EAI/Springer Innovations in Communication and Computing

Rajganesh Nagarajan
Pethuru Raj
Ramkumar Thirunavukarasu *Editors*

Operationalizing Multi-Cloud Environments

Technologies, Tools and Use Cases

EAI
RESEARCH MEETS INNOVATION

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

Rajganesh Nagarajan • Pethuru Raj
Ramkumar Thirunavukarasu
Editors

Operationalizing Multi-Cloud Environments

Technologies, Tools and Use Cases

Springer

EAI
RESEARCH MEETS INNOVATION

Editors
Rajganesh Nagarajan
Sri Venkateswara College of Engineering
Sriperumbudur, India

Pethuru Raj
Reliance Jio Infocomm. Ltd., RJIL
Bangalore, India

Ramkumar Thirunavukarasu
Vellore Institute of Technology
Vellore, India

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-030-74401-4 ISBN 978-3-030-74402-1 (eBook)
https://doi.org/10.1007/978-3-030-74402-1

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

Preface

Multi-cloud development is driven by a need to provide highly scalable and reliable applications to meet business goals that are difficult to achieve by using private-only or hybrid cloud architectures. The benefits of multi-cloud would offer promising solutions for critical issues that include selection of best class of services, avoiding vendor lock-in, cost-performance optimization, data privacy and compliance, and others. These solutions would solve complex computation issues faced by various stakeholders. Similarly, the emergence of intelligent cloud brokers with the self-healing properties would simplify the operations and fulfill the expectations of cloud users in a typical multi-cloud environment.

The core idea behind the book is to extract the infrastructural platform and process-based solutions for strengthening the expected service outcomes for multi-cloud environments. The solutions offered by intelligent cloud broker and their unsolved issues are highlighted by the contributors for examining and implementing next level solutions which are suitable for multi-cloud environment. Further, the core theoretical concepts and underlying reference architectures of multi-cloud environment has been exposed and illustrated by the contributors. Ultimately, the security issues in the multi-cloud environment have been given a higher priority while preparing the book. Various practices, such as identity access, authentication, and authorization, and SLAs are focused to strengthen the proposal and extract novel ideas and solutions from the contributors in this niche area.

The multi-cloud architecture adopted for effective deployment of big data and Internet of Things (IoT)-based real-time projects are also explored in the book. This book would be helpful to the researchers, cloud developers in industry, faculty members, students, and various academic, industrial fraternities who are interested in cloud computing and working in multi-cloud environment. The prerequisite knowledge required are basis of distributed computing, Internet of computing, service-oriented architecture, grid, and cloud computing. The level of the book is introductory and advance one.

Sriperumbudur, India — Rajganesh Nagarajan
Bangalore, India — Pethuru Raj
Vellore, India — Ramkumar Thirunavukarasu

About This Book

The cloud idea is on fast track and is tending towards the multi-cloud phenomenon, which is being projected as the next big thing in the mesmerizing cloud journey. Multiple cloud environments (private, public, and edge) have been leveraged by worldwide business enterprises towards gaining a number of business, technology, and user advantages. Setting up and sustaining multi-cloud environments are beset with a number of challenges and concerns. Cloud experts and exponents from the IT industry and academic institutions have come out with a number of ways and means for fulfilling multi-cloud strategy. In this book, we have incorporated well-written chapters on the various aspects of multi-cloud paradigm.

The initial portion of the book focuses on the motivations for the industry to implement multi-cloud option for distinct business and technological use cases. The second part of the book addresses the challenges involved in setting up and sustaining multi-cloud environments. There are business and technology-centric challenges in embracing and operating multi-cloud centres. The third portion focuses on the next-generation technologies and tools along with multi-cloud platforms, processes, patterns, and practices. The privacy and security issues with respect to the multi-cloud environment have been focused in the penultimate section. The final segment of the book is dedicated for cloud brokerage systems, where various traits and tenets of cloud brokerage services especially for accomplishing cloud intermediation, integration, orchestration, governance, security, management, configuration, etc. have been explained in detail through a few chapters.

