervals using past results.	Pay as you go	Response time
Static threshold	Reactive	Scaling done on the basis of predefined thresholds.	Pay as you go	CPU load, memory usage
Dynamic threshold	Reactive	Scaling done on the basis of input workload.	Pay as you go	Adaptive usage threshold
Model solving mechanism	Proactive	It models framework to know system's behavior and predict future states.	Pay as you go	Response time
Control theory	Both (proactive and reactive)	It uses a decision-making module which is fed by the sensors which operates on data and provide the scaling solutions.	Pay as you go	Response time
Queuing theory	Both (proactive and reactive)	Calculates the arrival time of the requests and provides the resources accordingly.	Pay as you go	Arrival rate of requests, response time
Reinforcement learning	Both (proactive and reactive)	Takes decisions for scaling without the requirement of manual intervention.	Pay as you go	Response time

4.2 Programmable Mode

Other than the automatic mode, there is also a programmable mode which is similar to the manual mode as elasticity is done using the API calls. This automatic mode is implemented in some of the cloud systems such as Rackspace, Datapipe, and Microsoft Azure. In this mode, the responsibility of the user is to monitor the applications and the virtual environment. Since this mode violates the concept of automation, it cannot be considered as an elasticity mode.

5 Research Challenges

The following are the major research challenges in virtual machine scaling

(a) *Interoperability:* The resources like storage, compute, memory, etc. must be rented from various cloud vendors without any problem to provide accuracy and redundancy. According to the cost and technical skills, cloud vendors utilize their own techniques and services. Thus, using the services of multiple cloud vendors still remains an issue due to the lack of standardized APIs as each of the vendors has their own mechanism and how the customers and the applications interact with the infrastructure of the cloud.

(b) *Granularity:* Only a small set of services like Amazon instances are provided by the IaaS vendors, though some of the customers have various requirements. For example, some applications may require more computational power than the memory. The provisioning and de-provisioning of the resources must have a coordination among them. Apart from resource granularity, billing granularity is another major challenge. The customers are charged by the cloud vendors according to the consumption of resources per unit time. Most of the cloud vendors make use of hour as a minimum billing time unit.

(c) *Resource availability:* Resources like CPU, memory, etc. are provided by the cloud vendors in limited quantities. Due to this the scaling of these resources is constrained by the infrastructure's capacity of the cloud. Cloud vendors do not provide unlimited resources to its customers except big and popular vendors like Amazon and Microsoft. Also, the resource provisioning may be hampered due to high response time, fixed geographical locations, etc.

(d) *Hybrid solutions:* There are many advantages and disadvantages of the reactive and proactive techniques for virtual machine scaling. Hence a practical solution could use both proactive and reactive techniques and techniques like vertical and horizontal scaling.

(e) *Spin-up time:* It is the total time which is needed for the allocation of resources to the applications. Even though the spin-up time can be a few minutes, the main issue is that the users are charged from the time the request is made to scale the resources even when the resources are not acquired. The arrival of provisioning resources may take a lot of time, and there are costs that are being charged, that are not the same as real costs that match the resources that are being provided. The factors upon which the spin-up time may depend are cloud layer, virtual machine size, the availability of the resources in the region, and the scaling mechanism. The smaller the spin-up time, the more efficient the scaling mechanism is.

(f) *Prediction-estimation error:* Prediction in the changes of the workload is done by the proactive techniques which automatically take actions in advance to scale the resources. These techniques are used to handle the spin-up time issue but they can lead to errors such as prediction-estimation error. This is a major challenge in the scaling mechanism which can cause over provisioning or under provisioning of resources. Proactive techniques are very complex and are not

accurate in some of the situations and also depend on the behavior of the application or sudden hike or decrease in the workload. Since anticipating some of the applications is hard, the main objectives of the predictive techniques may not be fulfilled.

6 Conclusion

Cloud computing has revolutionized the IT sector completely and most of the MNCs are using this technology. The most important advantage of cloud computing is scalability which allows the scaling of resources as per the demand of the user, thus increasing the overall throughput. This chapter has discussed virtual machine scaling and why there is a need to implement this feature. It also elaborates the various modes and policies that help in the implementation of virtual machine scaling. Even though there are many advantages of using the scaling mechanism in the applications, there are some major drawbacks and challenges too. The most used scaling in articles is horizontal scaling. OS and cloud architecture support horizontal scaling but vertical scaling is easy and gives better cost benefit in cloud infrastructure. OS and hypervisors should be improved for vertical scaling too. Cost and energy effective allocation of virtual machines and consolidated migration are future areas for research. The workload predictors that are used nowadays are considering the historic workload. It is very difficult to predict flash workload from the historic workload. The growth in the deep learning techniques and online data mining, real-time data can be used to tackle the problem of sudden burst in the data center. To handle the flash workload, categorization of the applications is also very helpful. Spot instances can be more useful in terms of cost optimization to tackle flash workload. Also, the current research focus is mainly on the cost optimization and QoS requirement.

References

1. *Oracle user guide*. https://docs.oracle.com/cd/E2730001/E27309/html. Last accessed 10 July 2020.
2. Lakew, E., Klein, C., Hernandez-Rodriguez, F., & Elmroth, E. (2014). Towards faster response time models for vertical elasticity. In *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)* (pp. 560–565).
3. Spinner, S., Kounev, S., Zhu, X., Lu, L., Uysal, M., Holler, A., et al. (2014). Runtime vertical scaling of virtualized applications via online model estimation. In *IEEE Eighth International Conference on Self-Adaptive and Self-Organizing Systems (SASO)* (pp. 157–166).
4. Wang, Y., Tan, C. C., & Mi, N. (2014). Using elasticity to improve inline data deduplication storage systems. In *Proceedings of the 2014 IEEE International Conference on Cloud Computing, CLOUD'14* (pp. 785–792). Washington: IEEE Computer Society.
5. Molt' o, G., Caballer, M., Romero, E., & de Alfonso, C. (2013). Elastic memory management of virtualized infrastructures for applications with dynamic memory requirements. *Procedia Computer Science, 18*, 159–168.

6. Farokhi, S., Lakew, E., Klein, C., Brandic, I., & Elmroth, E. (2015). Coordinating CPU and memory elasticity controllers to meet service response time constraints. In *International Conference on Cloud and Autonomic Computing (ICCAC)* (pp. 69–80).
7. Dawoud, W., Takouna, I., & Meinel, C. (2012). Elastic virtual machine for fine grained cloud resource provisioning. In *Global Trends in Computing and Communication Systems* (pp. 11–25). Berlin: Springer.
8. Lu, L., Zhu, X., Griffith, R., Padala, P., Parikh, A., Shah, P., et al. (2014). Application-driven dynamic vertical scaling of virtual machines in resource pools. In *Network Operations and Management Symposium (NOMS)* (pp. 1–9). Piscataway: IEEE.
9. da Silva Dias, A., Nakamura, L. H. V., Estrella, J. C., Santana, R. H. C., & Santana, M. J. (2014). Providing IaaS resources automatically through prediction and monitoring approaches. In *IEEE Symposium on Computers and Communication (ISCC)* (pp. 1–7).
10. Bajoria, V., Katal, A., & Agarwal, Y. (2018). An energy aware policy for mapping and migrating virtual machines in cloud environment using migration factor. In *8th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, Noida (pp. 1–5).
11. Tailwal, R., & Katal, A. (2017). An optimized time series based two phase strategy pre-copy algorithm for live virtual machine migration. *International Journal of Engineering Research and Technology, 6*(01). ISSN: 2278-0181
12. Mohan Murthy, M., Sanjay, H., & Anand, J. (2014). Threshold based auto scaling of virtual machines in cloud environment. In *11th IFIP International Conference on Network and Parallel Computing (NPC)*, Ilan (pp. 247–256)
13. Singh, B. K., Sharma, D. P., Alemu, M., & Adane, A. (2020). Cloud-based outsourcing framework for efficient IT project management practices. *International Journal of Advanced Computer Science and Applications, 11*(9), 114–152.
14. Han, R., Ghanem, M. M., Guo, L., Guo, Y., & Osmond, M. (2014). Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. *Future Generation Computer Systems, 32*, 82–98.
15. Beloglazov, A., & Buyya, R. (2010). Adaptive threshold based approach for energy-efficient consolidation of virtual machines in cloud data centers. In *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, MGC'10* (pp. 4:1–4:6). New York: ACM.
16. Beloglazov, A., & Buyya, R. (2010). Energy efficient resource management in virtualized cloud data centers. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, Melbourne, VIC (pp. 826–831).
17. Han, R., Guo, L., Ghanem, M. M., & Guo, Y. (2012). Lightweight resource scaling for cloud applications. In *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2012)*, Ottawa (pp. 644–651).
18. Miguel-Alonso, T. L.-B. J., & Lozano, J. A. (2014). A review of autoscaling techniques for elastic applications in cloud environments. *Journal Grid Computing, 12*, 559–592.
19. Huang, J., Li, C., & Yu, J. (2012). Resource prediction based on double exponential smoothing in cloud computing. In *2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, Yichang (pp. 2056–2060)
20. Fernandez, H., Pierre, G., & Kielmann, T. (2014). Autoscaling web applications in heterogeneous cloud infrastructures. In *IEEE International Conference on Cloud Engineering*, Boston (pp. 195–204)
21. da Silva Dias, A., Nakamura, L. H. V., Estrella, J. C., Santana, R. H. C., & Santana, M. J. (2014). Providing IaaS resources automatically through prediction and monitoring approaches. In *IEEE Symposium on Computers and Communications (ISCC)*, Funchal (pp. 1–7).
22. Dutta, S., Gera, S., Verma, A., & Viswanathan, B. (2012). SmartScale: Automatic application scaling in enterprise clouds. In *IEEE Fifth International Conference on Cloud Computing*, Honolulu (pp. 221–228).

23. Naskos, A., Stachtiari, E., Gounaris, A., Katsaros, P., Tsoumakos, D., Konstantinou, I., et al. (2015). Dependable horizontal scaling based on probabilistic model checking. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen (pp. 31–40).
24. Park, S., & Humphrey, M. (2009). Self-tuning virtual machines for predictable eScience. In *9th IEEE/ACM International Symposium on Cluster Computing and the Grid* (pp. 356–363).
25. Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., & Wood, T. (2008). Agile dynamic provisioning of multi-tier internet applications. *ACM Transactions on Autonomous and Adaptive Systems, 3*, 1:1–1:39.
26. Rao, J., Bu, X., Xu, C.-Z., Wang, L., & Yin, G. (2009). VCONF: A reinforcement learning approach to virtual machines auto-configuration. In *Proceedings of the 6th International Conference on Autonomic Computing, ICAC'09* (pp. 137–146).

Autonomic Resource Management in a Cloud-Based Infrastructure Environment

Bhupesh Kumar Singh, Mohammad Danish, Tanupriya Choudhury, and Durga Prasad Sharma

1 Introduction

The prime focus and objective of autonomic computing refers to the self-management capabilities in a distributed system environment for handling computing resources in such an adaptable manner that it operates dynamically while hiding the internal complexities from user base. The concept was formalized by IBM in 2001 aimed to design and develop system infrastructure, keeping an eye on future requirements of complex computing environment and its management. Autonomic computing is modeled to plan and design in making adaptive decision-making by regulating some high policies [1, 2]. It continuously checks and optimizes its status with respect to resource handling by tuning itself from time to time. Such environmental approach is proved to be quite intellectual in absence of some caretaker. The role of manager is performed efficiently. It is also helpful in managing the local and global environment simultaneously. Such architecture is commonly referred to as monitor, plan, and execute.

Numerous architectural frameworks based on self-regulating autonomic components have been recently proposed by several research scholars. Such autonomic systems majorly involve the mobile agents actively involved in communication mechanisms. Autonomous computing, as suggested by Paul Horn of IBM in

B. K. Singh (✉) · D. P. Sharma
Computing and Software Engineering, Arba Minch University, Arba Minch, Ethiopia
e-mail: dr.bhupeshkumarsingh@amu.edu.et; sharma.dp@amu.edu.et

M. Danish
Computer Science and Engineering, Al-Falah University, Faridabad, India

T. Choudhury
Department of Informatics, School of Computer Science, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_18

Fig. 1 Autonomic element structure

2001, shared the dream of automated control of all computing systems [3]. This refers to the self-management characteristics of distributed computing systems that detect and recognize system changes and take necessary corrective action entirely automatically, with near to zero human intervention. The main advantage is a dramatic reduction in the inherent complexity of computer systems and making computer more accessible and user-friendly. The dream is to make computer systems self-configuring, self-optimizing, self-protective, and self-curing (Fig. 1).

2 Literature Review

2.1 E-commerce J2EE Applications

Electronic commerce (EC) depends on the Internet of the new plan of action, advancement reality restriction, totally changed the customary business model, to make new business openings. As per the meaning of the World Trade Organization electronic business unique report, electronic trade is the creation, the board, of deals and appropriation exercises through the PC arrange, and it just not alludes to exchanges depending on the exercises of the Internet (the idea of the electronic business framework), likewise alludes to all the utilization of electronic data innovation to take care of issues rapidly, lessen costs, and grow exposure include esteem and make business open doors for business exercises, including through

the system from crude materials acquirement to inquiry, creation, stockpiling and electronic installment, client care, and a progression of business exercises [4, 5].

The utilization of online business endeavors is not just through direct contact with a huge number of new clients of the system but to manage them, profoundly smoothing out the business joins, lessen working expenses, improve operational proficiency, and increment venture benefit. It additionally can speak with other exchange accomplices everywhere throughout the world whenever required, upgrade the collaboration among ventures, and improve the seriousness of items. Contrasted and the customary plan of action, Internet business has the accompanying attributes.

1. *Diminished flow*: Electronic business does not require wholesalers, stores, and shopping centers; clients, through the system, straightforwardly arrange items from the shippers.
2. *Spare shopping time*: Clients can buy fulfilled products through the system without going out.
3. *To quicken the course of assets*: Online business turnover without between outside of the bank clients, wholesalers, shopping centers, and through the system inside the ledger straightforwardly, enormously quickens the capital turnover rate, and in addition lessens the business questions.
4. *Upgrade the correspondence among clients and makers*: Clients can communicate their own prerequisites through the system, request items, makers can rapidly comprehend the client needs, and dodge misuse of item creation.
5. *Invigorate the participation and rivalry among undertakings*: ventures can comprehend the contenders' item execution and value, deal volume, and other data through servers.

2.2 Hosting Data Center Environment

Innovation is on the ascent and is bringing about increasingly steady and powerful stages and programming arrangements. Such is the situation with Cloud condition. Earlier, having a server facilitated on the cloud was intended to store server information on a virtualized stage that can be obtained whenever, according to client necessity. Moving information into the cloud was to have more extra room in a much secure condition [6].

Be that as it may, in time, the thought extended, and progressively novel arrangements of the cloud condition developed. Private cloud is one such arrangement that came as a need for making sure about potential data in increasingly controlled cloud situations. The intention was to give additional convincing advantages through improved security and expanded adaptability to dispense with inconveniences encompassing server the board. Henceforth, associations were seen contributing their time, exertion, and cost to acquire modified answer for separate organizations [7].

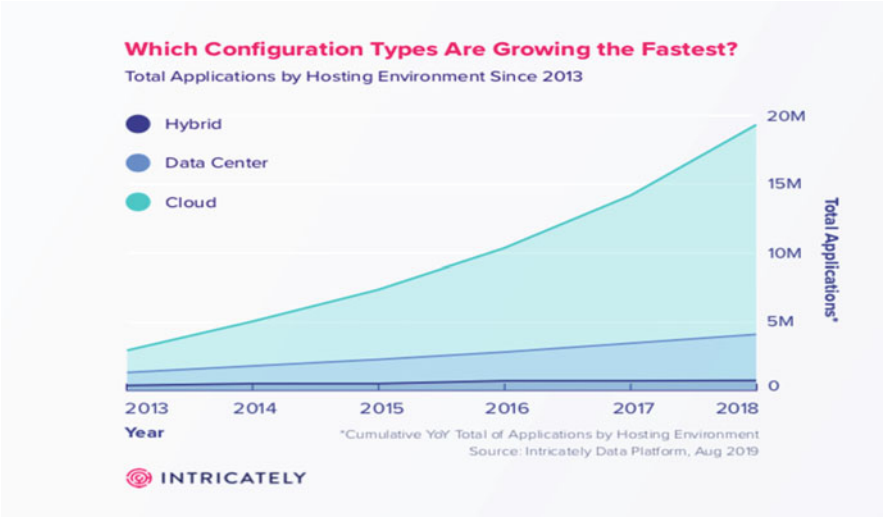

Fig. 2 Intricate reports about data-centered applications

At the point when we talk about private clouds, they are generally classified into two classes. As per intricate analysis, the following figure depicts the rise of hosting applications in the data center environment (Fig. 2).

Hosted private cloud solution: Today, when you search Google for what are your best options to make the best cloud possible for your company, you are given a rundown of cloud-based facilitating arrangements which are allowed by private cloud arrangements. Essentially, a significant portion of these cloud providers sell cloud services on their own servers.

Commercial cloud service on-premises: While secretly facilitated mists are at a server area, an on-premise cloud will furnish you with the choice to have a cloud domain inside. Such cloud arrangements require an inner server to have your own cloud server. They are kept an eye on by the association's IT division.

One of the main advantages of an on-premise cloud arrangement is that you gain total power on parts of security, versatility, and configurability of your servers. Be that as it may, you are restricted to the choice of adaptability relying upon the size of your server environment [8].

3 Cloud Computing Methodology

Cloud computing is a revolutionary technology in the IT industry. It provides an independent platform to every developer and software enthusiast, where there are no geographical boundaries and no resource access limitations. It provides user an independent choice for selection of software services, platform, and a true

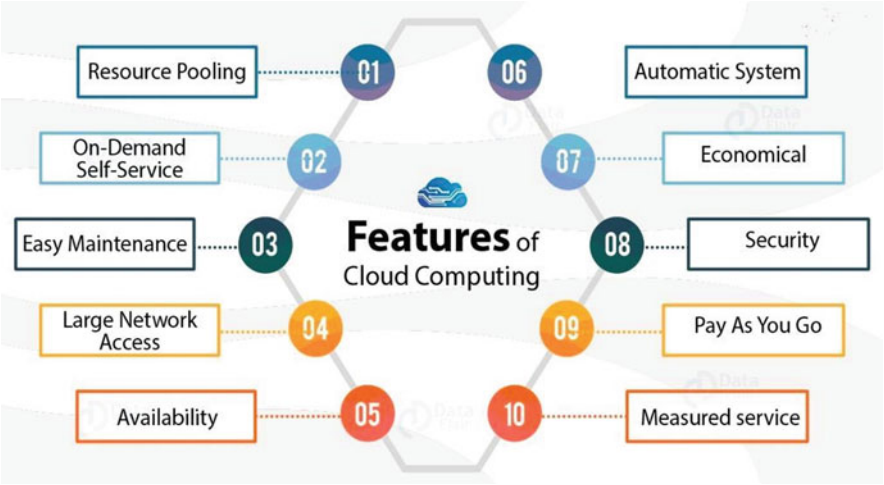

Fig. 3 Data flair cloud computing services architecture

choice for infrastructure selection. Recently, there has been enormous growth in this sector. Some major players are Amazon Web Services (AWS), Google Cloud, and Microsoft Azure. It usually provides three different types of services that are possibly provided through some remote client technologies. Small- and mid-scale companies are their major potential clients, as they could not set up sufficient infrastructure required for some complex software services since they are usually short of funds. In this way, these platforms of clouds are truly capable of serving their needs in limited budget-based infrastructure. Companies can subscribe to their services either on a daily basis or on a monthly basis. Even there is great extent of flexibility in terms of payment plans [9]. Some services can also be rented on hourly basis. Using such platforms are proved to be a boon for small- and mid-scale companies, as they are now able to defend in global market competition (Fig. 3).

3.1 Public Cloud versus Private Cloud

Generally, cloud is categorized into two types of major domain, i.e., public and private. In a public cloud-based Environment, the company offers a variety of services all over Internet. Data sharing is popular among corporate or individuals, and all of them share common platform for gaining benefits across the network. Services are subjected to cost or free depending upon the environment and usage of client. Companies like Amazon and Google provide services free for limited time duration with limited bandwidth of resources [10]. In private clouds, the system is quite similar except that these services are isolated for individual companies with prior approval capabilities.

The following features can be compared in terms of public and private clouds

Accessibility: Public clouds are available to everyone, while accessibility of private clouds is limited to a particular user base. Public clouds surrender their data to replicated locations for the ease of availability, while private clouds are generally accessible to a single node location.

Security: When it comes to security, cloud companies promise to provide a high-level security for user data protection. Since public cloud is available to everyone, and private cloud accessibility is limited to few entities, people generally prefer private cloud as a strong entity for security.

3.2 Selection Between a Public and a Private Cloud: A Case Study

Users often seem to be quite confused about the selection between public and private clouds. One could opt for any option, but certain parameters must be evaluated before deploying their business over cloud. Public cloud infrastructure is easy to implement but requires a lot of investment to proceed. On the contrary, private cloud investment is less than public cloud-based infrastructure but considered to be more secure. Cost of private cloud could not be justified by limited resources usage. While public cloud has huge customer base and so do have their business dimensions. One can earn by renting the same infrastructure to different customers. In a private cloud, due to company policies, they are not free to rent out their infrastructure to outside customers. Big business giants like Netflix, Dropbox, and Instagram use public clouds.

A Private cloud [11] is much more efficient due to its dedicated hardware available for company services. Uptime of such infra is always better than public cloud. Limited but better use of resources makes it more useful. Private cloud provides better security, dependable availability, and superior level of control mechanism in comparison to public cloud. Customizing services and management control are well suited in private clouds. They are suitable to independent software vendors which are likely to improve their efficiency with respect to time and customer base.

3.3 Private Cloud

Private cloud is a kind of cloud environment, where cloud computing is overseen by inside Information Technology. For instance, in private IaaS, interior tasks introduce and deals with the cloud the board stage for its framework. In Private PaaS/Enterprise PaaS, inner activities introduce and deal with the PaaS programming on either inward or open IaaS framework. Private cloud incorporates

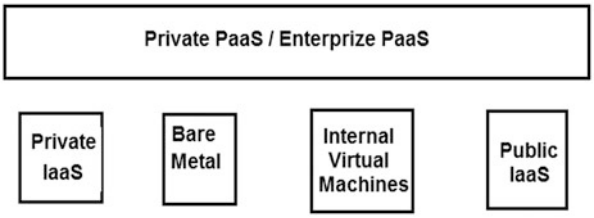

Fig. 4 Private cloud illustration in public enterprise

private infrastructure as a service (private IaaS) and private stage as assistance (private PaaS). In those cases, the open cloud is detached in the physical server and intelligent systems administration layer. Instances of such open mists that have embraced the private cloud name incorporate AWS Virtual Private Cloud.

Sun et al. [12] Prior to 2015, most private cloud endeavors have been only centered on private IaaS. As indicated by Gartner's Thomas Bittman, 95% of Private Clouds have neglected to satisfy the first guarantee [7]. Gartner's recommendation for associations in 2015 is to begin taking a gander at private PaaS. As indicated by Gartner for Technical Professionals' 2015 Planning Guide for Cloud Computing [13], "Private PaaS (Enterprise PaaS) improves designer efficiency, decreases operational exertion, and increments facilitating thickness. This incentive is unreasonably convincing for enormous ventures to disregard" (Fig. 4).

Highlights

1. Private cloud is of two types—private infrastructure as a service (PIaaS) and private platform as a service (PPaaS).
2. PPaaS is regularly alluded to as Enterprise–PaaS.
3. PPaaS/Enterprise–PaaS can be introduced in both inward and outer situations. For instance, it very well may be introduced on AWS and a client's inner server farm exposed metal, virtualized, and private IaaS.
4. Numerous associations are utilizing private–PaaS/Enterprise–PaaS to enhance or pull together private cloud systems.

3.4 Uses of Cloud Computing

The future utilization of distributed computing is just beginning to get a handle on. When the massive possibilities of distributed computing are worked out, ideas are being focused on it. Distributed computing is likely to change the way we use it to function at the individual and corporate levels [14].

For enterprises: The cloud can possibly change activities for companies just as it can cut costs. Workplaces operating PC systems would no longer have to handle the establishment of programming for every computer, just like licenses. The role

of corporate-level distributed computing may be either for the home-based tasks or as a programming or administration tool that the company produces for the general society.

Mobility: The portability it provides to the recreational customer as well as to the corporate and enterprise customer is one of the other clearest employments in distributed computing. A decent number of us are now acquainted with certain distributed computing organizations, close to Google Docs, or even e-mail.

3.5 Data Center

Each association requires a data center, independent of its size or industry. A data center is generally a physical office that organizations use to store their data just as different applications, which are indispensable to their working [9]. And keeping in mind that a data center is believed to be a certain something, in all actuality, it is regularly made out of specialized hardware relying upon what requires to be put away—it can run from switches and security gadgets to capacity frameworks and application conveyance controllers. To keep all the equipment and programming refreshed and running, a data center additionally requires a lot of foundation. These offices can incorporate ventilation and cooling frameworks, uninterruptible power supplies, and reinforcement generators (Fig. 5).

A cloud data center is fundamentally unique in relation to a conventional data center; there is nothing common between these two registering frameworks other than the way that the two of them store information. A cloud data center is not genuinely situated in a specific association's office—it is all on the web! At the point when your information is put away on cloud servers, it consequently gets divided and copied across different areas for secure capacity. In the event that there are any disappointments, your cloud administration supplier will ensure that there is always a backup for you.

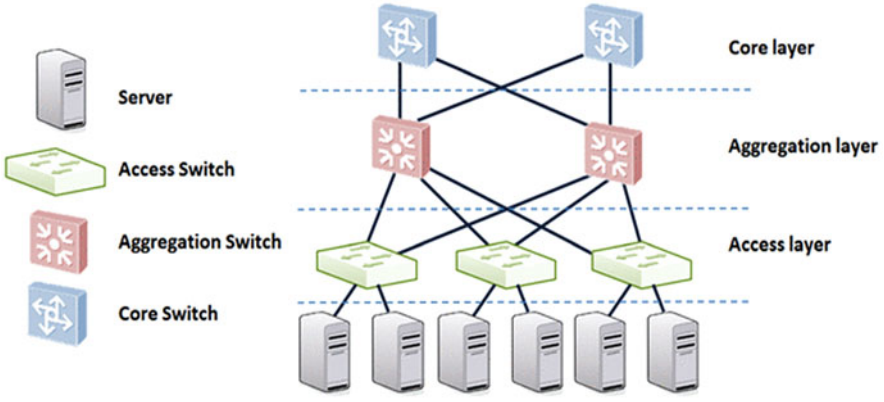

Fig. 5 Data center architecture [5]

Cost: With a conventional data center, you should make different buys, including the server equipment and the systems administration equipment. In addition to the fact that this is a disservice in itself, you will likewise need to supplant this equipment as it ages and gets obsolete. In addition, notwithstanding the expense of buying hardware, you will likewise need to recruit staff to administer its activities.

At the point when you have your information on cloud servers, you are basically utilizing another person's equipment and foundation, so it sets aside a ton of monetary assets that may have been spent while setting up a customary data center. Likewise, it deals with different incidental variables identifying with support and in this way helping you improve your assets better.

Accessibility: A conventional data center permits you adaptability as far as the gear you pick, so you know precisely what programming and equipment you are utilizing. This encourages later customizations since there is no one else in the condition and you can cause changes as you require.

With cloud facilitating, availability may turn into an issue. In the event that anytime you don't have an Internet association, at that point your remote information will get out of reach, which may be an issue for a few. In any case, sensibly, such occurrences of no Internet network might be not very many and far between, so this perspective shouldn't be an over-the-top issue. In addition, you may need to contact your cloud administrations supplier if there is an issue at the backend—however, this also should not take long to get settled.

Security: Traditional data centers must be secured in a traditional way: You will have to employ security personnel to ensure that your data is safe [9]. The downside here is that you will have absolute control of your data and facilities, making it safer to a large degree. Only people of confidence will be able to access your program.

Cloud hosting can be more dangerous, at least in theory, since anyone with an Internet connection can access your data. In fact, however, most cloud service [9, 15–21] providers do not leave a stone unturned to ensure the protection of your data. We hire professional personnel to ensure that the appropriate security measures are in place to ensure that the data are still in safe hands.

Scalability: Building your own framework without any preparation takes a ton of contribution both in money-related and human terms. In addition to other things, you should regulate your own support just as an organization, and therefore, it takes a long effort to get off the ground. Setting up a conventional data center is an expensive issue. Further, if you wish to scale up your data center, you may need to dish out additional cash yet reluctantly.

With cloud facilitating, notwithstanding, there are no upfront expenses as far as buying equipment, and this prompts reserve funds that can later be utilized to scale up. Cloud service [9, 15–21] organizations have numerous adaptable plans to suit your requirements, and you can purchase more storage as and when you are prepared for it. You can also reduce the amount of storage you have if that is your requirement.

3.6 IaaS in Public Domain

Cloud system administrations, referred to as Infrastructure as a Service (IaaS), are self-administered models for accessing, testing, and managing remote datacenter frameworks, such as process (virtualized), organization, and administration (such as firewalls). Customers should purchase IaaS based on consumption, such as electricity or other service billing, rather than purchasing equipment altogether [22]. Service providers are handling all that includes virtualization, servers, hard drives, and network maintenance. At present, numerous IaaS suppliers often provide databases, lines of information, and various administrations throughout the virtualization layer. When released into the market, consumers are responsible for new launches. Examples of IaaS include Amazon Web Services (AWS), Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE).

3.7 PaaS (Platform as a Service)

PaaS is a software that makes centralized control of an application development platform using cloud computing. This involves not only remote use of software but also a full application creation and delivery framework. It helps the developer to design, test, and execute their applications on the same platform that their end-user clients would use to run the program.

PaaS benefits: The benefits of PaaS technology are developers' ability to design, test, and deploy their applications within a single, streamlined environment. Distribution is done on the same platform it is built on, avoiding client device and hardware conflicts. A unified platform environment also frees developers from the need to adapt their applications to operate on various operating systems and hardware.

3.8 SaaS (Software as a Service)

On-demand applications refer to the computer programs delivered as a service over the Internet. This type of program is also known as on-demand applications. For the market, Apps on demand is a revolutionary model that provides the primary benefit of decreased operating and capital costs for business information technology services. When an organization chooses to integrate on-demand software as a solution to its business software needs, it will make internal IT services more effective [23].

SaaS characteristics: There are some main features at the core of the on-demand software or SaaS product distribution model. Such features and how they vary from the conventional on-site development model are as follows:

Centralized hosting/delivery: This aspect of on-demand applications varies from the conventional software delivery model that involves an additional operating overhead of delivering software to users across various distribution channels.

Systematic delivery system: This feature of on-demand applications varies from the conventional software distribution model that involves specific software packages for various operating systems and platforms. For an on-demand application, all SaaS solutions operate on a common platform with a standard style of the application interface.

How cloud storage works: Online storage works by allowing users to access and upload data from any computer they choose, such as a laptop, tablet, or smartphone, through an Internet connection service. Cloud storage users can also edit documents at the same time as other users, making working away from the office easier. The cloud storage rates differ depending on different needs. For an individual user, you will usually get initial amounts of free cloud storage—like Apple iCloud's 5 GB, which not long ago struggled with some well-publicized cloud protection issues. You will pay an additional charge for the building. Popular pricing models, depending on the product you use, offer monthly or annual rates [12, 24].

How cloud environment works: To grasp the workings of a cloud network, splitting it into two parts is easier: front and back. They are linked through a network, usually over the Internet. The front end is the machine's User or Server hand. The back end of the system is the "Internet" part (Fig. 6).

The front end is the company's systems or data nodes. The application is also essential for access to the platform for cloud computing. Not all cloud computing applications require the same user interface. Cloud computing, from data processing to video games, can potentially include any computer program. Usually, every system will have its own dedicated server.

Evaluating the risk of cloud computing: Wide organizations and hundreds of digital storage devices are also required. Cloud computing providers allow at least twice as many storage devices as possible to store information about their clients. That is because structures like these sometimes break down. The cloud software allows for the retrieval of copies of company information on other devices. This method of copying backup data is called data redundancy.

3.9 Challenges in Cloud Computing

Cloud computing is used to allow global access to common pools of resources, such as infrastructure, applications, data, servers, and computer networks. This is done either on a third-party server located in a data center or on a privately owned cloud. This makes the data access contrivance more secure and effective, with a nominal administration effort. Since cloud technology relies on the distribution of resources to achieve efficiency and economies of scale, similar to utility, it is also relatively cost-effective, making it an option for many small businesses and companies [23].

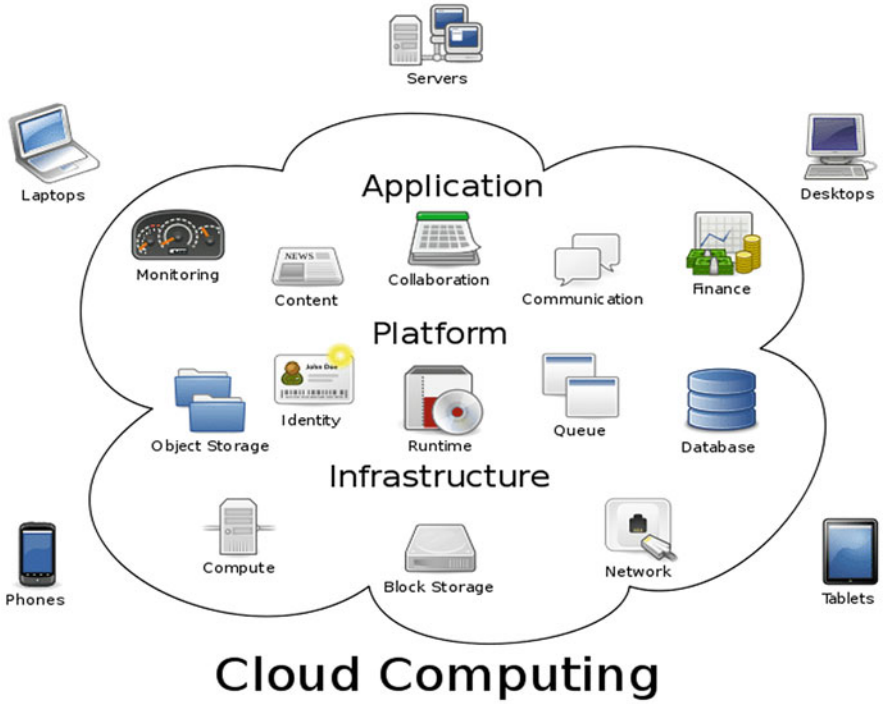

Fig. 6 Fast metric cloud execution environment

Cost cloud: Storage itself is inexpensive, but adapting the platform to the needs of the business can be costly. Organizations will save money on network maintenance, operations, and acquisitions. But they do need to invest in additional bandwidth, and the lack of routine control over an indefinitely scalable computing network will increase costs.

Service provider reliability: The capability and ability of the technical service provider are as critical as the quality. The service provider must be available when you need it. Sustainability and credibility are of great concern. Make sure you understand the techniques by which the organization tests its programs and defends statements of reliability.

Downtime is a significant shortcoming in cloud technology. No seller can guarantee a platform that is free from potential downtime. Cloud technology makes small businesses dependent on their access, so businesses with an untrustworthy Internet connection are likely to want to think twice before implementing cloud computing.

Credential security: Industrious monitoring of the login plays a key role in cloud security. The more users you have access to your cloud account, though, the less secure it is. Anyone who knows about your passwords may have access to the details you hold there.

Data privacy: Sensitive and personal data contained in the cloud should be defined as being exclusively for internal use and not shared with third parties. Businesses must have a plan for managing data that they obtain in a secure and effective manner.

4 Autonomic Computing

4.1 Definition

Autonomous Computing System: Autonomous computation is the ability of a computer to handle itself efficiently by means of advanced technology that expands computational capabilities and minimizes the time taken by technical practitioners to solve device issues and other maintenance, such as software upgrades.

4.2 Technical View

Autonomous computing aims to solve complexity by using technology to control data. Autonomous technologies predict the needs of the IT system and solve issues with minimal human involvement. As a result, IT professionals will concentrate on projects with greater business importance.

4.3 Self-Management Attributes of System Components

System modules and hardware (such as storage cabinets, desktop computers, and servers) may provide built-in control loop features in a self-managing, autonomous system. These control loops are made up of the same fundamental components.

4.4 Self-Configuration

This will quickly respond to changing conditions. Self-configuring modules respond seamlessly to external changes, using IT technical policies. Such modifications can include the addition of new modules or the replacement of old ones or dramatic system functionality improvements. Complex transformation aids in maintaining the IT network's sustained intensity and performance, resulting in market growth and flexibility.

4.5 Self-Healing

Self-healing components can detect system defects and implement policy-based corrective action without affecting the IT environment. Corrective intervention may require a substance altering its own state or effecting environmental changes in other components. The IT infrastructure as a whole is getting more robust, as it is less likely to malfunction daily operations.

4.6 Self-Optimizing

This will dynamically track and fine-tune services. Components that are self-optimizing will settle to fulfill end-user or business needs. Tuning activities may involve reallocating capital to maximize optimal utilization, such as reacting to rapidly shifting workloads, or ensuring that individual company transactions can be performed in a timely manner. Self-optimization helps provide a high quality of support for both end users of the network and consumers of a company (Fig. 7).

4.7 Self-Secure

Autonomous computation can predict, track, recognize, and defend against attacks from anywhere. Self-protecting modules can track aggressive actions as they emerge and take corrective measures to make themselves less vulnerable. Hostile activities can include unauthorized entry and use, contamination and propagation of viruses, and attacks of denial of service. Self-protecting technologies allow organizations to reliably implement privacy and security policies.

5 An Autonomic Engine for Managing Clouds

5.1 Autonomic Engine Introduction

Autonomous operating network of cloud and infrastructure. It is based on a shared system of cooperation and embraces extremely heterogeneous and dynamic cloud architecture and convergence between public/private cloud and autonomous cloud bursts. It is built on peer-to-peer substrates and can extend data centers, networks, and clouds across enterprises. In order to provide web infrastructure, tools can be assimilated to peer-to-peer overlays on demand and the move. Conceptually, the engine consists of a code layer, a software layer, and an interface layer; the following diagrammatic model defines those layers in more detail [25] (Fig. 8).

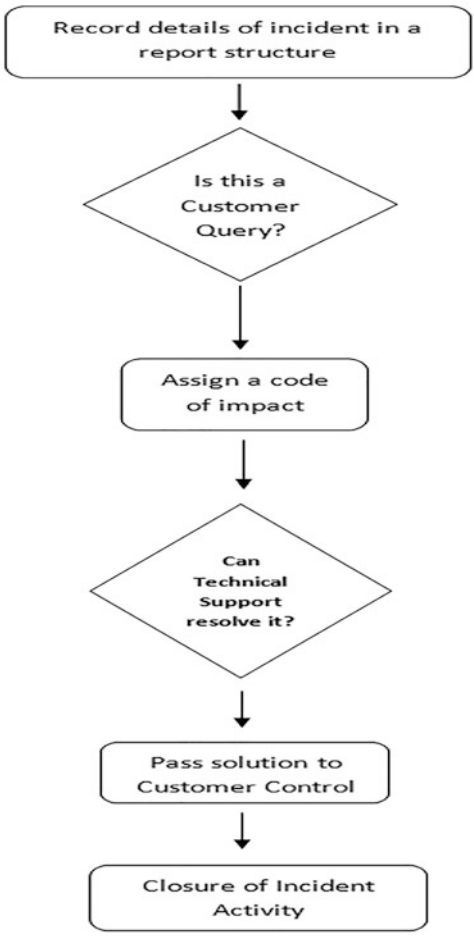

Fig. 7 A normal IT process layout

Autonomous engine adapts the squid knowledge discovery scheme to map information space to the complex range of peer nodes in a deterministic way. The resultant configuration is a localization that retains the distributed semantic hash table at the top of a hierarchical overlay which is self-organizing. This preserves localization of information and guarantees that domain-based requests are handled at a minimal rate, using robust query descriptors in the form of keywords, partial keywords, and wildcards. Builds an abstraction of tuple-based coordination space using Squid, which can be accessed associatively by all device peers without needing tuple and host identifier position details.

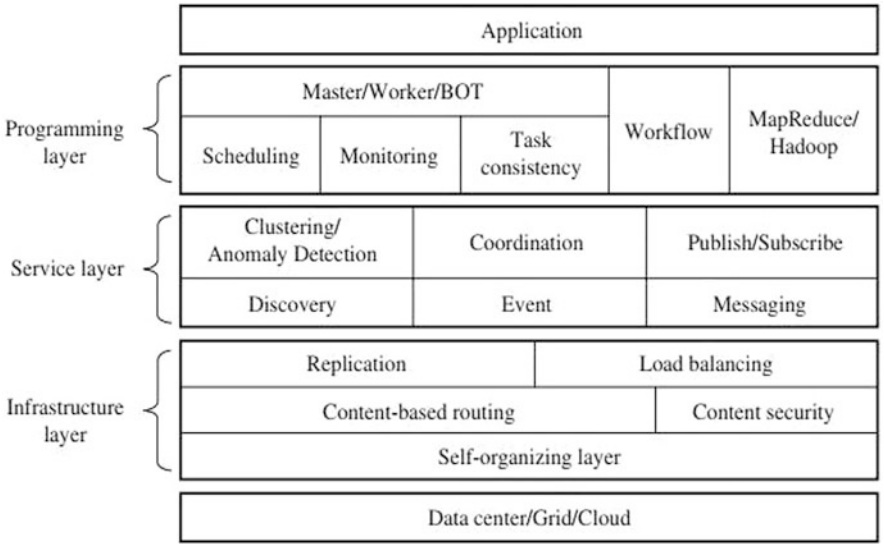

Fig. 8 An autonomic cloud engine layered execution structure

5.2 Gathering Resource Metadata

In order to make the predictions, the Autonomous Engine requires details. Tools for the underlying cloud/grid network. These resource metadata can be divided into two macro areas: system configurations and system benchmarks. Table 1 displays the present arrangement of such metadata, as used by our autonomous program.

System configuration is a set of resource parameters that define how to configure a resource. The basic item (system, node, and network) has been configured. This includes, for example, the frequency and micro-architecture of the computing components or the latencies and the bandwidth of the networks.

The key criteria for the compilation of resource metadata are:

1. Flexible structure: It is not easy, nor desirable, to impose a rigid structure.
2. Mining capabilities some cloud/grid systems explicitly disclose the resource metadata that is not necessarily accurate. And when some metadata is available, they may not be detailed enough to serve as the basis for configuring the simulation. After all, the ultimate aim of a cloud is to mask the nature of the system [26].