Contents

About the Editors

N. Rajganesh is presently working as an Associate Professor in the Department of Computer Science and Engineering, Sri Venkateswara College of Engineering, Tamilnadu, India. He obtained PhD degree from Anna University during the year 2018 for his thesis entitled "Fuzzy based Intelligent Semantic Cloud Service Discovery for Effective Utilization of Services". Having 15.5 years of experience in Teaching, he has contributed to research findings in various reputed international Journals. During his career, he has attended more than 20 Faculty Development Program/Workshop/Seminar, which are sponsored by AICTE, UGC, ISTE, and Anna University. He has functioned as resource person in more than 10 Faculty Development Programme and organized seminars and workshops. He is functioning as an active reviewer for top-notch journals from IEEE, Springer, Elsevier, and other publishers.

Pethuru Raj is working as the chief architect and vice-president in the Site Reliability Engineering (SRE) division of Reliance Jio Platforms Ltd. Bangalore. His previous stints are in IBM global Cloud Center of Excellence (CCoE), Wipro consulting services (WCS), and Robert Bosch Corporate Research (CR). In total, he has gained more than 19 years of IT industry experience and 8 years of research experience. Having finished the CSIR-sponsored PhD degree at Anna University, Chennai, he continued with the UGC-sponsored postdoctoral research in the Department of Computer Science and Automation, Indian Institute of Science (IISc), Bangalore. Thereafter, he was granted a couple of international research fellowships (JSPS and JST) to work as a research scientist for 3.5 years in two leading Japanese universities. He published more than 30 research papers in peer-reviewed journals such as IEEE, ACM, Springer-Verlag, and Inderscience. He has authored and edited 20 books thus far and contributed 35 book chapters for various technology books edited by highly acclaimed and accomplished professors and professionals. He focuses on some of the emerging technologies such as the Internet of Things (IoT), Artificial Intelligence (AI), Big and Fast Data Analytics, Blockchain, Digital

Twins, Cloud-native computing, Edge/Fog Clouds, Reliability Engineering, Microservices architecture (MSA), and Event-driven Architecture (EDA). His personal web site is at www.tinyurl.com/peterindia.

T. Ramkumar is presently working as Associate Professor and Head of the Department in School of Information Technology and Engineering, VIT, Vellore, India. He obtained PhD degree from Anna University during the year 2010 for his thesis entitled "Synthesizing Global Association Rules in Multi-Database Mining". Having 19 years of experience in higher education and research, he regularly contributes to research findings in various reputed international journals. Some of his research works have been published in journals from Springer, Elsevier, Wiley, World-Scientific, and other publishers. He has functioned as resource person in more than 20 faculty development programmes and organized seminars and workshops which are funded by ISTE, AICTE, and others. He has successfully guided three research scholars and led to PhD degree.

Part I
Multi-cloud Environments for Enterprise-Grade Applications

Chapter 1
Invocation of Multi-Cloud Infrastructure Services in Web-Based Semantic Discovery System

B. Bazeer Ahamed and Murugan Krishnamoorthy

1.1 Introduction

Cloud computing as a bobbing up imaginative innovation has affected the Information Technology (IT) enterprise to a large degree, where the leading companies like Google, Amazon and Microsoft have normal cloud innovation to supply perceptive and sturdy cloud administrations. The most enticing highlights like on-request self-administration, pervasive organization access, vicinity free asset pooling, rapid flexibility, and pay per use have made cloud advantages more attractive [1, 2]. Cloud computing has led to the expansion of distributed computing technology by virtue of the advanced Internet services which tend to complement the characteristics pertaining to distributed computing environment emerging from World Wide Web, Grid computing as nicely as shared organizations. To be exact, distributed computing local weather gives large scope foundations to most beneficial figuring these can modify steadily to the consumer simply as utility prerequisites.

An improvement vary of pay-per-use cloud administrations are presently available on the net as Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). With the increase in the number of services, there has also been an increase in demand and adoption of cloud services, making cloud service identification and discovery a challenging task. This is because of modified help depictions, non-normalized naming shows, heterogeneity in kind, and

B. B. Ahamed (✉)
Department of Information Technology, University of Technology and Applied Sciences, Al Musanna, Sultanate of Oman
e-mail: bazeer@act.edu.om

M. Krishnamoorthy
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: murugan.k@vit.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_1

highlights of cloud administrations. Consequently, choosing a becoming cloud administration as per customer necessities is an overwhelming assignment, in particular for purposes that utilize a piece of more than a few cloud administrations. This part clarifies the proposed semantic-based cloud framework administration revelation and choice framework that characterizes practical and non-useful ideas, properties, and relations of foundation administrations. Here, it is verified how the framework empowers one to find appropriate administrations ideally as referred to with the aid of purchasers.