The underlying infrastructure and methods, therefore, must be as general as possible to extract this information directly from the device, be planned [27, 28].

Table 1 Resource metadata as gathered by the Knowledge Module

System configuration		
Scope	Name	Description
System	No. of nodes	Number of nodes that are available to the system
System	Networks	Available networks (e.g., 10Gb Ethernet, Infiniband)
Node	Hypervisor	Virtual machine monitor of a node (e.g., Xen, KVM, VirtualBox, or none for physical nodes)
Node	No. of CPU	Number of CPUs, number of cores per CPU
Node	Amount of memory	Amount of main memory, amount and configuration of cache memories
Node	Network interface cards	Per-node interfaces to the system networks
System benchmark figures		
Benchmark class		Description
CPU		Raw computational speed benchmarks (includes FLOPS and MIPS measurements)
Memory		Sequential/random read/write access, cache access times
Network		Includes latency and bandwidth benchmarks
Disk		File creation/read/write benchmarks

5.3 Requirements for Monitoring

To evaluate at least two metrics, it is important to provide a monitoring infrastructure: system load and network availability. Device load applies to all the measurements that can be gathered to calculate the load from the different services. Such data are fed to the simulator to contextualize the process of forecasting results to a particular load on the system. Availability of the device needs the resource existence details. Periodic checks are necessary to ensure that a whole virtual machine or a single program is still up and running [29, 30].

6 Experimental Analysis and Results

We specified the average response quality (ARQ) metrics for the comparative performance assessment of the autonomic service styles as in the following equation.

$$\text{ARQ} = \frac{\sum_{i\text{Service count}=1} \left(\text{Service configuration Time} + \sum_{\text{Service Request Count}\, j=0} \text{Issue Request Solving Time}\right)}{\text{Total Service Request Count}}$$

The ARQ measures the autonomic efficiency of a collection of self- resources for a growing number of requests for self-management. A lower ARQ value means that autonomic performance is higher [31]. For experimental analysis, we generate a request a pool of five autonomic machines in three straight scenarios:

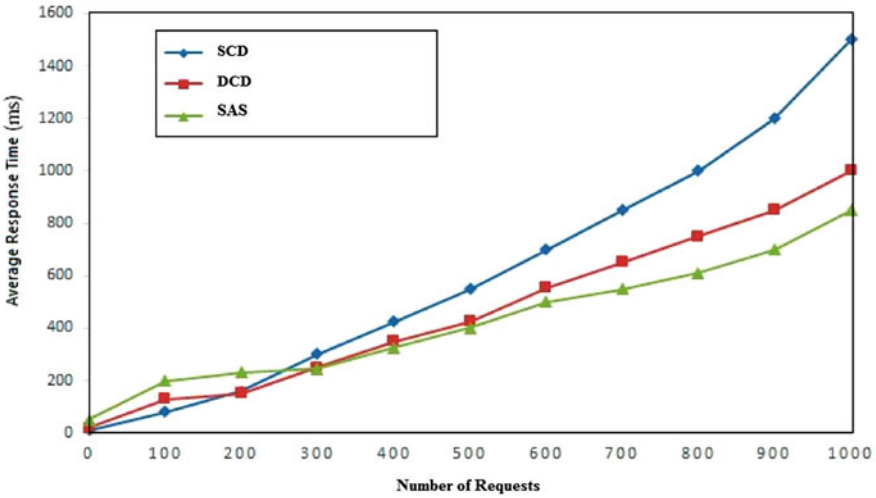

Fig. 9 Self-request evaluation

1. Statically configured local AMs;
2. Dynamically configured AMs from a server machine;
3. Runtime registered AMs from the Autonomic cloud.

The results of the evaluation of ARQ using the above equation are shown in the following figure (Fig. 9).

It shows that autonomic performance has deteriorated with a growing amount of self-management requests for static autonomic machines [32, 33]. The dynamic autonomic machines using the shared CPU from the server machine provided a better response. The request for self-service better referred to the increased number of self-management requests.

7 Conclusion and Future Research Directions

The following concluding points on autonomic computing were discussed in this chapter:

1. Autonomous computation should change the way software systems are created. On the one hand, the transition must be such as to ensure that our growing dependency on increasingly complex computing systems remains safe and secure [34].
2. A lot of problems remain until the full autonomic dream of computation can be accomplished. Autonomic computation and its associated theoretical fields require further study.

3. Self-management raises significant questions about the confidence and trustworthiness that can be put on autonomous systems. Security and security are of the utmost importance here.
4. Existing management interfaces must be built in such a way as to allow new types of interfaces [24]. Human–machine interactions are brought on by the concept of autonomy. These factors are closely related to advances in sociology, psychology, and cognitive science.
5. The major challenges raised by autonomic computing promote interdisciplinary work to open up important opportunities for advancement in computer science. Inspiration from control theory and physiology took the feedback loop to the core of the autonomic system architecture. Multi-agent systems provided motivation for the convergence of multiple feedback loops into coherent systems [35].
6. In a distributed environment, the structure breakdown of the autonomic manager and knowledge base into subentities provided high cohesion, low coupling, and allowed framework implementation. This is useful in high-performance, thread-based processes situations.

References

1. Frey, S., Diaconescu, A., & Demeure, I. Architectural integration patterns for autonomic management systems. In 9th IEEE international conference and workshops on the engineering of autonomic and autonomous systems (EASe 2012), Novi Sad, Serbia, 11–13 April 2012.
2. Wang, B., Qi, Z., Ma, R., Guan, H., & Vasilakos, A. V. (2015). A survey on data center networking for cloud computing. *Computer Networks, 91*, 528–547.
3. Tian, W., & Zhao, Y. (2015). Optimized cloud resource management and scheduling theory and practice (pp. 51–77). In Waltham, M. K. ISBN: 978-0-12-801476-9.
4. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, G., Schopf, J., et al. (2009). Adaptive computing on the grid using AppLeS. *IEEE Transactions on Parallel and Distributed Systems, 14*(4), 369–382.
5. Kashefi, A. H., Mohammad-Khanli, L., & Soltankhah, N. (2017). RP2: A high-performance data center network architecture using projective planes. *Cluster Computing, 20*, 3499–3513.
6. Chen, T., Gao, X., & Chen, G. (2016). The features, hardware, and architectures of data center networks: A survey. *Journal of Parallel and Distributed Computing, 96*, 45–74.
7. https://docs.apprenda.com/8/platform-operations/manage-clouds.html
8. Coutinho, E. F., Gomes, D. G., & de Souza, J. N. (2015). An autonomic computing-based architecture for cloud computing elasticity. 2015 Latin American network operations and management symposium (LANOMS), Joao Pessoa.
9. Nahar, K., & Chakraborty, P. (2020). Improved approach of rail fence for enhancing security. *International Journal of Innovative Technology and Exploring Engineering, 9*, 583–585.
10. Maenhaut, P., Moens, H., Volckaert, B., Ongenae, V., & De Turck, F. (2017). Resource allocation in the cloud: From simulation to experimental validation. 2017 IEEE 10th international conference on Cloud computing (CLOUD), Honolulu, CA (pp. 701–704).
11. Bahrami, M. (2015). Cloud computing for emerging mobile cloud apps. 2015 3rd IEEE international conference on mobile cloud computing, services, and engineering, San Francisco, CA (pp. 4–5).
12. Sun, A., Gao, G., Ji, T., & Tu, X. (2018). One quantifiable security evaluation model for cloud computing platform. 2018 sixth international conference on advanced cloud and big data (CBD), Lanzhou (pp. 197–201).

13. Dewangan, B. K., Jain, A., & Choudhury, T. (2020). GAP: Hybrid task scheduling algorithm for cloud. *Revue d'Intelligence Artificielle, 34*(4), 479–485. https://doi.org/10.18280/ria.340413.
14. Geetha, P., & Robin, C. R. R. (2017). A comparative-study of load-cloud balancing algorithms in cloud environments. 2017 international conference on energy, communication, data analytics and soft computing (ICECDS), Chennai (pp. 806–810).
15. Dewangan, B. K., Agarwal, A., Choudhury, T., & Pasricha, A. (2020). Cloud resource optimization system based on time and cost. *International Journal of Mathematical, Engineering and Management Sciences, 5*(4). https://doi.org/10.33889/IJMEMS.2020.5.4.060.
16. Wadhwa, M., Goel, A., Choudhury, T., & Mishra, V. P. (2019). Green cloud computing-a greener approach to IT. 2019 international conference on computational intelligence and knowledge economy (ICCIKE) (pp. 760–764).
17. Kaur, A., Raj, G., Yadav, S., & Choudhury, T. (2018). Performance evaluation of AWS and IBM cloud platforms for security mechanism. 2018 international conference on computational techniques, electronics and mechanical systems (CTEMS) (pp. 516–520).
18. Choudhury, T., Gupta, A., Pradhan, S., Kumar, P., & Rathore, Y. S. (2018). Privacy and security of cloud-based internet of things (IoT). Proceedings – 2017 international conference on computational intelligence and networks, CINE 2017. https://doi.org/10.1109/CINE.2017.28.
19. Bansal, S., Gulati, K., Kumar, P., & Choudhury, T. (2018). An analytical review of PaaS-cloud layer for application design. Proceedings of the 2017 international conference on smart technology for smart nation, SmartTechCon 2017. https://doi.org/10.1109/SmartTechCon.2017.8358374.
20. Dua, K., Choudhury, T., Rajanikanth, U., & Choudhury, A. (2019). CGI based syslog management system for virtual machines. *Spatial Information Research*, 1–12.
21. Srivastava, R., Sabitha, S., Majumdar, R., Choudhury, T., & Dewangan, B. K. (2020). Combating disaster prone zone by prioritizing attributes with hybrid clustering and ANP approach. *Spatial Information Research*. https://doi.org/10.1007/s41324-020-00369-z.
22. Marbukh, V. (2016) Systemic risks in the cloud computing model: Complex systems perspective. 2016 IEEE 9th international conference on cloud computing (CLOUD), San Francisco, CA (pp. 863–866).
23. Mengistu, T., Alahmadi, A., Albuali, A., Alsenani, Y., & Che, D. (2017). A "No Data Center" solution to cloud computing. 2017 IEEE 10th international conference on Cloud computing (CLOUD), Honolulu, CA (pp. 714–717).
24. Bhardwaj, T., Upadhyay, H., & Sharma, S. C. (2019). Autonomic resource allocation mechanism for service-based cloud applications. 2019 international conference on computing, communication, and intelligent systems (ICCCIS), Greater Noida, India (pp. 183–187).
25. Neyens, G. (2017). Conflict handling for autonomic systems. 2017 IEEE 2nd international workshops on foundations and applications of self* systems (FAS*W), Tucson, AZ (pp. 369–370).
26. Bencomo, N. (2017). The role of models@run.time in autonomic systems: Keynote. 2017 IEEE international conference on autonomic computing (ICAC), Columbus, OH (pp. 293–294).
27. Sheshasaayee, A., & Megala, R. (2017) A study on resource provisioning approaches in autonomic cloud computing. 2017 international conference on I-SMAC (IoT in social, Mobile, analytics and cloud) (I-SMAC), Palladam.
28. Hadded, L., Ben Charrada, F., & Tata, S. (2018). Efficient resource allocation for autonomic service-based applications in the cloud. 2018 IEEE international conference on autonomic computing (ICAC), Trento (pp. 193–198).
29. Jaleel, A., Arshad, S., Shoaib, M., & Awais, M. (2019). Design quality metrics to determine the suitability and cost-effect of self-* capabilities for autonomic computing systems. *IEEE Access, 7*, 139759–139772.
30. Tadakamalla, U., & Menascé, D. A. (2019). Autonomic resource management using analytic models for fog/cloud computing. 2019 IEEE international conference on fog computing (ICFC), Prague, Czech Republic (pp. 69–79).

31. Viswanathan, H., Lee, E. K., Rodero, I., & Pompili, D. (2015). Uncertainty-aware autonomic resource provisioning for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*(8), 2363–2372.
32. Fargo, F., Tunc, C., Al-Nashif, Y., Akoglu, A., & Hariri, S. (2014). Autonomic workload and resources management of cloud computing services. 2014 international conference on cloud and autonomic computing, London (pp. 101–110).
33. Tahir, M., Mamoon Ashraf, Q., & Dabbagh, M. (2019). Towards enabling autonomic computing in IoT ecosystem. 2019 IEEE Intl Conf on dependable, autonomic and secure computing, Intl Conf on pervasive intelligence and computing, Intl Conf on cloud and big data computing, Intl Conf on cyber science and technology congress (DASC/PiCom/CBDCom/CyberSciTech), Fukuoka, Japan (pp. 646–651).
34. Li, H., Chen, T., Hassan, A. E., Nasser, M., & Flora, P. (2018). Adopting autonomic computing capabilities in existing large-scale systems. 2018 IEEE/ACM 40th international conference on software engineering: Software engineering in practice track (ICSE-SEIP), Gothenburg (pp. 1–10).
35. Liu, Y., Li, A., Liu, S., Zhang, A., & Ting, Y. (2019). Autonomic self-testing of regression and internationalization based on cloud computing. 2019 11th international conference on measuring technology and mechatronics automation (ICMTMA), Qiqihar, China (pp. 141–144).

Digital Dimensions of Industry 4.0: Opportunities for Autonomic Computing and Applications

Neha Sharma, Madhavi Shamkuwar, and Preeti Ramdasi

1 Introduction

Human being strives to make their life comfortable by implementing various technical aspects. Over the last few decades, the quest for transforming the life of mankind has been possible with several technical innovations that make life comfortable [1]. The initial revolutions were facing the challenge of working strength and slowly addressed it through inventions of equipment for supporting manual work. Later, it further advanced to Machine-centered automation during the third revolution, followed by emerging thoughts on Autonomic Computing and Human-centered automation [2]. The operational complexity was significantly minimal in the first revolution, scientists focused more on addressing repetitive operations, however, the complexity of operation kept on increasing with time. Human observations and experiences were being utilized for the standardization of information parameters, which led to the need for systematic capturing of desired parameters [2]. More matured and futuristic thoughts were given, thus high speed and high-volume data were being stored for reference and reuse. While a huge part of the generation has witnessed paper-based communication systems, wireless communication has aided the fastest communication method for quick and effortless data transfer. Wireless communication has facilitated "anywhere anytime connectivity" enabling quicker

N. Sharma
Analytics and Insights, Tata Consultancy Services, Pune, India

M. Shamkuwar
Zeal Institute of Business Administration, Computer Application and Research, Savitribai Phule Pune University, Pune, India

P. Ramdasi (✉)
TCS Data Office, Analytics and Insights group, Tata Consultancy Services, Pune, India
e-mail: preeti.ramdasi@tcs.com

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_19

business decisions [3]. Decision support systems like ERP (Enterprise Resource Planning) and MES (Manufacturing Execution Systems), which are computerized systems used in manufacturing, helps in tracking and documentation when raw materials are transformed into finished goods and have replaced traditional passive decision systems and human experience-based decisions. The decision systems have molted into active decisions with the emergence of artificial intelligence and data-driven approach [4].

Further sophistication to manufacturing industrial processes is forming a futuristic intelligent manufacturing industry. This is achievable due to the availability of high computational power, autonomous devices, simulation software, 3D environment required for data visualization, systems for real-time data capturing, huge data storage systems, processing tools to process data in real-time, and techniques for data analytics. All these smart components and smart devices work in coordination with each other and form an intelligent ecosystem called Cyber-Physical Systems (CPS) [5–9]. Hence, CPS is the key infrastructure encouraging the development of smart manufacturing, whereas the Internet of Things is considered as its backbone technology that embodies interrelated computing devices, mechanical and digital machines, the ability of data transfer over a network with no or minimal human intervention [8]. However, the real strength lies in the computational intelligence being embedded in each of the components in the manufacturing ecosystem, bringing applications from fiction to fact and enabling the fourth industrial revolution [8]. Thus, a feature-packed fourth industrial revolution is fundamentally based on the objective of autonomic computation, optimization, and customization of production; automation and adaptation; human–machine interaction (HMI); value-added services and businesses, and automatic data exchange and communication [3].

By the virtue of all the transformations, the world has come closer and is known as the "Connected world!" In fact, with the advent of "Autonomic Computing," it is precisely termed as a hyper-connected world [5]. These latest technologies can perform every possible task with human–machine interaction in a very small time and with a greater degree of precision [9]. In this chapter, an attempt has been made to carry out a state-of-the-art review of computational intelligence in manufacturing. The next section presents a systematic literature review from various perspectives and section 3 discusses an interesting journey from the first to the fourth industrial revolution. Section 4 presents the key technology drivers of Industry 4.0 under three categories, that is, infrastructure, software, and process, whereas section 5 attempts to re-envision the manufacturing industry. Section 6 presents design principles, followed by references mentioned in the last section.

2 Related Work

This section throws light on multiple available research articles and captures various dimensions of Industry 4.0 and present in a systematic manner. Leurent et al. defined the type of industry as "front runners", "followers", and "laggard" in McKinsey's

report (2019) based on the time of absorption and implementation of Industry 4.0 [10]. The foremost term is for those who have absorbed the fourth Industrial Revolution (4IR) in the first 5–7 years, the second term is for the companies who would be absorbing it by 2030, and last are those who would be unable to absorb till 2030. Also, it is expected that "front runners" would contribute to 122% cash flow in industry, whereas "followers" might contribute to 10% cash flow; and "laggard" will thus be considered as no more profit-making industries. The report also specifies lighthouses as in factories that have adopted 4IR from the pilot project toward integrating all factory processes [10]. To know the industry perception regarding 4IR, several research projects are carried out and surveys are being conducted. These projects give various models and assessments to know more about the structure of the adopted model of 4IR. One such study conducted by Lichtblau et al. (2015) is termed "IMPULS—Industrie 4.0 Readiness" to quantify industries readiness for absorbing Industry 4.0 [11]. The readiness levels are measured over a scale from 0 to 5, where 0 indicates "Outsider" and 5 is termed as "Top performer." The study also involved various dimensions such as "Strategy and organization," "Smart factory," "Smart operations," "Smart products," "Data-driven services," "Employees"; and the points scored for each of the dimensions were noted. The study concluded by identifying various obstacles in 4IR implantation to achieve "Level 5-Top performer" [11].

PwC published an article "Industry 4.0: Building the digital enterprise" to give insights to companies in form of a maturity model to assess their capabilities to assimilate Industry 4.0 [12]. The four stages are Digital novice, Vertical integrator, Horizontal collaborator, and Digital champion. The dimensions are "Digital business models and customer access", "Digitization of product and service offerings", "Digitization and integration of vertical and horizontal value chains", "Data and Analytics as core capability", "Agile IT architecture", "Compliance, security, legal and tax", and "Organization, employees and digital culture". This maturity model has four stages and seven dimensions, which gives a comprehensive perspective. The model was represented using 33 questions, which consist of details of the use of 4IR, industry type, region, and annual avenue [12]. The other model has nine dimensions of strategy, like Leadership, Customers, Products, Operations, Culture, People, Governance, and Technology with 62 maturity items examined under five different levels [13]. Level 1 indicates that the attributes required for Industry 4.0 are missing in some companies while at level 5 the companies have all attributes for 4IR absorption. Many research areas cover the lean manufacturing aspects with Industry 4.0 to analyze the impact of using 4IR on lean manufacturing. The study made by Sanders et al. (2016) evaluated the linkages among the aforesaid concepts to know whether 4IR can leverage lean manufacturing [14]. The review made by Doh et al. (2016) on literature attempts to identify the automation need required in production systems to develop a framework integrating various information technologies with supply chain attributes [15].

To find the research trend regarding various dimensions of Industry 4.0, the four pillars of the 4IR, that is, "intelligent manufacturing," "smart manufacturing," "cloud manufacturing," and "IoT manufacturing" were studied in EBSCO research

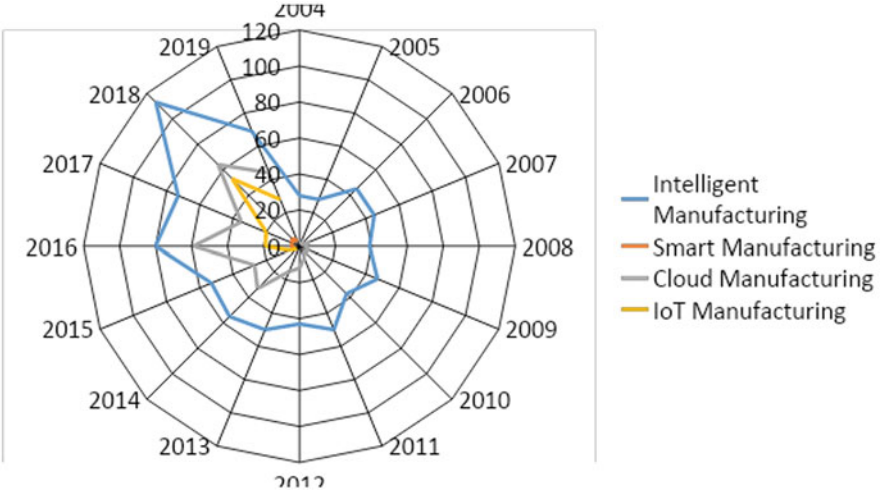

Fig. 1 Research trend of "intelligent manufacturing," "smart manufacturing," "cloud manufacturing," and "IoT manufacturing" in EBSCO research database from the year 2004 to 2019

database [16]. The research trend is well portrayed with help of a radar graph studied for the last 15 years as presented in Fig. 1. It is found that the term "intelligent manufacturing" is more popular among its fellow 4IR pillar. The popularity metrics of "cloud manufacturing," "IoT manufacturing" seems to be half of "intelligent manufacturing." Also, there is much scope for research on "Smart manufacturing" as the research seems to be in the nascent stage.

Furthermore, the EBSCO research database was studied to understand the journals-wise, subject-wise, and country-wise research on "intelligent manufacturing," "smart manufacturing," "cloud manufacturing," and "IoT manufacturing" as depicted in Figs. 2, 3, and 4 respectively.

Figure 2 clearly shows that the two journals, that is, "International Journal of Advanced Manufacturing Technology" and "IEEE Access" have covered the articles on all the four pillars of the 4IR, that is, "intelligent manufacturing," "smart manufacturing," "cloud manufacturing," and "IoT manufacturing." However, "Intelligent Manufacturing" has been very popular with a maximum number of publications. Figure 3 depicts that the "Business Management" subject has covered all the four pillars in great detail, followed by subjects like Information Science and Systems, Industrial Engineering, Economics, Computer-Aided Design and Production Control, Electronics, and Production Technology. It suggests that business decisions and economic growth are very much dependent on these four pillars. Figure 4 presents the countries that are leaders in the fourth industrial revolution and research related to it. The aim is to find business prospective, job opportunities, and sector analysis. Figure 4 shows that the USA, UK, Germany, India, and Netherlands are leading countries in Industry 4.0 research publications.

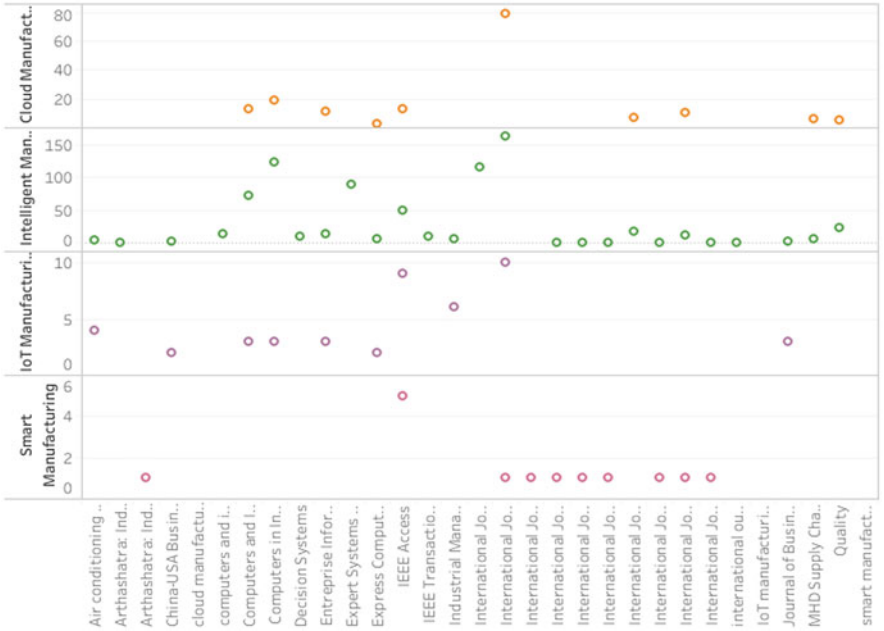

Fig. 2 Top 10 journals to publish the research on Industry 4.0 in EBSCO database

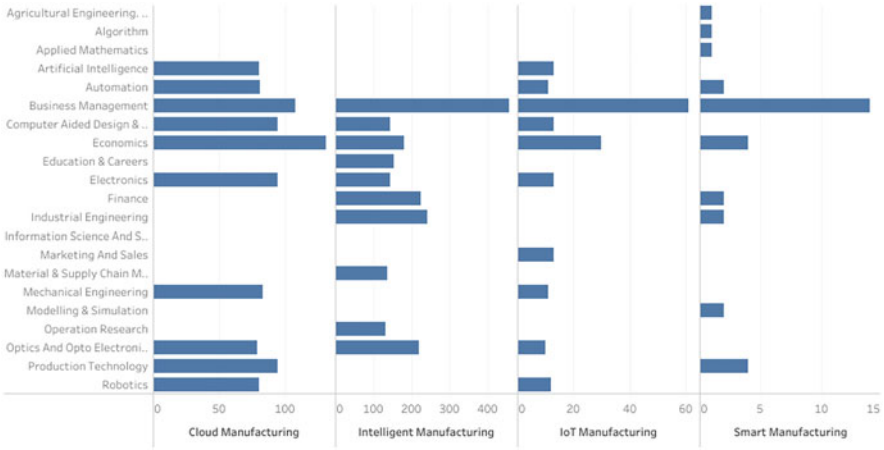

Fig. 3 Subject-wise research trend of industry 4.0 as per EBSCO research database

The above visualizations conclude that the fourth industrial revolution is the new heartthrob in the industrial zone and will continue to remain so for the next few decades. The four pillars are discussed whenever there is a symbiotic relationship between "intelligent manufacturing," "smart manufacturing," "cloud

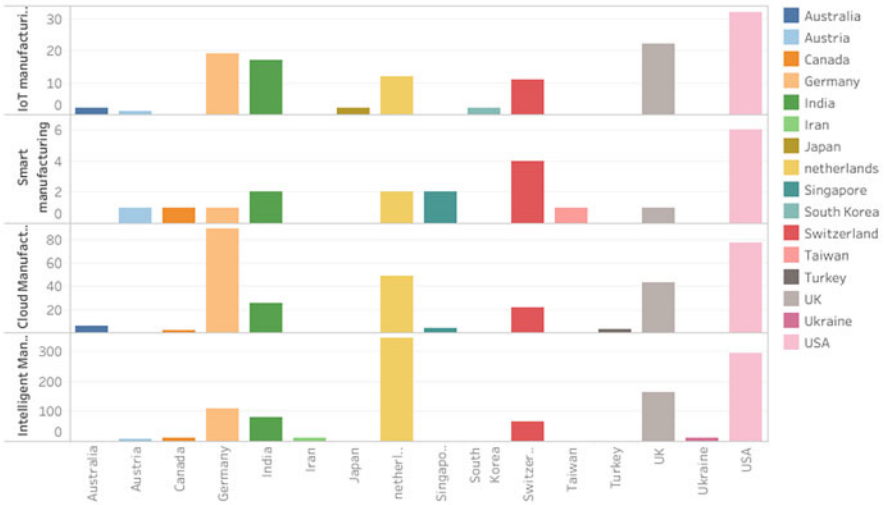

Fig. 4 Country-wise research trend of industry 4.0 as per EBSCO research database

manufacturing," and "IoT manufacturing" and the fourth industrial revolution. Although smart manufacturing needs to be more nurtured with the crux of 4IR, the rest three remains its soul.

3 First Industrial Revolution to Fourth Industrial Revolution: An Evolution/Synopsis

The present state of technological advancement is the result of four Industrial Revolutions over two and a half centuries [17, 18]. Industrial revolutions were slow at the beginning, but gradually with time and experience, they got the impetus and an ultimate great impact on carving our lifestyle, our living, our professional life, manufacture things, consume, communicate, and travel [19–22]. The first and second Industrial Revolution lasted for almost one century each, whereas the third and fourth revolutions attained maturity very fast [23]. Each Industrial Revolution influenced almost every aspect of human life in a big way and gave a boost to their income rate and exhibited extraordinary growth that was sustainable [24–27]. The three dynamics on which the equilibrium of Industrial Revolutions resides are a novel source of energy, a communication system, and a new financial system [28].

4 First Industrial Revolution

Industrial revolutions begin slowly, but over time they gather momentum and ultimately have a profound impact on shaping the way, and where, we produce things, and the way we live, consume, communicate and move.

Toward the end of the eighteenth century, the first industrial revolution started and lasted till the starting years of the nineteenth century. The paradigm shift in the manufacturing industry started in Great Britain and was mainly concentrated there for the entire duration and in the early nineteenth century spread to Europe and the United States [29–35]. During this century, inventors transformed the manufacturing processes by creating machines and devices that used two powers, namely, steam and water to mechanize the work which needed human labor and hence significantly increased production using mechanical systems [18]. The three key drivers for the first Industrial Revolution are:

New source of energy—The source of energy changed from wood power to a much denser source of energy, that is, coal power, which was in turn used to generate steam power that was used to power mechanical engines and devices [36].

New system for communication—The cylinder of the printing press was mechanized through steam power that could print newspapers and magazines in large number delivered speedily via steam-powered train/locomotive and ships to far and wide, which enabled the era of mass education and fast mass travel opportunities for the first time [37, 38].

New financial system—Energy source and communication systems were supported by the banks and industrial financers. London stock market established in the 1770s and the New York stock market established in the 1790s, also generated the finance to support the initiatives [20, 21].

The synergy among all three drivers was perfect but was not predicted before. The major focus was on textile, coal, iron, and railroads.

5 Second Industrial Revolution

A time period from the late nineteenth to the early twentieth century is termed as the second industrial revolution, which is popularly remembered as the technological revolution that led to unprecedented urbanization and globalization, as well as witnessed many important inventions [39, 40]. It mainly started in Britain, continental Europe, North America, Italy, and Japan which was a great leap forward to set a strong foundation for present technology and society as well as presented a rough sketch of today's world [41]. It brought a radical increase in production using electrical energy [8, 9, 18]. The inventors, researchers, industrialists, and government tried to improve the manufacturing and production methods of the first Industrial Revolution to be better, faster, and cheaper. Many path-breaking

inventions and improvements happened during this time like the iron was replaced by steel as it is stronger and cheaper [42]; the invention of the incandescent light bulb and phonograph in the late 1800s by Thomas Edison gave the gift of light and sound to the world [43]; laying down of underwater telegraph cable across the Atlantic Ocean in 1866, exactly 10 years after the telephone was invented by Alexander Graham Bell, improved the communication. Likewise, internal combustion engine invented by German scientist Gottlieb Daimler in 1886, which used gasoline was used to create the first automobile and the introduction of the assembly line by Henry Ford in 1914 made mass production possible [44, 45]. The focus areas during this era were on steel production, the automobile, and progresses in electricity [46]. The three factors that mark the technological advancement of the second revolution are:

New source of energy—One fossil fuel replaced the other, that is, oil or petroleum replaced coal as a new source of energy which was a much denser power source. People were aware of the presence of oil under the earth for thousands of years but were not sure about ways to use it. However, in the 1850s, scientists invented a technique to transform oil into a fuel called kerosene. Kerosene was then used for daily routines like cooking, heating, and lighting. Besides, researchers in basic physics from the UK universities got breakthroughs that resulted in the invention of electricity, which became a critical source of light and power and led to many interesting applications like the electric light bulb.

New system for communication—Another interesting application was new communication systems—from telegraph to telephone. Another long-lasting invention was that of internal combustion engine by Daimler and Benz in Germany, which used oil and electricity together.

New financial system—Second Industrial Revolution saw many entrepreneurs who established new businesses as corporations, which involved many investors and stockholders (can be individuals or institutions such as insurance companies). This new financial model allowed most of the peoples to engage in entrepreneurial activities due to lesser risks involved in startup ventures.

6 Third Industrial Revolution

With the advancements in microelectronics and other automation technologies, there was a need to revolutionize industrial working, which gave birth to the third industrial revolution from mid twentieth century. In this digital revolution era, magical semiconductors were introduced, mainframe computers and personal computers were disruptive and a game-changer "internet." Various researchers from IEEE (Institute of Electrical and Electronics Engineers), IFIP (International Federation for Information Processing), IFAC (International Federation of Accountants), and other relevant institutions came together to draw a roadmap for improving the manufacturing process [47–50]. New opportunities in the field of industrial informatics were discovered to automate the manufacturing process, like Flexible Manufacturing

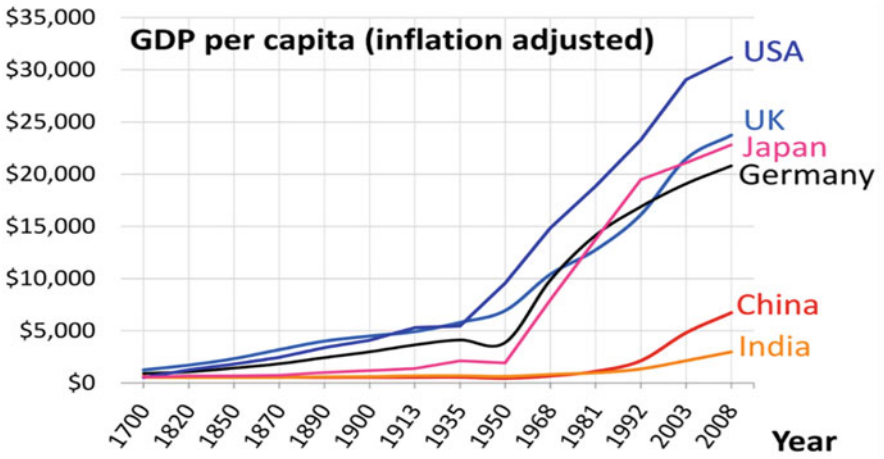

Fig. 5 Per capita GDP growth from 1700 AD to 2008 AD [52]

Systems was possible by the extensive use of computer numerical control (CNC) and industrial robots; similarly, Computer Integrated Manufacturing (CIM) was possible due to widespread use of computer-aided design (CAD), computer-aided manufacturing (CAM), and computer-aided processing planning (CAPP) [18, 47, 48, 51]. The backbone for industrial integration is the implementation of ICT infrastructure instead of earlier industrial electronics made a benchmark of using industrial informatics [47, 48, 51]. Figure 5 shows the growth in per capita GDP (inflation adjusted to 1990 International Geary–Khamis dollars) from 1700 AD to 2008 AD (for approximately 308 years) in China, Germany, India, Japan, the United Kingdom, and the United States of America [52]. It clearly shows the jump in the economy in the third industrial revolution, and the three factors responsible for digital advancement during the third revolution are:

New source of energy—The new energy source catering needs of the third industrial revolution is the use of solar power.

New system for communication—The new communication system was developed when personal computers were linked with Tim Berners-Lee's World Wide Web and Marc Andreesen's web browser Mosaic/Netscape system. However, the internet remained ahead of the other two factors and popular as a new communication system.

New financial system—The new financial system is mainly internet-driven innovations and people-driven initiatives such as crowdfunding and peer-to-peer finance, which catalyze the financial process and indicates its democratization.

7 Fourth Industrial Revolution

The fourth Industrial Revolution (also known as Industry 4.0) is the most recent one in the twenty-first century, started just a few years back, with an idea to integrate and extend the manufacturing process within and outside the organization and help them become a leading digital organization. Basically, "Industrie 4.0" an initiative by the German Government to be the strategic forerunner in revolutionizing and innovating the manufacturing sector and have clearly specified in its "High-Tech Strategy 2020 Action Plan" [53]. Gradually, other industrial countries also proposed similar approaches like "Industrial Internet" [54] by the United States and "Internet +" [55] by China. The focus in Third Industrial Revolution was on the automation of processes and machines, whereas Fourth Industrial Revolution is an attempt to optimize the digitization done in the previous revolution with a focus on end-to-end digitization and accumulation of industrial ecosystems by seeking complete digital integrated solutions [56]. Before the fourth industrial revolution, the production processes were not sustainable, it led to global warming, polluting the environment in every possible form, extensive consumption of nonrenewable energy resources [57]. A revolutionary change was thus required, which not only gives automated production but also is environment friendly. An aging population is another factor that demands automated systems and business transformation [58].

The foundation of Industry 4.0 is laid primarily by technologies like Cyber-Physical Systems (CPS), Cloud computing, IoT, and big data analytics, however, technologies like wireless sensor networks, embedded system, mobile Internet, adaptive robots, simulation, horizontal and vertical integration, Industrial Internet, dispersed device networks, additive manufacturing, and virtual reality also plays an important role [59–66]. Industry 4.0 is an attempt toward smart factories with smart machines, which is getting smarter with the availability of more data, and hence factories improve the efficiency, increases productivity, reduces waste, and make decisions and human involvement. Ultimately, it's the network of the digitally connected machines that creates and shares information, gathering and analyzing the data, and the supremacy of the fourth Industrial Revolution lies in the pace in doing so. As Industry 4.0 unfolds, its attempt to transform from mass manufacturing toward personalized production to produce things in smaller numbers in a more flexible and customized manner with minimum labor and yet being economical. The credit for this futuristic transformation goes to new materials, completely new processes like 3D printing, and automated robots integrated manufacturing services available online. And this approach will create a new and sustainable economy altogether [67, 68].

Figure 6 presents the key elements of each industrial revolution and the era in which it sustained while the next revolution was taking its shape. Whereas, Table 1 compares the distinguishing characteristics of each industrial revolution to give a complete overview of the journey.

Industry 4.0 is popular in both manufacturing industries and service systems. The core of Industry 4.0 is to establish the global value-added networks using cyber-

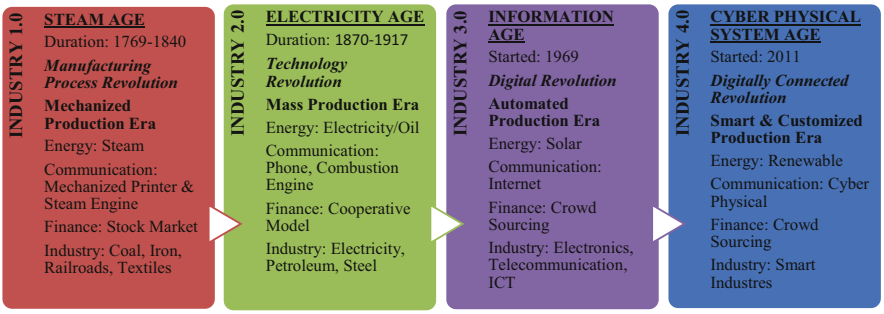

Fig. 6 Key highlights of the industrial revolutions

Table 1 Comparative summary of distinguishing characteristics of each industrial revolution

Distinguishing point	First industrial revolution	Second industrial revolution	Third industrial revolution	Fourth industrial revolution
Working strength	High strength manual work	Equipment supported manual work	Machine centered automation	Human-centered automation
Operation complexity	Simple and repetitive operation	Complicated and repetitive operation	Complicated and non-repetitive operation	Complicated and unpredictable operation
Information type	Human sensory information	Standard identification information	One dimension production information	Multi dimension production information
Communication technology	Human language communication	Paper-based communication	Fieldbus/TCP-IP/distributed control system	5F, Wi-Fi, wireless communication
Human–machine interaction	Mechanical switch and control over	Machine control panel	Digital dashboard	Mobile device (speech–gesture recognition)
Human decisions	Human experience decisions	Passive decision following the plan	Decision supported by ERP/MES	Active decision + AI

physical systems [69–71] that incorporate production facilities with warehousing systems and logistics [72]. Intelligent industries thus construct products and services with high quality and cost-effective. This computation intelligence embedded in software drives manufacturing processes of industry products is in more demand. The robotic and automatic technology provides a digital ecosystem that is operational productive, flexible, versatile, safer, and collaborative.

8 Fundamental Technology Drivers of Industry 4.0

Industry 4.0 prime aspects are to optimize and customize production; automate and adapt; human–machine interaction (HMI); value-added services and businesses, and automatic data exchange and communication [73, 74]. This section presents the study of key technologies that have triggered computational intelligence in the manufacturing sector, thereby driving the industry toward Industry 4.0. The technologies can be broadly categorized into infrastructure, software applications/tools, and processes. Infrastructure includes technology like the Internet of Things (IoT), Cyber-Physical System (CPS), and Cloud Computing. Software applications and tools that generally play an important role are Information and Communications Technology (ICT), Robotics Process Automation (RPA), Artificial Intelligence (AI), Graph Theory, and Digital Twin. Finally, big data analytics is used to process the structured and unstructured data generated and get some insight for the manufacturing industry to do things differently.

8.1 Infrastructure

8.1.1 Internet of Things (IoT)

For the future's complex industrial ecosystems, IoT is capable of providing transformational operation solutions in a digital enterprise and hence is one crucial infrastructure of Industry 4.0. According to GTAI (2014), IoT is a key enabler that has made its revolutionizing mark in creating "Smart factories" so that the production for existing manufacturing systems is carried to the next level in form of advanced manufacturing and virtual networks [75]. Generally, in the manufacturing industry, IoT can be used to achieve regular work of control and automation like heating, lighting, machining, remote monitoring, robotic vacuum, and so on, using automatic identification (auto-ID) technology in IoT.

IoT is a network of interconnected digital objects (devices) to collect and exchange data generated by them or the processes. These devices are sensors, actuators, and smart objects embedded in a network especially established for a designated purpose [76]. It is not a standalone technology but a blend of many technologies like universal wireless standards, analytics, artificial intelligence, machine learning, and so on [77]. IoT facilitates object-to-object communication with advanced connectivity among smart objects, smart systems, and smart services for data sharing. These objects have smart sensing abilities for inter-object coordination so that they interact through interconnectivity and handle the huge data which is generated by these objects during sensing movements. Table 2 shows the applicability of IoT in various manufacturing industries.

Nowadays due to the rapid use of the Internet and other technical advancements, data is universal and omnipresent give gigantic rise to big data [91, 92].