Due to ever-altering cloud computing environment, applicable carrier composition is needed. With increase in the number of service providers the requirements of service consumer are very demanding time-to-time. Thus, it is difficult to compose offerings based on consumer's preferences. As a result service composition has become a major issue in cloud computing environment. Henceforth, a savvy cloud revelation and shape instrument turns out to be a lot of vital for an appropriate help with greater exactness and least time that fulfills purchaser express measures. As of late, committed programming professionals have been accustomed to get information into allotted computing stipulations that could supply increased adaptability, versatility just as self-sufficiency concerning asset the executives, administration provisioning while at the equal time actualizing massive scope applications. Similarly, for normalization in depiction of cloud administrations, it is needed to make use of a semantic records portrayal methodology viewed metaphysics that focuses to a view of the space of interest.

To design a novel Multi-agent-based framework that implements cloud ontology for cloud service discovery and selection in multi-cloud environment. This frameworkexploits cloud service description annotated with cloud ontology, concepts, propertiesto retrieve semantically related service each of which are ordered by their semantic relevance to the user request. The utilization of keen specialists in this device partitions the excellent burden of the cloud administration revelation and desire into a number of free undertakings. Since the proposed device relies upon disseminated design, the overall performance of the procedure of service discovery can be more desirable by deploying unique software program agents, which can decompose the discovery procedure into a variety of impartial subtasks that can be handled effectively by the software program agents.

1.2 Literature Review

Tim Berners-Lee changed innovative method of accessing and gathering information after the invention of World Wide Web (WWW) in the year 1989. Continuously growing repository of WWW makes it difficult for its user to search for specific information across it. In order to overcome this problem, Semantic Web [1] was introduced afterwards. The idea here was to associate the meaning of the data published on WWW to enable the system to process human understandable information. Hence, Semantic Web can be viewed as an integration of machine understandable

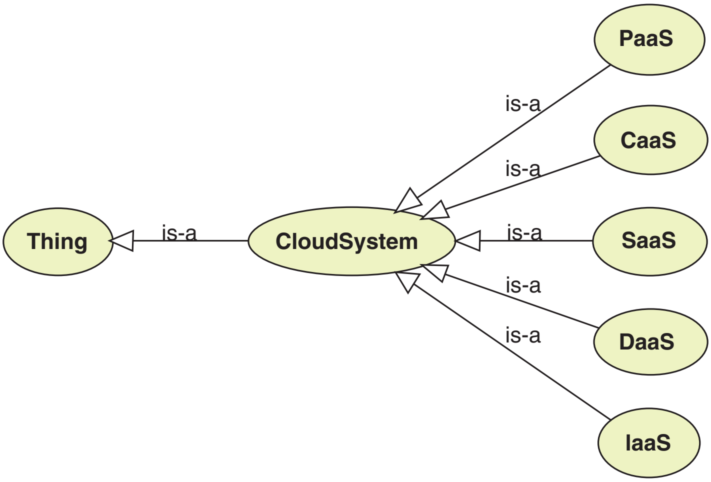


Fig. 1.1 A simple cloud ontology

semantics along with conventional web content. Ontology in Semantic Web represents the technology used for knowledge representation, which formalizes the meaning, thereby facilitating search for contents and improved crawling. An example of cloud ontology that indicates the hierarchical relations of cloud concepts is proven in Fig. 1.1. For instance, the conception of "Cloud System" has five exceptional child nodes (IaaS, PaaS, SaaS, CaaS, and DaaS).

Web service can be perceived to be a public interface for a particular utility that can be invoked in order for performing a commercial enterprise function or a crew of functions. Thus, a web service can be virtually defined to be a self-contained as well as self-describing modular application, which should be published, discovered, and invoked in the World Wide Web. The interactions among the web offerings typically take place in the form of SOAP calls that comprise the content material of XML data. The description of the interfaces with the web services can be articulated using WSDL (Web Service Description Language). The UDDI (Universal Description, Discovery, and Integration) general defines a protocol that can be useful for discovery of services comprising exclusive internet service description. The functional properties relating to a web service disseminate the necessary details pertaining to its functionality that comprises the inputs as well as the outputs of the desired web service along with the pre-condition as well as post-condition related to its functionality.

This information shall be made available to the user through the respective web service provider as and when the selection of the requested service is allowed for the user [2]. "As described by the Software Engineering Institute (SEI) QoS," The probability that a system will deliver particular levels of measurable computational and communication properties such as availability, bandwidth, latency etc. The QoS properties can be used in order to evaluate the degree of conformance of the desired service to a specific quality requirement. Such properties are split into two categories, namely technical and managerial. The technical properties necessarily elaborate the properties that are associated with the operation of the service incorporating availability, security, and reliability. The managerial properties are related to management of the service integrating contract, cost, payment, as well as ownership [3, 4].

A virtual entity referred to as an agent should operate on behalf of another entity in order to perform a task for the attainment of a defined objective. Such agent systems are necessarily self-contained software entities that integrate domain knowledge and are capable of displaying actions with a certain degree of independence to generate action needed to achieve a diversified portfolio. Administrator systems are typically designed to function in an environment that could be constantly evolving. The following are the distinctive characteristics pertaining to such operatives [5, 6]. Zheng et al. [7], that is based on the QoS information along with user preferences, have addressed a three-layer matching method for semantic web service discovery. Here, the client is allowed to set their personal weights for recognition of their non-public choices. A framework for execution of semantic net services proposed in [8] makes use of a context aware broker agent. Here, software agent makes use of the placing information all collectively for distinguishing proof of web administrations regarding instances proposed for improvement of viability and accomplishing scalability while imparting kinds of assistance. An approach to web service selection based on agents has proposed the uses of WSDL information supplied by service provider [9] here use the data mining tool. Weka and think about customer analysis and QoS assessments for web organization revelation. In any case, the QoS regards right here are not categorical and matter upon surmise exorbitantly. Consequently, customers that are sophisticated sufficient can also suppose that it is tough to decide the unique QoS esteems relating to any ideal web administration. Vedivel et al. [10] address a methodology for conjuring and getting to the web advantages that are superior semantically identifying with the subject of portable Web based totally business. This approach necessarily allows the wireless customers to make use of internet services without an online carrier requestor and thereby the usage of mobile and computerized software agents. Gu et al. [11] propose an extended service discovery approach that involves the conventional entities such as service consumer, service provider, and UDDI. In addition to it, the authors introduce a new component called Certifier that holds the responsibility of verification of QoS pertaining to a desired service prior to the process of registration. Here, the consumers are allowed for verification of QoS with Certifier while requesting for the web service. However, this model does not implement user feedback in the process of service discovery even though it integrates QoS into UDDI. A framework for discovery of automatic web services based on software analyzing agents is presented [12].

The authors Alirezaei et al. [4] introduce a new component called Certifier that holds the responsibility of verification of QoS pertaining to a desired service prior to the process of registration. However, this model does not implement user feedback in the process of service discovery even though it integrates QoS into UDDI. The author fused here an autonomic director to control the situation of the internet administrations there with the aid of using five programming professionals, to be specific, an arranging specialist, an execution specialist, apiece specialist, a revelation specialist, and a staring at specialist. Nevertheless, authors do not detail the method of implementation of autonomous manager, and thus the practicality of this model remains unassumed. The multi-agent system discussed in [13] is claimed to be capable enough for efficient web service discovery, Web service discovery QoS registration thereby exhibiting the functionality such as QoS registration, verification, certification, and confirmation with the implementation of multi-agents such as query agent, certification agent, and response agent.