Table 2 Application of IoT in the manufacturing sector

Application	Company name	Purpose	IoT device used	References
Motorcycle assembly line	Loncin Motor Co. Ltd.	1. Increases manufacturing flexibility, work visibility, traceability 2. Raw materials data collection in real-time 3. For the production system, management of motorcycle assembly line	RFID	[78]
Automotive part manufacturing	SME Engine Valve Manufacturer – Huaiji Dengyun Auto-Parts (Holding) Co., Ltd.	1. Automotive part manufacturing process solutions 2. ERP and execution of manufacturing system	RFID	[79]
Shop floor material management	Guangdong Chigo Air Conditioning Co. Ltd	1. Real-time object visibility and traceability 2. Automatic and accurate object data identification	RFID	[80]
Smart infrastructure (e.g., parking area monitoring, traffic control, connected home)	Apple	1. A building is made up of integrated smart devices 2. Control over door locks from remote devices 3. Thermostat regulation 4. Refrigerator regulation to control food items temperature and so on 5. Provides flexibility, reliability, safety, and efficiency in infrastructural functions	Sensors	[81]

(continued)

Table 2 (continued)

Application	Company name	Purpose	IoT device used	References
Healthcare: Sensors, smart T-shirts		1. Patient monitoring system that connects with exchanges health information with doctors 2. T-shirt senses heartbeats, blood pressure, measures calories using a smart phone	Sensors	[82, 83]
Supply chains/logistics		1. Effective logistics and supply chain operations 2. Up to date and detailed information Product traceability	IoT	[84–86]
Security and privacy		1. Provides security and privacy 2. Prohibits from unauthorized access	IoT	[87]
Smart community, Canada and China		1. Community monitoring 2. Ubiquitous healthcare system	IoT	[88]
A cloud implementation using Aneka, Australia		1. Application developers can able to share data 2. Framework for IoT applications	IoT	[89]
IoT-enabled energy management, Italy and Spain		1. Energy management strategies 2. IoT implementation 3. Providing a framework to support the integration of energy data	IoT	[90]

8.1.2 Cyber-Physical System (CPS)

Cyber-physical system (CPS) brings together intensively connected computational entities and the physical world along with its processes. The physical processes are monitored and controlled by computers on the networks, and the feedback from the physical system is used to perform computations accurately and within the stipulated time. However, software and physical component operate at different temporal and spatial scales and interact in innumerable ways that change with context [93–95]. The most important part of these CPS systems are accessing and processing data available on the internet. CPS can be defined by the following characteristics:

- Computational capability in physical components.
- Automation of high degree.
- Interconnectivity at multiple scales.
- Integration at multiple temporal and spatial scales.
- Reconfiguring dynamics.

Industry 4.0, which talks of smart factories, has CPS at its core, to provide the competencies to be self-awareness, self-comparison, self-prediction, self-reconfiguration, and self-maintenance [96–99]. Information and communication technology (ICT) is also a significant part of the smart factory and is represented by CPS. As per the research related to CPS, machines in factories will be able to communicate and optimize production due to decentralized control systems. These transformative technologies exchange data, result in appropriate actions, and controls each other independently using smart machines and storage systems, autonomously. Key issues at smart factories like meeting the requirements of individual customers, handling business dynamics, optimizing decision-making, improving resource and energy efficiency, improved work–life balance, and so on, can be solved by CPS that makes use of big data and high-class interconnectivity to achieve the expectations and goals of intelligent and self-adaptable machines [100, 101].

8.1.3 Cloud Computing

Cloud-based is another technology booming the Industrial 4.0 revolution for significantly recognizing [102] the potential with its feature of networked system integration. Cloud manufacturing leads to the creation of new service platform provided by Cloud computing, which is Manufacturing-as-a-Service (MaaS) giving high performance and low cost [77]. It gives benefits like large data management, resource sharing, services sharing, dynamic allocation, flexible extension, fact computation, quick production, modularization and service-orientation [102–104], and many other benefits to facilitate the manufacturing and production process. The term Cloud computing in Industrial manufacturing or Cloud manufacturing (CM) is used apart from Industrial Cloud computing regarding Industry 4.0; any of them

implies a coordinated production with "available-on-demand" feature for linked production. This production strategy minimizes the time required for the product life cycle, facilitates optimum utilization of resources, and constantly varying customer's demand, and coping with the same using focused works [105–107].

Cloud computing working is highly dependent on crowdsourcing platforms and social networking to develop product models. This technology further helps in decision making, automated production, demand- prediction, works in line with the human workforce, and thus plays a significant role in industrial development. The best example of Cloud manufacturing is in the automotive sector—Google cars, which test the vehicles both on road and in highly simulative environment. The data required and generated while testing and driving is being stored in Graphical processing units and Cloud TPUs (Tensor Processing Units) [108]. Not only the data, the code required for performing the car processes is also stored in the Cloud. The Cloud stores all the information such as username–password, connectivity type, authentication information, prerecorded data of drive, in advance storing of data when the destination is not in the network range [109].

8.2 Software Applications/Tools

8.2.1 Information and Communications Technology (ICT)

ICT is an integration of communication technologies that can collect, store, transmit, and manipulate data [110]. ICT involves various signal-processing techniques, communication standards, wired/wireless systems, enterprise systems, middleware, and audiovisual systems. The use of ICT gives better autonomy and better-controlled systems, which provides controlled operations for plant managers, production managers, and workers to perform related operations in an organization [111]. As per the World Economic Forum 2016, it is predicted that by the year 2020 the job that seems to as of lesser importance will be the most demanding jobs with skills not predictable earlier. World Economic Forum (WEF 2016) predicted that future personnel will require a combination of KSA, knowledge–skills–abilities with 52% cognitive abilities, 42% systems skills, and 40% complex problem-solving skills. The workforce will inquire about ICT skills apart from aforesaid skills and abilities, who can code, analyze big data, developing applications, handle complex databases, and manage networks.

Whereas, OECD 2016 predicted that 65% of the job that does not exist till now will be developed in the coming decade and ICT skills will add innovation quotient for a business to flourish as well as provisions will be made for digital infrastructure on which the organization relies on. The use of ICT demands information resources and results in preparation for the dynamic market growth, customized products, virtual production, business competitiveness, agility, productivity, reduce costs, and customer–clients agreements in sectors such as education, automobile, healthcare, tourism, and so on. [112, 113].

8.2.2 Robotics Process Automation (RPA)

Robots were initially developed to reduce human errors, avoid risks, work efficiently, and with accuracy. Industry 4.0 uses the latest technologies to cope with the ever-changing industrial requirements for smart manufacturing. Industrial robots are smart machines tailored, especially, to cater to the requirements of the manufacturing process of an industry. They are highly automated in nature, and capable of making smart decisions to enhance industrial processes. Industrial robotics or Robot manufacturing as said earlier avoids injuries or casualties while production and further ensures quality standards for gaining fiscal achievements.

They are designed to collaborate with human beings and with other peer robots connected over a network. Their configuration makes them adaptable to the new products and manufacturing processes [114]. Robotics process automation further collaborates with other recent technologies such as the Internet of Things (IoT) to control and remotely monitor other industrial robots, cloud computing that is responsible to process big data, and advanced information analytics to provide smart factories with many such robots [115]. These robots further autonomously detect product's performance, finds and applies optimized solution for it. To create a smart factory that not only automates the manufacturing process but also gives transparent and enhanced quality work a strong framework integrating all the required technologies needs to be designed [116]. Novel robots with well-structured process models provide smart solutions to the industries for real-time projects and assure their guidance and support [117].

8.2.3 Artificial Intelligence (AI)

AI is not a single technology but is a superset of many other technologies like image processing, natural language processing, robotics, machine learning, and so on. Industrial AI develops, validates, and deploys machine learning algorithms created for industrial applications. AI gives a boost to performance and acts as a problem solver of many industrial applications. It applies a systematic approach to connect the academic outcomes and industry expectations. The production is raised with the automated industrial AI and leading to an increase in demand and competition [118]. The technologies complimentary are the Industrial Internet of Things (IIoT), big data analytics, cloud computing, and cyber-physical systems [119–129]. Industrial AI has five key elements as shown in Fig. 7.

Analytics brings a value worth to the enormous data collected and predicts the future of the organizations. Big data technology and Cloud or Cyber technology are the sources of information for AI. Domain knowledge involves solving the problem and providing AI-based solutions, techniques to gather the right data from the right place, understanding metadata and their association among them, machine-dependent metadata. When the pattern is identified from the dataset "Evidence" will improve the AI model.

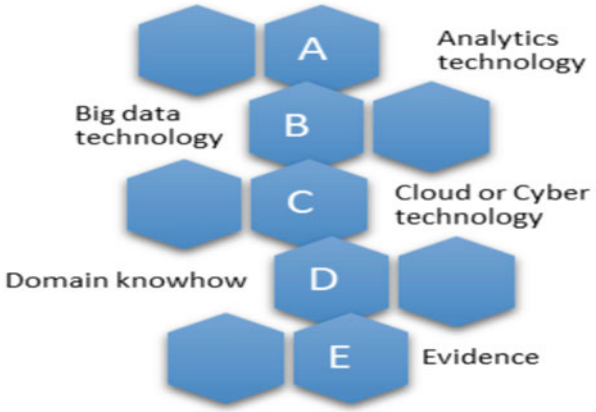

Fig. 7 Five key elements of industrial artificial intelligence

8.2.4 Graph Theory

While smart factory is wildly spreading, researchers and manufacturers are involved into newer and better ways of dealing with big data that is getting accumulated from a variety of sources in a structured and unstructured format. At the same time, data scientists and data engineers are behind an important task to generate meaningful values through data classification, clustering, and identifying significant relationships of data objects to each other. Latest technological solutions are graph databases and graph data modeling—a form of NoSQL software, which maintains data quite differently against a traditional relational database. Graph systems represent data as nodes linked to one another by edges. The edges are assigned a set of properties that mentions the relationship between nodes. The latest Graph database tools to name a few are, OrientDB, ArangoDB, Neo4j, IBM Graph, Apache Giraph, Amazon Neptune, Cassandra, Azure Cosmos DB, and DataStax.

The solution best suits the complex connected data environment where real-time data processing is demanded and hence a high processing efficiency is required. Graph database is also a choice as it provides flexibility and scalability at no additional cost and complexity through a schema-less data structure.

8.2.5 Digital Twin

Digital twin concept was bought in front of the world by Grieves, where the virtual objects are created using their physical objects to instigate the behavior of physical objects in real-world environment [130, 131]. Thus, the physical entities of the physical world gets mapped to the models of the digital world using the connection between them; the physical entities, models, and the connection forms three different components of the digital twin. This digital twin is bidirectional and

is used in Industry 4.0 for the manufacturing process. The concept aims to produce a copy of a part or a product and then using them for the reasoning of other instances of the other part or products [132]. Digital twin concept proves to be efficient in design, manufacturing, production, and servicing of products. A lot of business use cases simulated for different situations to predict their success or failure rate by taking into consideration the input taken from physical twin. The digital twin collects, views, analyses, and controls the industry process or equipment work.

8.3 Process

8.3.1 Big Data Analytics

Big data analytics produces an environment that focuses on the "predictions" rather than "history" of events and thus concentrates on building the models, which contribute significantly for forecasting the possibilities, which may occur in the future. The paradigm change in big data analytics is shifted from data collection to an outcome-based analytics apart from traditional "analyses." The concept is anticipated in Industry 4.0 era as it gives smarter business moves and outcomes [107, 133]. This big data environment includes logs, transactions, social media responses, web retorts, and data populated using sensors from a variety of data channels [134].

The responsibility of a big data environment is providing the right information at the right place and at the right time, which is achieved through proper data processing [135]. This data processing becomes crucial as the dataset is large and complex and may not be handled by traditional data analytics software [136]. A significant return on investment of 15–20% is achieved through the use of big data technologies by retailers, which lays a new research–academia–industrial wave [137]. For example, Customer Relationship Management (CRM) when integrated with BDA technologies leads to enhanced customer's satisfaction by customer engagement [138]. Table 3 below presents the top 10 companies using big data analytics with their transactions.

9 Reenvisioning Manufacturing Industry

Globally, any economic development strategy aims at newer investments and sustainable growth. A manufacturing industry is always been considered a major contributor toward economic development. For business growth and increased profitability, it is very obvious that the manufacturing industry strategically thinks of balancing its investments in areas like the marketplace, the workforce, the shop floor, and the community. Therefore, innovation has gained the utmost importance in the ecosystem of the manufacturing industry [139]. Manufacturing industry is made computational intelligent by three technologies given below: smart manufacturing, IoT-enabled manufacturing, and cloud manufacturing.

Table 3 Use of big data analytics by industry leaders

Company name	Company type	Role played by BDA
Amazon	Online retailers	Customers purchase history and behavior
American Express	Financial services corporation	Analyze and predict consumer behavior.
BDO	National accounting and audit firm	Identifying risk and fraud during audits.
Capital One	Financial services corporation	Increasing the conversion rates
General Electric (GE)	Power generation and water technologies	Boost productivity and raise national income
Miniclip	Digital games	Improve use experiences
Netflix	Entertainment streaming service	Provides insights into online viewing habits
Next Big Sound	Online music	Predict music trends
Starbucks	Coffee house chain	Potential success of new franchisee
T-Mobile	Mobile communications	Predict customer fluctuations.

9.1 A Journey Toward Smart Manufacturing

Innovation in business means introducing new or improved products, services, or processes, newer approaches toward addressing ever growing competition, improving branding strategy, modifying the cost–profit financial model, and ultimately, meeting the end customer's satisfaction! A primary key to move any of the businesses forward is to "analytically" understand how innovation can add value to the customers. This study and observations lead either to an incremental innovation or disruptive innovation. The ladder to innovation is built through introducing new technology, techniques, and working practices. On a parallel track, innovation is matured through an important contribution by mathematicians and statisticians as well. Humongous data flowing in and out of the manufacturing processes and systems has become the playground for statisticians to try developing newer data models. Soon the keywords like Co-innovation, Collaboration, and Coexistence have gained unparalleled importance! This has developed an alliance of manufacturers and statisticians while they have started working toward common business goals. Thus, laying down the platform for a smart, intelligent, and more efficient manufacturing industry! The rise of smart factories is nurtured with technological enhancements in the areas of big data, artificial intelligence, cloud technology, and the industrial internet of things (IIoT). At the core of the transformation of manufacturing industry into a computationally intelligent manufacturing industry is ongoing experiments integrating any of these technologies to collectively develop applied solutions with business outcomes! [140]. The key characteristics of any smart factory could be as following [141]:

Fig. 8 Data-driven transformation—benefits at manufacturing industry [142]

- Ecosystem connectivity.
- Transparent supply chain.
- Flexible, optimized processes.
- Feasibility of data analytics and capability of advanced data analytics.
- Early predictions of risks and opportunities.
- Applied solutions based on machine learning and artificial intelligence, blockchain technologies.

9.1.1 Data Democracy: Directing Data Economy at the Ecosystem of Manufacturing Industry

Smart manufacturing is driven by "Data." The data economy at factories includes generating data, capturing and storing data systematically, then analyze data aligned to the business unit's objectives. Data analysis helps in making the production process efficient, transparent, and flexible. Figure 8 explains four of the major areas of manufacturing that contribute to increasing the revenue and cost reduction while industries are experiencing digital transformation through the fourth industrial revolution.

Earlier the mindset was to keep data in a closed environment, say within the boundaries of a specific function or unit. For example, product-related data, design, and development-related data was neither shared with the operations unit nor thought of utilizing along with service lines data. No other function was ever thought of analytically looking at supply chain data than the supply chain function and its stakeholders itself! However, Industry 4.0 suggests making industrial data available and accessible for researchers to experiment with the transformation in cross-functional areas of the manufacturing ecosystem through a digital thread to reap disruptive outcomes.

9.1.2 Data Points for Analytics: Data Generating Functions and Units

This section explains different functions in the ecosystem of an industry that holds the potential to generate significant data for performing analytics. One can understand how smart manufacturing makes use of raw data for optimizing and improving every phase toward finished goods production! [143]. These days, this concept of Industry 4.0 is tailored and applied to IT industries as well.

While planning for a new product or service, it is crucial to study the entire product life cycle management right from design, through supply chain to consumer feedback. The objectives in this study could be, minimum compromise on manufacturing production processes, improved quality of the product, shorter time to market, flexible manufacturing processes, and minimum logistics, quick and efficient way of market feedback, and remote monitoring of product and optimization services. Furthermore, manufacturers are also thinking about the concept of product as a service.

Analytics based on feedback captured at various stages of the product life cycle helps improve the design process, design approach, and improve product quality. Analytics around the supply chain is based on data captured on performance indicators like Inventory Turnover, Delivery/Shipment Time, Cash-to-Cash Cycle Time, and Warranty Costs. Manufacturing Execution System (MES) is an important platform that generates transactional process data related to manufacturing like planning, scheduling, maintenance, and material movement. Integration of MES data and Analytics plays a key role here in data exchange and optimizing workflows thus shortening time to market. Production data from machines and plants are used for creating reports, alarms, alerts, and further analysis of the production process. This helps in predicting the performance and early identification of the risk of failure. Correlation analysis among the various parameters is an important analytical tool to make decisions on process improvement and leaning the waste.

The flexible manufacturing system (FMS) is core to smart manufacturing. This aims at responding quickly and easily to predicted or unpredicted changes. To deal with these changes, new manufacturing software methodologies and processes are being adopted. Data integration and analytics greatly contribute to identifying alternative approaches. With smart industrial assets, the data is being made available for analytical use, thus making industries computationally intelligent. IIoT—Industrial Internet of Things has proved to be a backbone of smart manufacturing. IIoT allows the integration of physical production and operations with smart digital technology, machine learning, and big data to create a more holistic and better-connected ecosystem. Integrated systems that are integrated again as system of systems are based on Industrial robotics automation, the Internet of things, Cloud computing, and big data and analytics. Figure 9 illustrates key technology ingredients alongside product and process life cycles, industrial systems and subsystems, consumers, and industrial applications.

This forms a platform leveraged by manufacturing industries for revolutionary changes and growth [144]. Each of these components individually or in suitable combination with the other refers to a digital platform for data analysis. The

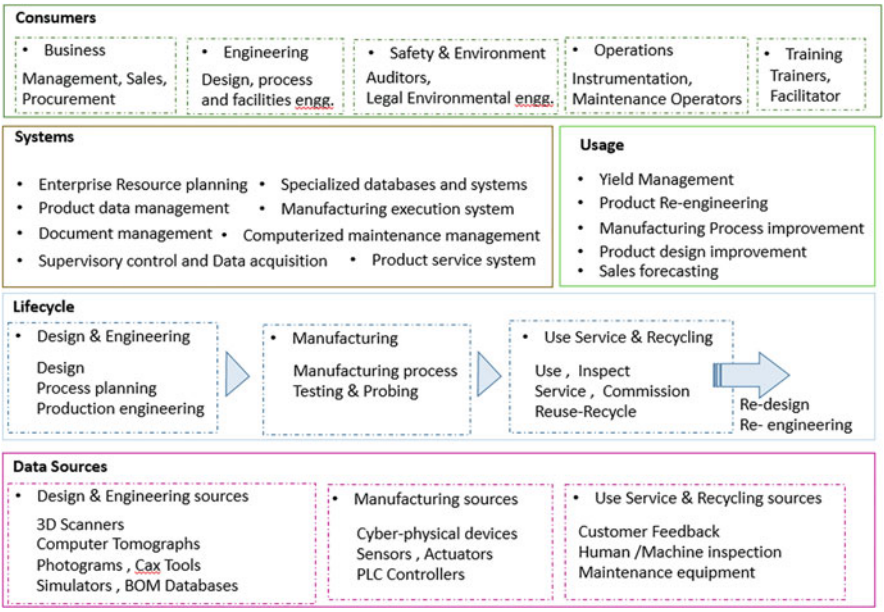

Fig. 9 Industrial ecosystem and technology platform—key ingredients

following sections briefly covers important aspects of the Industrial Internet of Things, Cloud computing, and Robotic automation in industries.

9.2 IoT-Enabled Manufacturing

The internet of things, or IoT, is a system of interrelated heterogeneous components on the network like many computing devices, industrial software systems, PLC-programmable logic controllers, actuators, a variety of sensors, and so on. Every component is uniquely identifiable through its ID and holds the capability of data capture and transfer. Such special-purpose industry-specific networks of IoT is termed as IIoT. IIoT helps to eliminate human-to-human or human-to-computer interaction and drastically enhances the speed of data transfer, maintains the first version of data, with high computational power supports intelligent systems. IIoT contributes to factory digitalization, thus enabling end-to-end smart manufacturing.

A digital twin is the latest buzzword that can be seen as an extreme example of IIoT. As the name suggests, the digital twin is an actual digital version of every machine, equipment, and industrial site. Thus, the digital twin act as a live simulation model of a part of the industry or the complete industry.

9.2.1 Implementation of IIoT

With innovation in internet technologies, manufacturing companies now rely on the power of the internet for their high-performing, uninterrupted services. Industrial Internet of Things (IIoT) suggests interconnectedness through data communication systems. Sensors are installed at each unit. These units generate data. It is packaged and transferred to cloud storage systems. Prior to storing entire data into the cloud for further processing, sometimes part of data is processed or passed through edge computing devices for immediate/real-time processing and analysis. For optimization of the data transfer and making the units smarter, a local or distributed computing through fog and/or edge computing approach is adopted nowadays. Data is accessible through the cloud to generate analytical insights. This data is churned through AI/ML algorithms and tools. This data is transformed into actionable knowledge for consumption by the decision-makers. Figure 10 explains high-level architecture of IIoT implementation. [145].

An IIoT strategy is proved effective based on meaningful data capture and thoughtful data storage in the cloud system.