A distributed approach to web service discovery has been proposed in [14] that is supposed to be reliable, flexible, and scalable. This approach integrates intelligent search with centralized shared space among subset of spaces. However, this method ignores automation of search process and thus cannot be regarded as an efficient one. M. Hogan et al. [15] have presented a method that integrates information mining approach along with service ranking strategies into SOA plan for administration revelation. A semantic web administration disclosure approach that makes use of SPARQL to communicate the realistic ascribes relating to web administrations like pre-conditions and post-conditions has been delivered in [16]. However, this method does not take into account the nonfunctional attributes referring to web offerings and, in particular, QoS and performance. A reputation-based framework for semantic web provider discovery is presented in [17] that exploits a recognition administration system in order for series of service scores on types of context. A structure that receives patron inclinations as a scanning instrument for administration determination is tended to in [18], which positions the administrations as for their prerequisites. A two phase semantic based web service discovery technique has been discussed approach has been talked about, and to present a smart cloud intermediary for disentangling the cloud administration choice cycle that can discover services with the implementation of a two-stage matching procedure, i.e., operational matchmaking and operation-composition matchmaking, has been discussed. Nagarajan & Ramkumar [19, 20] present an artificial intelligence based semantic web service discovery model in which a two-level filtration is conducted by the service requestor during service discovery. The administrations process from the perspective of two engineering requirements, specifically Service-situated Architecture (SOA) and Cloud figuring. The cutting-edge examination endeavors acted in the intervals of administration disclosure, advent and provisioning of administrations in accordance to the SLA has been commonly checked on from the perspective of SOA and Cloud. However, the overall performance of this model in terms of time as properly as scalability along with its economic effectivity has not been discussed.

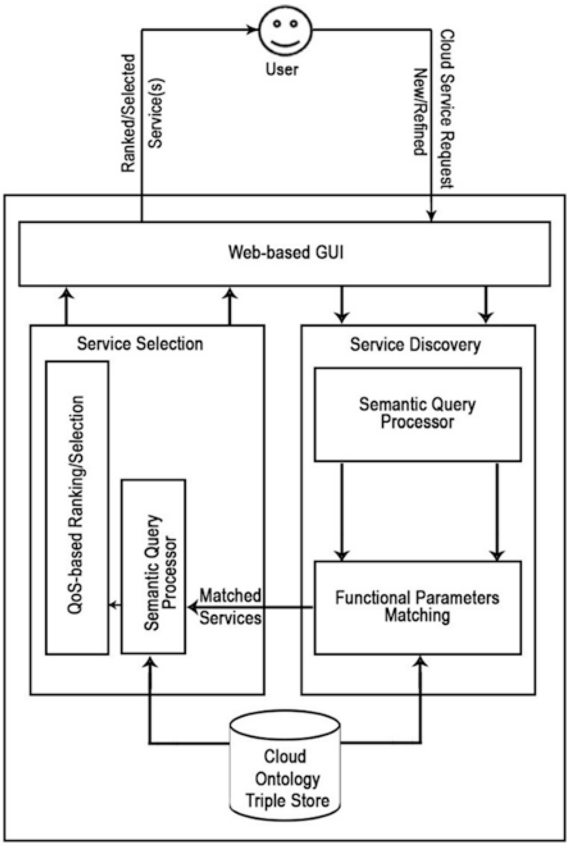
The framework type of administrations, for example, registering and capability, is considered for this proposition. Concerning the purchaser determination, the proposed supplier builds the cloud philosophy to tackle the accessible administrations from the help storehouse. With the information of ontological portrayal, the proposed consultant finds the foundation administrations from the available seller. The appropriate administrations are addressed utilizing semantic organization, which empowers the client to assume about the available administrations in accordance to their posted necessities. At last, the service provider suggests the administrations with its more highlights to the cloud clients. Further, it handles the virus begin trouble through embracing the framework factorization guideline and predicts better QoS esteems for currently confirmed up administrations. To approve our methodology, we have directed exploratory chips away at benchmark datasets, and the result suggests that the proposed strategy outflanks preferred outcomes over the model-based approaches.

1.3 Proposed Methodology

The proposed model as verified in Fig. 1.2 includes three principle levels as nitty-gritty underneath.

1. Client Request: A customer first asks for cloud administration revelation through a Web-based graphical UI (GUI). Client wishes to choose esteems for excep-

Fig. 1.2 Proposed cloud service discovery and selection model



tional utilitarian and nonpractical boundaries, for example, RAM, CPU, ECU, plate space, switch speed, accessibility, ranking and cost, and so on.
2. Administration Discovery: Then the client's solicitation is handled via a semantic inquiry processor, which makes use of the Simple Protocol and RDF Query Language (SPARQL). The estimations of the sensible boundaries are coordinated with the characteristics that are on hand in the cloud metaphysics.
3. Administration Selection: Next, the practically coordinated characteristics are long past via another semantic question processor using the SPARQL language to coordinate the preset nonfunctional boundaries with the qualities current in the philosophy. After the coordinating is done, the located administration is brought to the client, again through the Web-based GUI, which suggests the success of the entire cycle.

The work procedure for the assistance disclosure and choice cycle contains the accompanying advances.

- User sends needs via the digital GUI.
- The solicitation is handled via the semantic inquiry processor; the semantic question processor assessments for coordinating of the utilitarian boundaries.
- The coordinated administrations are passed to the semantic inquiry processor for coordinating of nonpractical boundaries.
- The last placed administration is shipped off the customer via the online GUI.

Calculation 1 delineates the semantic matchmaking measure. This calculation begins by way of emphasizing through all administrations commented on via belongings from client's solicitation (recovered through the potential get all cloud administrations by semantic comment). The semantic rating of each and every one of the recovered cloud administrations is decided with the assistance of semantic likeness work. The semantic closeness work computes the semantic magnitude between two unique properties involving the Tversky likeness model. Consequently, for the two seemed at here assets, the capability get cloud thoughts is utilized to get their cloud ideas, and the capability get cloud homes restores the arrangement of their object properties and data type residences from the cloud cosmology. At that point, the capability Tversky closeness Algorithm 1 is conjured to confirm the stage of pertinence with the aid of uprightness of the calculation of the shared characteristics and contrasts pertaining to the notion about sets. Then, the ultimate rating of the potential semantic likeness is gotten by using registering the normal among the ideas, object properties, and records type properties coordinating scores.

Algorithm 1: Semantic Matchmaking of Cloud Services

INPUT1: A set of cloud infrastructure resources CIR depicting needed functionalities of an assistance

OUTPUT1: set of semantically coordinated with the cloud administrations S’(1) ⊆S

S=get all cloud contribution via semantic annotation (CIRq)

if each set S∈S do

Set score = =0

// reoccurrence all IaaS resources used to interpret the functionalities of cloud service 's'

if every r1’ ∈ CIRs do

if every r1 ∈ CIRq do

Set Score = rating + semantic similitude (r & & 'r’)

end if;

end if;

if score & gt;= 0 then;

S. semantic score= = log (1+score)

S’(1) = S’ U {s}

end if

end for

return S’(1)

end function

Algorithm 2: Tversky Similarity () Function

INPUT: Two list of cloud features F1 and F2

OUTPUT: A similarity between 0 and 1 using the function Tversky Similarity (r; r')

C=Common Features (F1 and F2)

UF1=Unique Features (F1 and F2)

UF2=Unique Features (F2 and F1)

$sim = \frac{|c|}{|c|+|U_{F1}|+|UF2|}$

return sim

end function

Here, the Tversky's model in Algorithm 2 is utilized with the coordinating semantic Cloud administrations, as it has been considered as possibly the most incredible likeness models to date. The model considers the highlights that are normal to two ideas and moreover the isolating highlights specific to each. Even more explicitly, the likeness of a concept c1 to an idea c2 is a component of the highlights regular to c1 and c2, these in c1 yet not in c2 and these in c2 yet not in c1.

Leave F1 separately through the rundown of highlights of IaaS = {Operating System, Virtualization, Servers, Storage, Data Center, Networking, Firewall, and Security}.

Leave F2 separately through the rundown of highlights of NaaS = {Networking, Firewall, Security, Virtual Private Network, and Mobile Network Virtualization}.

Leave C separately through the normal Features in the center of them for example {Network, Firewall, Security}.

UF1 = {Operating System, Virtualization, Servers, Storage, Data Center}.

UF2 = {Virtual Private Network, Mobile Network Virtualization}.