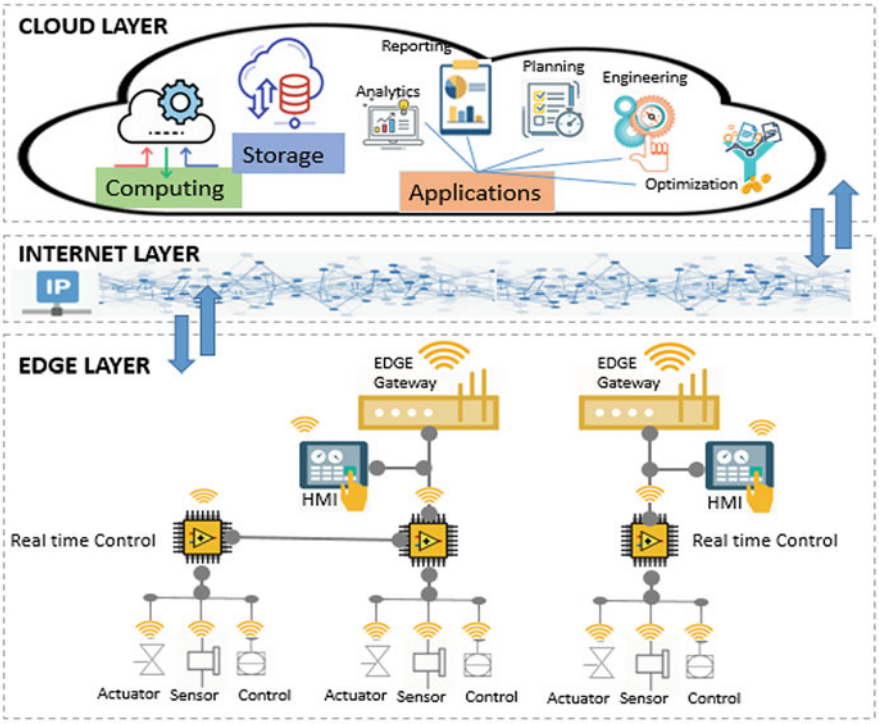

Fig. 10 IIoT architecture

9.3 Industrial Robotics and Automation

Yet another thing that contributes to a smart manufacturing facility is industrial robots-carried forward from the previous revolution of industry. Robots are well connected with the sensor network implemented within the manufacturing shop floor, and they get the data from sensors. They become more flexible in terms of work. This means, programming the robot has become less complex, but at the same time, feasibility of acting as per a modified program has increased due to hardware and software innovations.

Artificial intelligence and machine learning are being embedded in robotics systems. With this, automation systems have got a new capability of "self-decision and action." At times, it is termed as autonomous robot or adaptive self-learning robot system. Parallel systems are also getting evolved like Cobots—a collaborative robot. They are expected to interact with humans for their task in industry and assist manual activity [146]. They further have been classified based on the work robots are designed for. The IFR—International Federation of Robotics [147, 148] defines four types of collaborative manufacturing applications with the involvement of human beings and robots, which are mentioned subsequently:

- Co-existence: Both work with each other, but do not share a workspace.
- Sequential Collaboration: Both share all or some part of a workspace but do not work with each other on a part or machine at the same time.
- Cooperation: Both work on the same part or machine at the same time and are in motion.
- Responsive Collaboration: The robot is responsive to the worker's action in real-time.

9.4 Cloud Manufacturing

Cloud manufacturing is a paradigm shift to the traditional manufacturing processes, utilization of resources, and knowledge. This service-oriented platform has its strength in collation and integration of "traditional manufacturing" approach and modern technologies like the Internet of Things, high-performance computing, and cloud computing. Such a rich knowledge base is now made available for on-demand consumption to several industrial users. This innovation is very much appreciated with improved resource efficiency and higher productivity [150].

At the same time, this concept of virtual manufacturing environment has opened up vast opportunities of research in the following areas, although not limited. They are high-performing industrial network, cloud computing, industrial software as a service, micro-services, resource scheduler, or cloud cooperation management. A functional framework of a cloud manufacturing platform looks like the following Fig. 11.

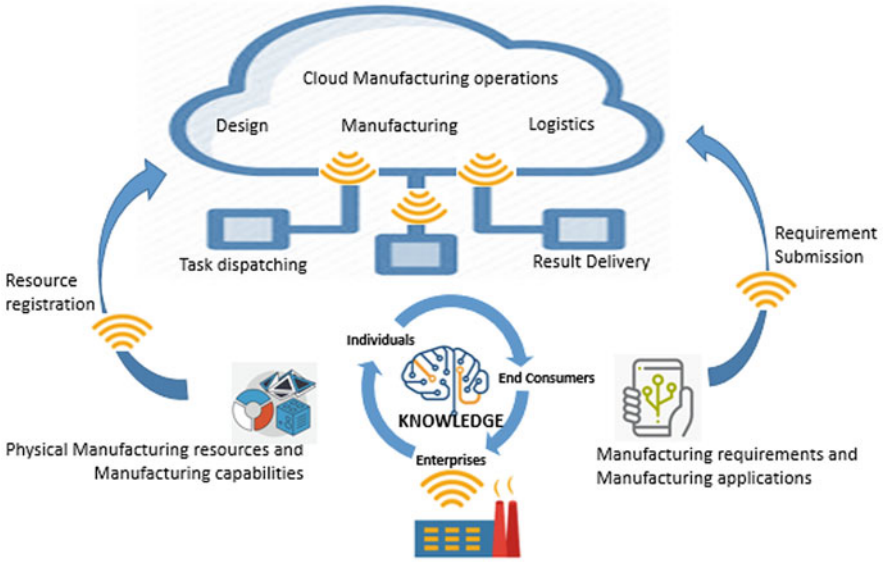

Fig. 11 A functional framework of cloud manufacturing platform [149]

10 Design Principles

The penetration of electronic devices and the internet is increasing at a rapid pace to alleviate the utilization of technology to impact lives for years to come. The number of electronic devices spans from personal computers, laptops, palmtops, tablets, mobiles, and many more, which are connected over the internet, so there is a need to consider certain design principles to cater to a variety of smart devices and their huge numbers. The "Next Digital Revolution" or "Next Generation of Internet" as both the terms coined involves adding smart technologies to TV, oven home, security appliances, refrigerator, and so on, thus automating every appliances and device around us for domestic or for industrial purpose. The design principles of Industry 4.0 should be taken into consideration before implementation as it provides a detailed description of the entire ecosystem, which facilitates coordination among various Industry 4.0 components. The seven major design principles of Industry 4.0 are presented in Fig. 12.

Academicians, researchers, experts, and industries can map various technology key drivers with the seven design principles and utilize them as per the need of its implementation in industries. The "green color" indicates the applicability of the design principle for the given technology, thus Table 4 gives a framework for implementation of 4IR with respect to seven design principles.

The most significant design principle that coordinates efficiently with the industry key technologies is agility and integrated business processes, they are common for communication and networking, sensors and actuators, RFID–RTLS

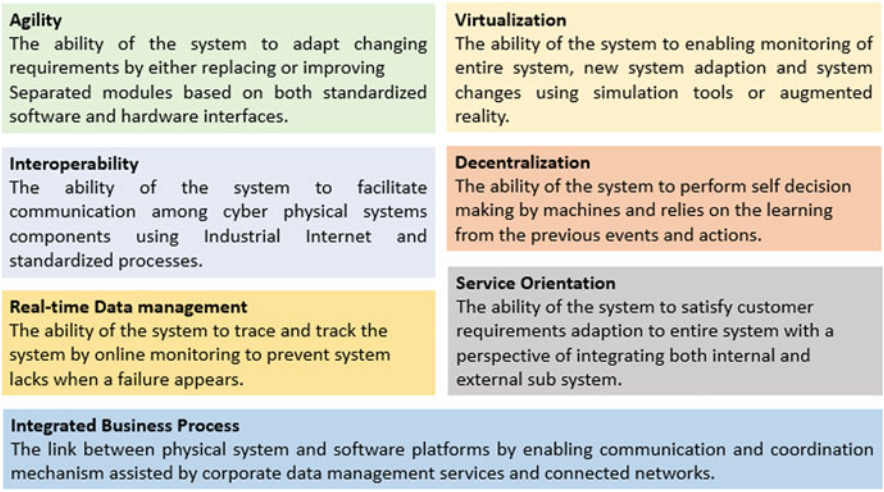

Fig. 12 Seven design principles of Industry 4.0

technologies, and cloud technologies. The relationship between cybersecurity and cloud systems is laid on the infrastructure of the industrial internet communication and networking infrastructure; these are covered under the umbrella of the integrated business process. When implementing 4IR key technologies these design principles are helped in the decision-making of choice of the technology. Figure 13 presents the design principles that play a vital role in the popularity of Industry 4.0 technologies.

The successful implementation of Industry 4.0 involves relooking into core functions such as product development, manufacturing, logistics, marketing, sales, and after-sale services performed by smart products and smart processes.

A smart product is composed of three key components: (1) physical/mechanical part(s), (2) a smart component that has sensors, microprocessors, embedded operating system, and user interface, and (3) connectivity part that has ports, antenna, and protocols.

The technological platforms perform connection, communication, and coordination from various products and services of external sources in the form of data exchange, data collection, data processing, and analytics. The products and services are monitored and changes are being recorded using big data analytics under various technical conditions. Besides, cloud computing and cloud computing technologies are used for distributed systems to ensure coordinated and linked production. To strengthen the interoperability with big data processing platforms networking essentials are mandated, which are agent-based services, real-time analytics, and business intelligence systems. To improve the product performance and utilization, the real-time data management is provided using big data technologies and cloud systems required for fast communications data processing, management of

Table 4 4IR key technology drivers and design principles

	Agility	Virtualization	Service orientation	Decentralization	Real-time data management	Integrated business processes	Interoperability
Additive manufacturing							
Adaptive robotics							
Adaptive manufacturing							
Mobile technologies							
Simulation							
Virtualization technologies							
Data analytics and artificial intelligence							
Communication and networking							
Sensors and actuators							
RFID–RTLS technologies							
Cybersecurity							
Cloud technologies							
Embedded systems							

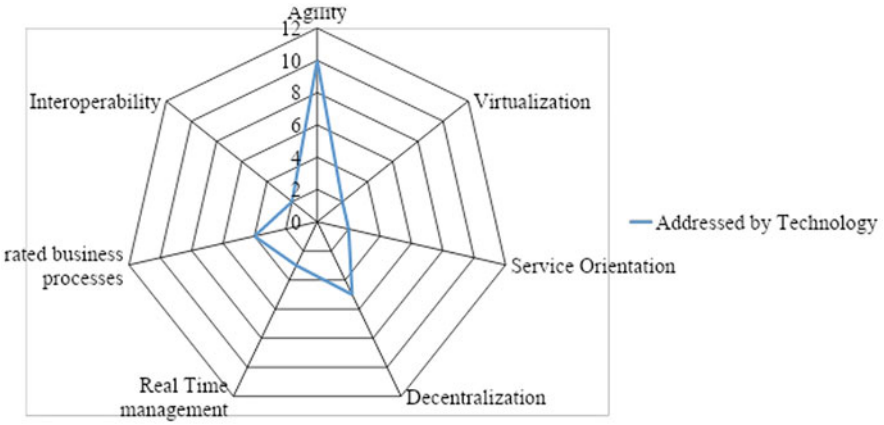

Fig. 13 Popularity of 4IR technologies with various design principles

data flow, and extracting know-how to improve entire product performance and utilization. Numerous algorithms and iterative processes need to be adapted for self-learning and self-assessment to make a balance between the current and expected conditions.

11 Research and Innovation Directions

(a) *Blockchain for IIoT*

Blockchain ensures data integrity in a distributed network. Recently, researchers have proposed the enhancement of the IIoT architecture by integrating blockchain technology. Also, there is a focus on developing a lightweight blockchain for IIoT.

(b) *5G Communications Technologies for IIoT Networks*

5G wireless converged automation protocols are expected to enable time synchronous operations. IIoT network is to be designed to be scalable and capable of delivering optimal performance for all industrial applications [150].

(c) *Network Security*

Cyberattacks and manual security testing are emerging threats considering the scale of big IIoT networks. Few related pointers would be auto identification of cyberattacks, automating the process of threat modeling, security analysis, and penetration testing.

(d) *Fog Computing*

Fog computing is an extension of the cloud computing concept to the network edge to make it ideal for the Internet of Things (IoT) that demand real-time interactions. Smart manufacturing is the largest area of applications of Fog computing because a variety of a large number of sensors employed

on the network generate an enormous amount of data. It's a great hope to manufacturing companies to reduce operating costs, as the architecture minimizes the need for a higher bandwidth requirement for sending data all the way to the cloud. According to Mung Chiang, who is one of the leading researchers in the field of edge and fog computing, "Fog offers the missing link for what data needs to be pushed to the cloud, and what can be analyzed locally, at the edge". Thus, this opened up many opportunities to develop innovative customized solutions for the industries and placing security features in a fog network.

(e) *DevOps in Manufacturing*

The Forrester research reveals that 50% of organizations are implementing DevOps in their rush toward a successful digital transformation. DevOps in the manufacturing industry is also becoming more and more necessary as Industry 4.0 and the Internet of Things find more applications in the space. Software components integrated with machines and processes are key to moving the industry forward. Following requirements would become pointers for further innovation. More agility-to speeding up processes become more agile in responding to customer/market, automatic code testing, continuous integration, and delivery.

12 Conclusion

The Industry 4.0 concept has been revolutionized with recent developments in information and communication technologies and has made its mark in a new business era. Industry 4.0 will change the entire face of the company with its innovative, robust, and efficient business process executions. This paper conducts an in-depth review of all digital dimensions pertaining to Industry 4.0. The paper summarizes fundamental concepts, systematic literature review, journey from 1IR to 4IR, key technology drivers, and related manufacturing innovations to dive at the core of 4IR. The paper aims to lay a foundation for all researchers looking for its 360° study. The major power of the 4IR will be optimally utilized when all industries, especially SMEs—small and medium-sized start implementing it as a very industrial process. 4IR is a layered concept that unveils several key concepts at its core and at the same time, it largely forms the foundation with the future concepts like Education 4.0 and Society 5.0. The future belongs to those industries that have a digital ecosystem with technologies such as IIoT, cloud manufacturing, smart manufacturing, artificial intelligence, and machine intelligence. The paper is a technical journey for all those who are performing a technical horizontal scan for devising the new industrial dawn in this twenty-first century.

References

1. Reisman, G. (1998). Capitalism: A complete understanding of the nature and value of human economic life. Jameson Books (p. 127). ISBN 978-0-915463-73-2.
2. Kagermann, H., et al. (2013). *Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 working group*. Germany: Acatech National Academy of Science and Engineering.
3. Jasperneite, J. (2012). Was Hinter Begriffen Wie Industrie 4.0 Steckt. *Computer and Automation, 12*, 24–28.
4. Lasi, H., et al. (2014). Industry 4.0. *Business and Information Systems Engineering, 6*(4), 239–242.
5. Lu, Y. (2017). Industry 4.0: A survey on technologies, applications and open research issues. *Journal of Industrial Information Integration, 6*, 1–10. https://doi.org/10.1016/j.jii.2017.04.005.
6. Industry 4.0 – Industrial revolution.
7. Alexopoulos, K., et al. (2016). A concept for context-aware computing in manufacturing: The white goods case. *International Journal of Computer Integrated Manufacturing, 29*(8), 839–849.
8. Lu, Y. 2017b. Cyber physical system (CPS)-based industry 4.0: A survey. *Journal of Industrial Integration and Management, 2*(3). https://doi.org/10.1142/S2424862217500142.
9. Tong, J. T. (2016). *Finance and society in 21st century China: Chinese culture versus western markets* (p. 151). Boca Raton, FL: CRC Press. ISBN 978-1-317-13522-7.
10. Leurent, H., Boer, E.D., Fourth industrial revolution beacons of technology and innovation in manufacturing. White Paper, January 10, 2019. https://www.weforum.org/whitepapers/fourth-industrial-revolution-beacons-of-technology-andinnovation-in-manufacturing
11. Lichtblau, K., et al. (2015). *IMPULS-industrie 4.0-readiness*. Aachen-Köln: Impuls-Stiftung des VDMA.
12. Geissbauer, R, et al. (2016) Industry 4.0: Building the digital enterprise.
13. Schumacher, A., et al. (2016). A maturity model for assessing industry 4.0 readiness and maturity of manufacturing enterprises. *Procedia CIRP, 52*, 161–166.
14. Sanders, A., et al. (2016). Industry 4.0 implies lean manufacturing: Research activities in industry 4.0 function as enablers for lean manufacturing. *Journal of Industrial Engineering and Management, 9*(3), 811–833.
15. Doh, S. W., et al. (2016). Systems integration in the lean manufacturing systems value chain to meet industry 4.0 requirements. In M. Borsato et al. (Eds.), *Transdisciplinary engineering: Crossing boundaries* (pp. 642–650).
16. https://www.ebsco.com/
17. Lucas, R. (2003). The industrial revolution past and future.
18. Da Xu, L., et al. (2018). Industry 4.0: State of the art and future trends. *International Journal of Production Research, 56*(8), 2941–2962. https://doi.org/10.1080/00207543.2018.1444806.
19. Berlanstein, L. R. (1992). *The industrial revolution and work in nineteenth-century Europe*. New York: Routledge.
20. Feinstein, Charles (September 1998). "Pessimism perpetuated: Real wages and the standard of living in Britain during and after the industrial revolution". Journal of Economic History 58 (3): 625–658. doi:https://doi.org/10.1017/s0022050700021100.
21. Lucas, R. E. (2002). *Lectures on economic growth* (pp. 109–110). Cambridge: Harvard University Press. ISBN 978-0-674-01601-9.
22. Lucas, R. (2003). *The industrial revolution*. Minneapolis, MN: Federal Reserve Bank of Minneapolis.
23. Szreter, M. (February 1998). Urbanization, mortality, and the standard of living debate: new estimates of the expectation of life at birth in nineteenth-century British cities. *The Economic History Review, 51*(1), 104. https://doi.org/10.1111/1468-0289.00084.

24. McCloskey, D. (2004). Review of The Cambridge Economic History of Modern Britain 4.
25. Landes, D. (1969). *The unbound Prometheus*. Cambridge: Press Syndicate of the University of Cambridge. ISBN 978-0-521-09418-4.
26. Horn, et al. (2010). *Reconceptualizing the industrial revolution*. Cambridge MA: MIT Press. ISBN 978-0-262-51562-7.
27. Esposito, J. L. (Ed.). (2004). *The Islamic world: Past and present. Volume 1: Abba - Hist* (p. 174). Oxford: Oxford University Press. ISBN 978-0-19-516520-3.
28. Anthony Wrigley, E. (2018). Reconsidering the industrial revolution: England and Wales. *Journal of Interdisciplinary History, 49*(01), 9–42.
29. Ray, I. (2011). *Bengal Industries and the British industrial revolution (1757-1857)* (pp. 7–10). Abingdon: Routledge. ISBN 978-1-136-82552-1.
30. Landes, D. (1999). *The wealth and poverty of nations*. New York, NY: W.W. Norton & Company. ISBN 978-0-393-31888-3.
31. Keibek, S. A. (2016). The male occupational structure of England and Wales, 1600–1850 (PhD). In *University of Cambridge*. Cambridge.
32. Eric, H. *The age of revolution: Europe 1789–1848* (p. 27). London: Weidenfeld & Nicolson Ltd.. ISBN 0-349-10484-0.
33. Joseph, I. *Africans and the industrial revolution in England*. Cambridge: Cambridge University Press. ISBN 0-521-01079-9.
34. Berg, et al. (1992). Rehabilitating the Industrial Revolution (PDF). *The Economic History Review, 45*(1), 24–50. https://doi.org/10.2307/2598327. JSTOR2598327.
35. Hudson, P. (1992). *The industrial revolution* (p. 11). London: Edward Arnold. ISBN 978-0-7131-6531-9.
36. Gupta, B. Cotton textiles and the great divergence: Lancashire, India and shifting competitive advantage, 1600–1850 (PDF). International Institute of Social History. Department of Economics, University of Warwick. Retrieved 5 December 2016.
37. Taylor, R. (1951). The transportation revolution, 1815–1860. ISBN 978-0-87332-101-3.
38. Roe, J. W. (1916). English and American tool builders, New Haven, Connecticut. Yale University Press, LCCN16011753. Reprinted by McGraw-Hill, New York and London, 1926 (LCCN27-24075); and by Lindsay Publications, Inc., Bradley, IL. ISBN 978-0-917914-73-7.
39. Muntone, S. *Second Industrial Revolution. Education.com*. New York, NY: The McGraw-Hill Companies.
40. James, H. (1999). The second industrial revolution: The history of a concept. *Storia Della Storiografia, 36*, 81–90.
41. Smil, V. (2005). Creating the twentieth century: Technical Innovations of 1867–1914 and their lasting impact. ISBN 0-19-516874-7.
42. Landes, D. S. (1969). *The unbound prometheus: Technological change and industrial development in Western Europe from 1750 to the present* (p. 92). Cambridge, NY: Press Syndicate of the University of Cambridge. ISBN 0-521-09418-6.
43. Maxwell, C. (1911). Faraday, michael. In H. Chisholm (Ed.), *Encyclopædia Britannica* (Vol. 10, 11th ed., p. 173). Cambridge: Cambridge University Press.
44. Beaudreau, C. *Mass production, the stock market crash and the great depression*. New York, Lincoln, Shanghi: Authors Choice Press.
45. Ford, H. et al. (1922). My life and work: An autobiography of Henry Ford.
46. A nation of steel: The making of Modern America 1965–1925. Baltimore and London: Johns Hopkins University Press. ISBN 978-0-8018-6502-2.
47. Kaynak, O. (2005). The exhilarating journey from industrial electronics to industrial informatics. *IEEE Transactions on Industrial Informatics, 1*(2), 73.
48. Wilamowski,B. (2005). Welcome to the IEEE transactions on industrial informatics, a new journal of the industrial electronics society. *IEEE Transactions on Industrial Informatics, 1*(1), 1–2.
49. IFAC. (2007). Proceedings of IFAC international workshop on intelligent manufacturing systems (IMS'07), May 23, Alicante, Spain.

50. Xu, L. (2007). Editorial: Inaugural issue. *Enterprise Information Systems, 1*(1), 1–2. https://doi.org/10.1080/17517570712331393320.
51. Carcano, A., et al. (2011). A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics, 7*(2), 179–186.
52. Maddison, A. (2007). *Contours of the world economy I-2030AD*. Oxford: Oxford University Press. ISBN 978-0199227204.
53. Recommendations for implementing the strategic initiative INDUSTRIE 4.0, 2013., http://www.acatech.de/fileadmin/userupload/BaumstrukturnachWebsite/Acatech/root/de/MaterialfuerSonderseiten/Industrie4.0/FinalreportIndustrie4.0accessible.pdf.
54. The industrial internet consortium: A global nonprofit partnership of industry, government and academia, 2014. http://www.iiconsortium.org/about-us.htm.
55. Li K. Q., & Premier of the State Council of China. Report on the work of the government. In Proceedings of the 3rd session of the 12th national people's congress, March 2015.
56. Tan, W., et al. (2010). A methodology toward manufacturing grid-based virtual enterprise operation platform. *Enterprise Information Systems, 4*(3), 283–309. https://doi.org/10.1080/17517575.2010.504888.
57. Ustundag, A., & Cevikcan, E. (2017). *Industry 4.0: Managing the digital transformation*. Heidelberg: Springer.
58. Jasperneite, J. (2012). Was Hinter Begriffen Wie Industrie 4.0 Steckt. *Computer and Automation, 12*, 24–28.
59. Lasi, H., et al. (2014). Industry 4.0. Business & information. *Systems Engineering, 6*(4), 239–242.
60. Kagermann, H., et al. (2013). *Recommendations for implementing the strategic initiative Industrie 4.0: Final report of the Industrie 4.0 working group*. Germany: Acatech National Academy of Science and Engineering.
61. Hermann, M., et al. Design principles for industrie 4.0 scenarios. Proceedings of 2016 49th Hawaii international conference on systems science, January 5–8, Maui, Hawaii. https://doi.org/10.1109/HICSS.2016.488.
62. Moeuf, A., et al. (2017). The industrial management of SMEs in the era of industry 4.0. International Journal of Production Research. Published online 8 September 2017. https://doi.org/10.1080/00207543.2017.1372647.
63. Qiu, M., et al. (2006). Efficient algorithm of energy minimization for heterogeneous wireless sensor network. In *Embedded and ubiquitous computing, vol. 4096 of lecture notes in computer science* (pp. 25–34). Berlin: Springer.
64. Qiu, M., & Sha, E. (2007). Energy-aware online algorithm to satisfy sampling rates with guaranteed probability for sensor applications. In *High performance computing and communications, vol. 4782 of lecture notes in computer science* (pp. 156–167). Berlin: Springer.
65. Wan, J., et al. (2010). Fuzzy feedback scheduling algorithm based on central processing unit utilization for a software-based computer numerical control system. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 224*(7), 1133–1143.
66. Soliman F. , Youssef, A. Internet-based e-commerce and its impact on manufacturing and business operations.
67. *Industrial Management and Data Systems, 103*(8–9), 546–552, (2003).
68. Markillie, P. (2012). A third industrial revolution. Economist, Special Report.
69. Rivkin, J. (2011). *The third industrial revolution*. New York, NY: New York Times.
70. Riedl, M., et al. (2014). Cyber-physical systems alter automation architectures. *Annual Reviews in Control, 38*(1), 123–133.
71. Wan, J., et al. (2013). Enabling cyber-physical systems with machine-to-machine technologies. *International Journal of AdHoc and Ubiquitous Computing, 13*(3-4), 187–196.
72. Wan, J., et al. (2014). Contextaware vehicular cyber-physical systems with cloud support: Architecture, challenges and solutions. *IEEE Communications Magazine, 52*(8), 106–113.
73. Frazzon, M et al. (2013). Towards socio-cyber-physical systems in production networks. In Proceedings of the 46th CIRP conference on manufacturing systems (pp. 49–54).

74. Posada, J., et al. (2015). Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Computer Graphics and Applications, 35*(2), 26–40.
75. Qin, J., et al. (2016). A categorical framework of manufacturing for industry 4.0 and beyond. *Procedia CIRP, 52*, 173–117.
76. GTAI (Germany Trade & Invest). (2014). *Industries 4.0-smart manufacturing for the future*. Berlin: GTAI.
77. Sharma, N., Shamkuwar, M., & Singh, I. (2019). The history, present and future with IoT. In V. E. Balas, V. K. Solanki, R. Kumar, & M. Khari (Eds.), *Internet of things and big data analytics for smart generation. Intelligent systems reference library* (Vol. 154, pp. 27–51). Singapore: Springer.
78. Xia, F., et al. (2012). Internet of things. *International Journal of Communication Systems, 25*(9), 1101–1102.
79. Liu, W. N., et al. (2012). RFID-enabled real-time production management system for Loncin motorcycle assembly line. *International Journal of Computer Integrated Manufacturing, 25*(1), 86–99.
80. Dai, Q. Y., et al. (2012). Radio frequency identification-enabled real-time manufacturing execution system: A case study in an automotive part manufacturer. *International Journal of Computer Integrated Manufacturing, 25*(1), 51–65.
81. Qu, T., et al. (2012). A case of implementing RFID-based real-time shop-floor material management for household electrical appliance manufacturers. *Journal of Intelligent Manufacturing, 23*(6), 2343–2356.
82. Baunsgaard, V. V., & Clegg, S. R. (2015). Innovation: A critical assessment of the concept and scope of literature. In W. Selen, G. Roos, & R. Green (Eds.), *The handbook of service innovation* (pp. 5–25). Springer.: London.
83. Pang, Z., et al. (2015). Design of a terminal solution for integration of in-home health care devices and services towards the internet-of-things. *Enterprise Information Systems, 9*, 86–116.
84. Upton, J. F., & Stein, S. L. (2015). *Responder technology alert monthly (Oct-Nov 2014) (no. PNNL-24014)*. Richland, WA: Pacific Northwest National Laboratory.
85. Flügel, C., & Gehrmann, V. (2009). Scientific workshop 4: Intelligent objects for the internet of things: Internet of things—Application of sensor networking logistic. In H. Gerhäuser, J. Hupp, C. Efstratiou, & J. Heppner (Eds.), *Constructing ambient intelligence, communications in computer and information science* (Vol. 32, pp. 16–26). Berlin: Springer.
86. Yan, B., & Huang, G. (2009). Supply chain information transmission based on RFID and internet of things. In Q. Luo (Ed.), *Proceedings of the ISECS international colloquium on computing, communication, control, and management* (pp. 166–169). Sanya: IEEE.
87. Zhengxia, W., & Laisheng, X. (2010). Modern logistics monitoring platform based on the internet of things. In R. Li & Y. Wu (Eds.), *Proceedings of the international conference on intelligent computation technology and automation (ICICTA)* (pp. 726–731). Changsha: IEEE.
88. Zhou, J., et al. (2015). Security and privacy in cloud-assisted wireless wearable communications: Challenges, solutions, and future directions. *Wireless Communications, IEEE, 22*, 136–144.
89. Li, X., et al. (2011). Smart community: An internet of things application. *IEEE Communications Magazine, 49*(11), 68–75.
90. Gubbi, J., et al. (2013). Internet of things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660.
91. Shrouf, F., & Miragliotta, G. (2015). Energy management based on internet of things: Practices and framework for adoption in production management. *Journal of Cleaner Production, 100*, 235–246.
92. Manyika J, et al.. (2011) Big data: The next frontier for innovation, competition, and productivity. New York, NY: McKinsey Global Institute.
93. Monostori, L. (2014). Cyber-physical production systems: Roots, expectations and R&D challenges. *Procedia CIRP, 17*, 9–13.

94. Mourtzis, D., & Vlachou, E. (2016). Cloud-based cyber-physical systems and quality of services. *The TQM Journal, 28*(5), 704–733.
95. National Institute of Standards and Technology. Workshop report on foundations.
96. Farooq, M. U., et al. (2015). A review on internet of things (IoT). *International Journal of Computer Applications, 113*(1), 1–7.
97. De Silva, P., & De Silva, P. (2016). Ipanera: An industry 4.0 based architecture for distributed soil-less food production systems. Proceedings of the 1st manufacturing and industrial engineering symposium, Colombo, Sri Lanka.
98. Kim, J. (2017). A review of cyber-physical system research relevant to the emerging IT trends: Industry 4.0, IoT, big data, and cloud computing. *Journal of Industrial Integration and Management, 2*(3). https://doi.org/10.1142/S2424862217500117.
99. Lee, J., et al. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia CIRP, 16*, 3–8.
100. Gürdür, D., et al. (2016). Making interoperability visible: Data visualization of cyber-physical systems development tool chains. *Journal of Industrial Information Integration, 4*, 26–34. https://doi.org/10.1016/j.jii.2016.09.002.
101. Lee, J., et al. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters, 3*, 18–23.
102. Khaitan, S. K., & McCalley, J. D. (2015). Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal, 9*(2), 350–365.
103. Thames, L., & Schaefer, D. (2016). Software-defined cloud manufacturing for industry 4.0. *Procedia CIRP, 52*, 12–17.
104. Zheng, X., et al. (2014). Cloud service negotiation in internet of things environment: A mixed approach. *IEEE Transactions on Industrial Informatics, 10*(2), 1506–1515. https://doi.org/10.1109/TII.2014.2305641.
105. Wang, C., et al. (2014) "IoT and cloud computing in automation of assembly modeling systems." IEEE Transactions on Industrial Informatics 10 (2): 1426–1434. doi:https://doi.org/10.1109/TII.2014.2300346.
106. Moghaddam, M., & Nof, S. (2017). Collaborative service-component integration in cloud manufacturing. *International Journal of Production Research, 56*, 677–691. https://doi.org/10.1080/00207543.2017.1374574.
107. Thames, J. L., & Schaefer, D. (2017). Cybersecurity for industry 4.0 and advanced manufacturing environments with ensemble intelligence. In L. Thames & D. Schaefer (Eds.), *Cybersecurity for industry 4.0.1* (pp. 243–265). Berlin: Springer (Springer Series in Advanced Manufacturing).
108. Sharma, N., & Shamkuwar, M. (2019). Big data analysis in cloud and machine learning. In M. Mittal, V. Balas, L. Goyal, & R. Kumar (Eds.), *Big data processing using spark in cloud. Studies in big data* (Vol. 43, pp. 51–85). Singapore: Springer. https://doi.org/10.1007/978-981-13-0550-4_3.
109. Thames, J. L., & Schaefer, D. (2017). Industry 4.0: An overview of key benefits, technologies, and challenges. In L. Thames & D. Schaefer (Eds.), *Cybersecurity for industry 4.0.1* (pp. 1–33). Berlin: Springer (Springer Series in Advanced Manufacturing).
110. Yeshodara, N.S., Nagojappa, N.S., & Kishore, N. (2014) IEEE international conference on cloud computing in emerging markets (CCEM), https://doi.org/10.1109/CCEM.2014.7015485.
111. Hashim, J. (2007). Information communication technology (ICT) adoption among SME owners in Malaysia. *International Journal of Business and Information, 2*(2), 221–240.
112. Bloom, N., et al. (2014). The distinct effects of information technology and communication technology on firm organization. *Management Science, 60*(12), 2859–2885.
113. Colin, M., et al. (2015). Information and communication technology as a key strategy for efficient supply chain management in manufacturing SMEs. *Procedia Computer Science, 55*, 833–842.

114. Ketteni, E., et al. (2015). Information and communication technology and foreign direct investment: Interactions and contributions to economic growth. *Empirical Economics, 48*(4), 1525–1539.
115. Lee, J., et al. (2015). A cyber-physical systems architecture for industry 40-based manufacturing systems. *Manufacturing Letters, 3*, 18–23.
116. Brizzi, P., et al. (2013). Bringing the internet of things along the manufacturing line: a case study in controlling industrial robot and monitoring energy consumption remotely. In Emerging technologies & factory automation (ETFA) (pp. 1–8).
117. Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 28*(1), 75–86.
118. Liu, Q., et al. (2014). Cloud manufacturing service system for industrial-cluster-oriented application. *Journal of Internet Technology, 28*(1), 373–380.
119. Lee, K. (2016). Artificial intelligence, automation, and the economy. The White House Blog.
120. Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters, 3*, 18–23.
121. Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2233–2243.
122. Lee, J., et al. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters, 1*(1), 38–41.
123. Shi, J., et al. (2011). A survey of cyber-physical systems. In Wireless communications and signal processing (WCSP), 2011 international conference on IEEE (pp. 1–6).
124. Lee, J., et al. (2015). Industrial big data analytics and cyberphysical systems for future maintenance & service innovation. *Procedia CIRP, 38*, 3–7.
125. Zhang, L., et al. (2014). Cloud manufacturing: A new manufacturing paradigm. *Enterprise information system, 8*(2), 167–187.
126. Wu, D., et al. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems, 32*(4), 564–579.
127. Yang, S., et al. (2015). A unified framework and platform for designing of cloud-based machine health monitoring and manufacturing systems. *Journal of Manufacturing Science and Engineering, 137*(4), 040914.
128. Baheti, R., & Gill, H. (2011). Cyber-physical systems. *Impact of Control Technology, 12*(1), 161–166.
129. Leitao, P., et al. (2016). Smart agents in industrial cyber–physical systems. *Proc IEEE 2016, 104*(5), 1086–1101.
130. Tuptuk, N., & Hailes, S. (2018). Security of smart manufacturing systems. *Journal of Manufacturing Systems, 47*, 93–106.
131. Grieves, M. (2014). Digital twin: Manufacturing excellence through virtual factory replication.
132. Qi, Q., & Tao, F. (2018). Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison. *IEEE Access, 6*, 3585–3593.
133. Schleich, B., et al. (2017). Shaping the digital twin for design and production engineering. *CIRP Annals, 66*(1), 141–144.
134. Sharma, N., Patil, M., & Shamkuwar, M. (2019). Why big data and what is it? Basic to advanced big data journey for the medical industry. In V. E. Balas, L. H. Son, S. Jha, M. Khari, & R. Kumar (Eds.), *Internet of things in biomedical engineering* (1st ed., pp. 189–212). Cambridge: Academic Press, Elsevier, Science Direct.
135. Rich, S. (2012). Big data is a "new natural resource." IBM says.
136. Lee, J., et al. (2013). (2013) recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters, 1*(1), 38–41.
137. Barton, D., & Court, D. (2012). Making advanced analytics work for you. *Harvard Business Review 2012, 90*(10), 78–83,128.
138. Perrey, J., et al. (2013). Smart analytics: How marketing drives shortterm and long-term growth. In D. Court, J. Perrey, T. McGuire, J. Gordon, & D. Spillecke (Eds.), *Big data, analytics, and the future of marketing & sales*. New York, NY: McKinsey & Company.

139. http://www.wellgrounded.com.au/wp/innovation/why-is-innovation-important/
140. https://blog.marketresearch.com/the-top-7-things-to-know-about-smart-manufacturing
141. http://automationexcellence.com/Overview.html
142. http://www.performance-ideas.com/2016/07/13/industry-4-0-big-data/
143. Ramdasi, P., & Prasad R. (2018). Industry 4.0: Opportunities for analytics, conference: 2018 IEEE Punecon. https://doi.org/10.1109/PUNECON.2018.8745382.
144. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-018-0162-3
145. https://en.wikipedia.org/wiki/Industrial_internet_of_things
146. https://en.wikipedia.org/wiki/Cobot
147. https://en.wikipedia.org/wiki/Cobot
148. *International Journal of Production Research, 57*(15–16):4854–4879. https://doi.org/10.1080/00207543.2018.1449978
149. Ghosh, A., et al. (2019). Industrial IoT networks powered by 5G new radio. *Microwave Journal, 62*(12), 24–40. 9 p.
150. Roblek, V., Meško, M., & Krapež, A. (2016). A complex view of Industry 4.0. *SAGE Open, 6*(2), 2158244016653987.

A Study of Resource Management and Security-Based Techniques in Autonomic Cloud Computing

Neha Agrawal, Rohit Kumar, and Pankaj Kumar Mishra

1 Introduction

Distributed systems have always been in focus of the research community due to its large-scale devices, complex functioning, hard management, synchronization need, etc. [1]. Public cloud is a major example of the distributed system. Cloud computing has been a dominating technology over the past years. It offers hardware and/or software in the service form over the Internet on the user-demand. The cloud services can be software applications (software-as-a-service), platform for the deployment and execution of applications (platform-as-a-service), and hardware infrastructure (infrastructure-as-a-service) [2, 3]. The revolutionary paradigms of the cloud range from the utility computing, auto-scaling, processing on the fly, etc., to the network function chaining, virtualization, live Virtual Machine (VM) migration, etc. Cloud computing provides different resources/facilities in the form of services, and releases user from the burden of resource setup, environment monitoring, regular updates, etc. In return, cloud users pay money for the measured services as per the service level agreement (SLA) [4]. But, this ease of use does not come free and cloud owner need to maintain a heavy setup with the help

N. Agrawal (✉)
CSE Group, IIIT Sri City, Chittoor, AP, India

R. Kumar
Department of CSE, DSPM-IIIT Naya Raipur, Chhattisgarh, India
e-mail: rohit@iiitnr.edu.in

P. K. Mishra
Mathematics Division, Chandigarh University, Chandigarh, India

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8_20

of different tools and techniques. This complex management needs an intelligent orchestration process, and an automatic mechanism is further needed to improve the cloud management.

Due to the heterogeneity, dynamism, and failures in cloud computing, resource allocation is a tedious task [5]. Autonomic computing comes for rescue in such difficult situations, where services need to be delivered to the intended users with minimum guaranteed level of quality-of-service (QoS) as per the SLA. Autonomic computing assures the delivery of different services in a dynamic heterogeneous environment in an autonomous way [6]. The merge of these two promising technologies, namely cloud computing and autonomous computing, gives rise to a new technological domain called autonomous cloud computing (ACC). ACC technology allows efficient resource allocation while satisfying the QoS requirements. This technology automatically heals the unexpected failures at run-time and thus optimizes the QoS parameters [2]. Due to the distinctive features of these technologies, users get felicitated with the benefits such as improved network management, high throughput, low workload completion time, and improved QoS. ACC aims to achieve the properties such as self-protection, self-optimization, self-healing, and self-configuration.

The conceptual view of ACC is shown in Fig. 1. As shown in Fig. 1, the cloud service users may be of different types, demanding different types of cloud services depending on the application type. All users communicate with the

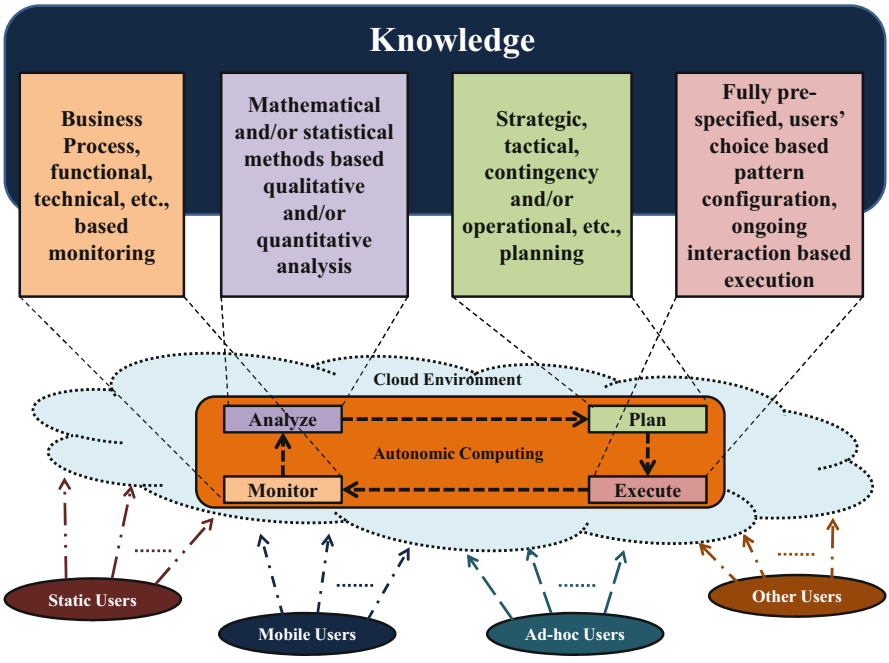

Fig. 1 Conceptual view of autonomic cloud computing

cloud provider using different technologies and access the services of ACC. The autonomic behavior of the cloud allows to automate the users' need according to the application-specifications. The four essential components of an autonomic structure are included in the cloud environment, namely *monitor, analyze, plan*, and *execute*. In addition to this, each of these modules has its in-built functions that help to operationalize the cloud environment in an autonomous way.

The *monitor* module offers the services that help to collect the essential properties of any business process, functional element, technical component, etc. The *analyze* module allows to perform the mathematical and/or the statistical operations to evaluate the collected metric from monitoring phase. The evaluation may be focused toward the quantitative or the qualitative analysis depending on the type of application and/or the service. The *plan* module provides a function of the strategic, tactical, contingency, and/or operational planning. Similar to the *analyze* module, the functionality of this module also depends on the application type and/or the service need. Finally, the *execute* module gives the options of different types of service execution. The process execution may be based on the users' specifications fully described in advance, or it may depend on some pattern. Additionally, the execution model may choose to run in parallel with the on-going interaction with the users.

Resource management involves different operations/activities like resource provisioning and resource scheduling, etc. But, in an ACC environment, the autonomicity of the operations becomes important, especially at run-time. Autonomous resource availability, provisioning, and scheduling are the focus points of this work. These ACC characteristics are analyzed from a security point of view in this work, and the relevant comparative analysis of the respective solutions is also provided.

This work offers the following major contributions.

- Different resource management techniques in an ACC environment are discussed.
- Security aspects in resource management techniques are explored.
- The challenges in the autonomic security analysis of the resource management are detailed, and are highlighted from a user's point of view.
- The comparisons of the relevant existing works are also provided.

The following chapter is organized as follows. Section 2 presents the related work on the resource management in ACC including the focus on resource provisioning and scheduling. The security-based analysis of the resource management techniques in ACC is performed in Sect. 3. Finally, Section 4 presents the conclusion of the work with the relevant possible future aspects.

2 Related Work

Plethora research is available on resource management in ACC. In this section, discussion of the relevant literature is provided as follows.

As the cloud services are increasing day-by-day, designing of efficient resource provisioning/management frameworks is essential in cloud computing. In dynamically changing workload cloud environment, resource management plays a key role. Lack of proper management technique in such environment may leads to the degradation in guaranteed QoS to the cloud customers. Several solutions have been proposed for resource provisioning and scheduling in ACC. The description of the noteworthy approaches is given below.

A service provisioning model using Nash Equilibrium is described in [4]. In this work, the authors proposed a distributed algorithm for run-time resource allocation. The resources are dynamically allocated based on the workload fluctuations. The performance of the approach is evaluated in a simulated environment and the results show that the efficiency is improved by 50–70%. An uncertainty-aware and role-based resource provisioning model for cloud computing is proposed in [7]. The proposed approach supports mobility, self-healing, self-organization, and self-optimization. The authors evaluated the performance in simulated and experimental environments. Results prove that the approach provides improved service quality and response time. To provide programming and infrastructure support to end-to-end applications, a framework named as CometCloud is proposed in [8]. The framework is autonomic and allows the federation of geographically located data in an on-demand fashion, and computes the resources. The framework consists of three layers—(1) federation/infrastructure, (2) autonomic management, and (3) interface/programming. The first layer manages dynamic federation of resources and offers essential services. The second layer provides resource provisioning and execution of application workflow. The third layer provides interfaces between different resources and application resources.

For dynamic dataflows, a resource provisioning model is described in [9]. This application model provides run-time provisioning of resources to improve QoS, execution cost, and scalability. The authors propose two greedy heuristics algorithms, namely sharded and centralized, and compare with genetic algorithm. Results show that the model reduces the cost by 27.5% without compromising the QoS and scalability. For provisioning of heterogeneous cloud resources, an approximation mechanism is proposed in [10]. In this the strategy-proof optimal and approximation mechanisms are designed. The PTAS mechanism calculates the payment that users have to pay for the used resources. To evaluate the performance of the proposed mechanism, real-workload traces have been used. PTAS is adaptive and leads the system into the equilibrium state. An uncertainty-aware framework for resource and data management is proposed in [11]. The framework works in two phases, captures the data in the first phase and performs the optimization in the second phase. The results show that the optimization improves the task allocation compared to the single-objective schemes with regard to the battery usage, network load, etc.

To deal with massive data in hybrid cloud, a framework is described in [12]. This framework autonomously tunes the query parameters and in-memory data structure

which provides efficient retrievals and also minimizes the resource consumption. To manage uncertainty in the cloud, a control-theoretic approach is proposed in [13]. In this approach, the online learning mechanism is integrated with fuzzy cloud controller. The major uncertainty sources and the challenges in elasticity management is also presented. For request scheduling and resource management, an optimal theoretical model for multi-agent cloud system is proposed in [14]. The authors discuss the genetic algorithm for finding the optimal solution for resource scheduling. The performance of the model is evaluated using energy consumption and reliability metrics. For each metric, a sub-model is created and the connection between them is established using Bayesian method. An adaptive and fuzzy resource management technique for dynamic workload cloud environment is proposed in [15]. In this approach, the sensors are used to gather the last resource value for each virtual machine and forwarded to the fuzzy controller. The received information is then analyzed and helps in making the allocation decision. Based on the workload, the membership function are dynamically updated to meet the QoS requirements. Results show that the approach outperforms the static-fuzzy and role-based approaches.

Having the increasing demand of cloud services, data-centers face energy consumption issues. To address this, cloud service provider use renewable sources to reduce the carbon emission. In [16], the authors perform a thorough analysis of trade-off between the energy consumption and performance of cloud applications. Then a method named as SaaScalar is proposed which implements several application controllers (control-loop-based) to meet the performance metrics. A feedback-based autonomic provisioning approach for MapReduce systems is proposed in [17]. It is a control-theoretic approach based on the existing techniques. The dynamic models for MapReduce systems are introduced and two control use cases are also identified. Even feedback and classical feedback controller improve the cluster reconfiguration process. The performance is validated online using 60 nodes MapReduce cluster.

A profile-based resource allocation scheme is described in [18]. This scheme alleviates the mapping process used by the service provider during resource allocation. The performance is evaluated using different virtual function with varying resource configurations and workloads. To maximize the profit, an autonomic resource provisioning approach is also described in [19]. In this, an optimization module is used which further uses cost and revenue models. Authors validate the performance in hybrid cloud and the results demonstrate that the substantial profit is achieved. A review of resource management techniques in autonomic cloud computing is also given in [5]. The authors discuss various resource provisioning techniques and also provide their comparative analysis. From the study, it has been observed that the QoS parameters can be improved if the in-advance resource reservation is done. The performance can be improved significantly using proper mapping of resources and workloads. It is also stated that a work-load aware resource allocation approach can improve the resources utilization (Table 1).

Table 1 Comparison of the existing resource management techniques in ACC

S. no.	Author(s) and year	Major contributions
1.	Ardagna et al. (2013) [4]	•Use of Nash gaming for service provisioning•Use of best-reply dynamics •Resource allocation policies in a real prototype
2.	Viswanathan et al. (2015) [7]	•Supports mobility, self-healing, self-optimization, resource provisioning, and self-organization
3.	Diaz et al. (2015) [8]	•Support for programming and infrastructure •Software-defined cyber-infrastructure based synthesis •distributed resources based on-demand federation in an autonomous way
4.	Kumbhare et al. (2015) [9]	•"Dynamic dataflows" based concept using different tasks to control the dataflow's cost and QoS •Optimization problem for run-time resource provisioning and deployment to balance the application's QoS •Greedy heuristic based proposal and comparison against a solution based on Genetic Algorithm to achieve near-optimal solution
5.	Mashayekhy et al. (2015) [10]	•Addresses autonomic VM allocation and provisioning problem •The mechanism tries to calculate the users' payment based on the resource-usage •Proposed approximation mechanism is a polynomial-time approximation scheme (PTAS)
6.	Viswanathan et al. (2016) [11]	•Proposes a workflow scheme for a wide variety of data •Proposes an uncertainty-based unified method for resource and data management
7.	Malensek et al. (2016) [12]	•Proposes a hybrid cloud-based storage model for in-memory data structure tuning. •Tries to obtain minimal resource consumption and efficient retrievals to address the problems of remote sensors
8.	Jamshidi et al. (2016) [13]	•Fuzzy cloud controller based online learning mechanism is discussed •The dynamic resource provisioning based uncertainty is also discussed
9.	Singh et al. (2017) [5]	•Discusses resource allocation challenges •Broad analysis on cloud-resource management •Discusses autonomic resource provisioning and scheduling
10.	Sun et al. (2017) [14]	•Proposes a correlation model of energy, performance, and reliability •Tries to capture the failures and recovery-based effects of the resource. •An optimization model is also developed the trade-off evaluation between energy consumption and performance •Proposed a genetic algorithm based design to implement the global request scheduling based solution
11.	Haratian et al. (2017) [15]	•Discusses QoS, SLA, and resource management strategies •Proposes a fuzzy resource management framework to reduce the number of SLA violations involving
12.	Hasan et al. (2017) [16]	•Evaluates a trade-off of the energy consumption and performance •An auto-scaler is also discussed to implement the control loop satisfying the performance and resource metrics
13.	Berekmeri et al. (2018) [17]	•Proposes an approach for a coarse grained control in different use cases •Identifies two use cases for minimal resources usage and strict performance
14.	Van et al. (2019) [18]	•Proposes a VNF profile to simplify the process of resource allocation mapping •Compares different methods for deriving a model from profiled datasets
15.	Beigi et al. (2020) [19]	•Proposes an autonomic way to manage the cloud-based web services. •Using software-defined features, the solution optimizes resource allocation

3 Security-Aware Resource Management Techniques in ACC

Since the heterogeneous cloud resources are geographically located, security-aware resource provisioning techniques are required to handle various security risks. The abovementioned resource provisioning approaches are unable to defend the systems against the security attacks. To offer the secure cloud services, security aware resource provisioning techniques are needed. The description of the relevant approaches is provided in this section.

An autonomic mobile cloud resource/service management framework for ad hoc cloud is discussed in [20]. The approach works in two modes—mobile and static. The privacy and security issues in ad hoc cloud computing are also provided. A security architecture is developed to study the defense approaches in autonomic cloud. Based on this study, it has been observed that the ad hoc cloud improves the efficiency and reduces the cost. A self-protection resource provisioning approach named as SECURE is described in [21]. The approach offers self-protection against the various security attacks and thus ensures the availability of cloud services. The performance of this approach has been tested on SNORT. Results show that the approach provides improved performance with regard to the attack detection and false positive rates. In this work, the impact of security on QoS has also been analyzed. Due to large-scale and high server density, cloud data centers are prone to several attacks, failures, and mis-configurations. These unexpected events may lead to thermal anomalies such as coldspots, hotspots, and fugues. To address this, a thermal anomaly detection approach is proposed in [22] which compares the observed and expected thermal images of data centers. Additionally, to improve the detection accuracy, a thermal anomaly aware resource allocation mechanism is also described. Experimental results prove that this approach outperforms the existing anomaly detection approaches.

Elasticity feature of cloud allows the service provider to manage the dynamic workload using provisioning and de-provisioning of computing resource autonomously. In [23], authors describe how to achieve elasticity in cloud firewalls. The aim of elasticity here is to achieve the guaranteed performance using less number of firewall instances. This chapter determines the number of firewall instances to satisfy the dynamic traffic load. For this, an analytical model based on the queuing theory and Markov model is developed. The model is then validated using discrete-event simulation and Amazon Web Service (AWS) cloud platform. Results show that this model can be used in fluctuating workload to achieve the guaranteed performance.

An anomaly detection and autonomic optimization approach is proposed in [24]. In this, the behavior of normal users is normalized and then hierarchical matching and blacklist methods are used to identify the abnormal users. Experimental results show that this approach provides 6.9 and 5.3% higher classification accuracy compared to the traditional approaches. Similarly, to identify the abnormal behavior Kolmogorov Complexity metric is used in [25]. A secure resource management approach for IaaS cloud is proposed in [26]. It offers in-depth security

by determining—(1) grouping of VMs based on the similarity in reachability requirements, (2) VMs deployment to diminish the security risks. The allocation problem is formalized using constraint satisfaction problem (CSP) and can be solved using satisfiability modulo theories (SMT) solvers. Results show that the approach reduces risk and improves manageability of VMs' security configurations.

To provide high availability of computing resources, a failure aware resource management approach is described in [27]. A re-configurable distributed VM infrastructure is designed. For the construction and re-configuration of framework, node selection strategy is proposed. To calculate the nodes' reliability status, failure management techniques are used. In the selection process, the author considers the performance and reliability status of nodes are used. Experimental results show that the job and task completion rates have been increased by 17.6 and 91.7%, respectively.

A self-protection multilayered architecture, named VESPA, is proposed in [28], for cloud computing resources. It is policy-based and manages security at two levels—within and across the infrastructural layers. Coordination among the self-protection loops helps in the detection and reaction for the cross layers. It has been observed from the experimental results that the architecture is suitable and effective for self-protecting the cloud resources. Another self-protection scheme is described in [29] for protecting the cloud data. The scheme uses active data bundles for self-protecting the data, cipher-text policy attribute based encryption for access control, and RSA for key management. Results show the performance of scheme is acceptable (Table 2).

Table 2 Comparison of the security-aware resource management techniques in ACC

S. no.	Author(s) and year	Major contributions
1.	Fu (2010) [27]	•To enhance system availability, a failure aware resource management approach is proposed. •Re-configurable distributed VM infrastructure is designed. •Node selection strategy is used for construction and re-configuration of framework. •Job completion and task completion rates are increased by 17.6 and 91.7%, respectively
2.	Wailly et al. (2012) [28]	•Multilayered self-protecting architecture, named VESPA, is proposed. •Policy-based and manages security at two levels. •Results show that the design is effective for self-protecting the cloud resources
3.	Prangishvili et al. (2013) [25]	•Kolmogorov Complexity metric is used to identify the abnormal behavior
4.	Haj et al. (2013) [26]	•Security aware resource management scheme is described. •Resource allocation scheme is formalized using constraint satisfaction problem (CSP) and solved using satisfiability modulo theories (SMT). •It reduces the security risks and improves the manageability of VMs' security

(continued)

Table 2 (continued)

S. no.	Author(s) and year	Major contributions
5.	Shila et al. (2017) [20]	•Generic autonomic mobile cloud (AMCloud) management framework •Automatic and efficient service management of ad hoc cloud in mobile and static modes •Possible security and privacy issues are detailed
6.	Salah et al. (2017) [23]	•Determines the firewalls required to dynamically adjust against the incoming traffic. •Also discusses a Markov chains and queuing theory based model to capture the behavior of the firewalls. •The behavior comprises of virtual firewalls and a load-balancer
7.	Sarhan and Carr (2017) [29]	•A self-protection data scheme is discussed. •It uses active data bundles, cipher-text ABE, and RSA techniques. •Offers acceptable performance
8.	Gill and Buyya (2018) [21]	•Discusses self-protection approach called SECURE. •Ensures the availability of the services to the authentic users
9.	Lee et al. (2018) [22]	•Discusses Anomaly-aware Resource Allocation •Discusses model-based thermal anomaly detection mechanism
10.	Zheng et al. (2018) [24]	•Proposes an abnormal behavior analysis model for mobile cloud environment. •User behavior mapped to a sequence of same length, offset, and amplitude. •Discusses a pattern growth method for autonomous optimization

4 Conclusion and Future Work

ACC enables the cloud providers to manage the cloud services efficiently in an autonomous manner. In this study, firstly the resources management techniques in cloud computing are discussed. Secondly, the security-based methods for resource management are explored. The comparative analysis of the discussed approaches for each category is also presented.

The future work includes to present detailed taxonomy of ACC in cloud computing and to discuss research challenges and open research problems.

References

1. van Steen, M., & Tanenbaum, A. S. (2016). A brief introduction to distributed systems. *Computing, 98*, 967–1009.
2. Buyya, R., Calheiros, R. N., & Li, X. (2012). Autonomic cloud computing: Open challenges and architectural elements. In *2012 IEEE 3rd International Conference on Emerging Applications of Information Technology* (pp. 3–10).
3. Agrawal, N., & Tapaswi, S. (2019). Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges. *IEEE Communications Surveys and Tutorials, 21*(4), 1–27.

4. Ardagna, D., Panicucci, B., & Passacantando, M. (2013). Generalized Nash equilibria for the service provisioning problem in cloud systems. *IEEE Transactions on Services Computing, 6*(4), 429–442.
5. Singh, S., Chana, I., & Singh, M. (2017). The journey of QoS-aware autonomic cloud computing. *IT Professional, 19*(2), 42–49.
6. Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer, 36*(1), 41–50.
7. Viswanathan, H., Lee, E. K., Rodero, I., & Pompili, D. (2015). Uncertainty-aware autonomic resource provisioning for mobile cloud computing. *IEEE Transactions on Parallel and Distributed Systems, 26*(8), 2363–2372.
8. Diaz-Montes, J., AbdelBaky, M., Zou, M., & Parashar, M. (2015). CometCloud: Enabling software-defined federations for end-to-end application workflows. *IEEE Internet Computing, 19*(1): 69–73.
9. Kumbhare, A. G., Simmhan, Y., Frincu, M., & Prasanna, V. K. (2015). Reactive resource provisioning heuristics for dynamic dataflows on cloud infrastructure. *IEEE Transactions on Cloud Computing, 3*(2), 105–118.
10. Mashayekhy, L., Nejad, M. M., & Grosu, D. (2015). A PTAS mechanism for provisioning and allocation of heterogeneous cloud resources. *IEEE Transactions on Parallel and Distributed Systems, 26*(9), 2386–2399.
11. Viswanathan, H., Lee, E. K., & Pompili, D. (2016). A multi-objective approach to real-time in-situ processing of mobile-application workflows. *IEEE Transactions on Parallel and Distributed Systems, 27*(11), 3116–3130.
12. Malensek, M., Pallickara, S., & Pallickara, S. (2016). Autonomous cloud federation for high-throughput queries over voluminous datasets. *IEEE Cloud Computing, 3*(3), 40–49.
13. Jamshidi, P., Pahl, C., & Mendonça, N. C. (2016). Managing uncertainty in autonomic cloud elasticity controllers. *IEEE Cloud Computing, 3*(3), 50–60.
14. Sun, P., Dai, Y., & Qiu, X. (2017). Optimal scheduling and management on correlating reliability, performance, and energy consumption for multiagent cloud systems. *IEEE Transactions on Reliability, 66*(2), 547–558.
15. Haratian, P., Safi-Esfahani, F., Salimian, L., & Nabiollahi, A. (2019). An adaptive and fuzzy resource management approach in cloud computing. *IEEE Transactions on Cloud Computing, 7*(4), 907–920.
16. Hasan, M.S., Alvares, F., Ledoux, T., & Pazat, J. (2017). Investigating energy consumption and performance trade-off for interactive cloud application. *IEEE Transactions on Sustainable Computing, 2*(2), 113–126 (2017)
17. Berekmeri, M., Serrano, D., Bouchenak, S., Marchand, N., & Robu, B. (2018). Feedback autonomic provisioning for guaranteeing performance in mapreduce systems. *IEEE Transactions on Cloud Computing, 6*(4), 1004–1016.
18. Van Rossem, S., Tavernier, W., Colle, D., Pickavet , M., & Demeester, P. (2019). Profile-based resource allocation for virtualized network functions. *IEEE Transactions on Network and Service Management, 16*(4), 1374–1388.
19. Beigi-Mohammadi, N., Shtern, M., & Litoiu, M. (2020). Adaptive load management of web applications on software defined infrastructure. *IEEE Transactions on Network and Service Management, 17*(1), 488–502.
20. Shila, D. M., Shen, W., Cheng, Y., Tian, X., & Shen, X. S. (2017). AMCloud: Toward a secure autonomic mobile Ad Hoc cloud computing system. *IEEE Wireless Communications, 24*(2), 74-81.
21. Gill, S. S., & Buyya, R. (2018). SECURE: Self-protection approach in cloud resource management. *IEEE Cloud Computing, 5*(1), 60–72.
22. Lee, E. K., Viswanathan, H., & Pompili, D. (). Model-based thermal anomaly detection in cloud datacenters using thermal imaging. *IEEE Transactions on Cloud Computing, 6*(2), 330–343.
23. Salah, K., Calyam, P., & Boutaba, R. (2017). Analytical model for elastic scaling of cloud-based firewalls. *IEEE Transactions on Network and Service Management, 14*(1), 136–146.

24. Zheng, R., Zhu, J., Zhang, M., Wu, Q., Liu, R., Liu, K., et al. (2018). An anomaly recognition and autonomic optimization method to user's sequence behaviors for D2D communications in MCC. *IEEE Access, 6*, 63005–63020.
25. Prangishvili, A., Shonia, O., Rodonaia, I., & Rodonaia, V. (2013). Formal security modeling in autonomic cloud computing environment. In *WSEAS/NAUN International Conferences, Valencia*.
26. Al-Haj, S., Al-Shaer, E., & Ramasamy, H. V. (2013). Security-aware resource allocation in clouds. In *2013 IEEE International Conference on Services Computing* (pp. 400–407).
27. Fu, S. (). Failure-aware resource management for high-availability computing clusters with distributed virtual machines. *Journal of Parallel and Distributed Computing, 70*(4), 384–393.
28. Wailly, A., Lacoste, M., & Debar, H. (2012). Vespa: Multi-layered self-protection for cloud resources. In *Proceedings of the 9th International Conference on Autonomic Computing* (pp. 155–160).
29. Sarhan, A. Y., & Carr, S. (2017). A highly-secure self-protection data scheme in clouds using active data bundles and agent-based secure multi-party computation. In *2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud)* (pp. 228–236).

Index

T. Choudhury et al. (eds.), *Autonomic Computing in Cloud Resource Management in Industry 4.0*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-71756-8