Presently the closeness measure among IaaS and NaaS is registered as follows:

$$sim = \frac{|c|}{|c| + |U_{F1}| + |U_{F2}|} = \frac{3}{3+5+2} = 0.3 \tag{1.1}$$

Additionally, for the Object property like OS (Windows 10, Windows 8.1) in the proposed philosophy, Windows 10 and Windows 8.1 are similar by ethicalness of their regular properties like Windows Phone 8 help, secure VPN uphold, Software Compatibility, and divergent by means of prudence of their disparities, to be particular, Migration cost, Prior OS joining, Branch Cache Support, User Account manipulate, and so forth. The discovey system returns the result which is shown in the Table 1.1.

Table 1.1 Evaluation of user request with discovery result

Request constraints	User query	Detection result
Service type	Iaas	Naas
Operating system	Windows 8.1	Windows 10
Expense	0.01	0.03
Memory(GB)	4	2
Bandwidth(GB/s)	6	4
Central processing unit	2	4
Determined storage (hard disk) (GB)	100	200

1.4 Implementation

To actualize the proposed structure, the Hypertext Preprocessor (PHP) stage is utilized. The cosmology is made utilizing the Protege metaphysics supervisor, which depends on Java stage. ARC2 structure is utilized for Semantic application uphold in PHP.

1.4.1 Building the Cloud Ontology

Here, the Multi-Cloud administration cosmology relies upon the area model of the Infrastructure as a Service (IaaS) layer. The Cloud basis administrations are instantly portrayed with the aid of this metaphysics, and the framework administration revelation determined with the aid of its utilitarian and non-useful QoS boundaries is encouraged through mappings of specialized portrayals of the cloud expert co-ops. The proposed cosmology fills in as a semantic primarily based vault throughout the enrollment, disclosure, and preference cycle. The necessary administrations region records have been gathered from numerous assets: cloud scientific categorization, cloud metaphysics, and enterprise organized norms (The National Institute of Standards and Technology (NIST) [21]. Likeness questioning is carried out with the aid of uprightness of suggestions with the cloud philosophy. The said philosophy is expressly characterized in OWL and can be seen at: http://www.mrparhi.in/Iaas/major_project.owl. This metaphysics is planned utilizing the Protégé v4.3 Ontology Editor, an open-source instrument, which has been created by using the Stanford University. The proposed philosophy carries the principal classification named as Cloud Service Discovery with subclasses as Cloud Service Attributes, Cloud Service Providers, and QoS. The Cloud Service Attributes class includes all the utilitarian credits of the Cloud professional businesses while the QOS subclass incorporates all the nonpractical boundaries and the cloud expert organization's category includes all the Cloud expert organizations (Fig. 1.3).

In this stage, the purchaser needs to pick out the applicable characteristics as indicated through his/her necessities from a bunch of utilitarian boundaries like OS

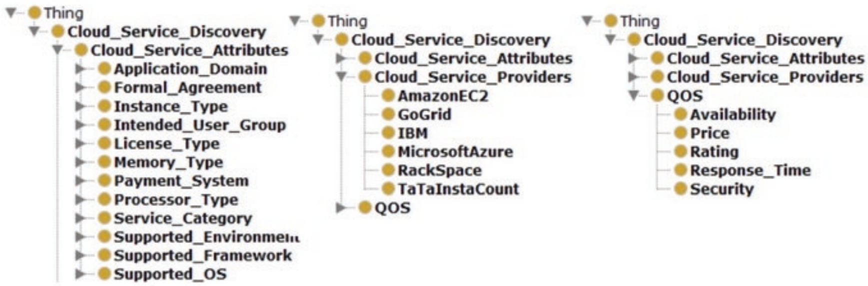


Fig. 1.3 Classes and subclasses used in the proposed cloud ontology

```
String queryString = "PREFIX table:<http://www.mrparhi.in/     Cloud_Ont
   #>"
     + "SELECT ?service ?url ?service_name ?rating ?avail ?price  ?ram
        ?cpu  ?os  ?disk  ?band WHERE{"
     + "?employee table:has_service_provider_name ?service;"
     + "table:has_url ?url;" + "table:has_service_name ?service_name;"
     + "table:has_rating  ?rating;" + "table:has_availability  ?avail;"
        + "table:has_price ?price;" + "table:has_ram  ?ram;"
     + "table:has_cpu  ?cpu;" + "table:has_os  ?os;" + "
        table:has_disk_space  ?disk;" + "table:has_bandwidth  ?band;" +
         "table:has_type  ?type.";
```

Fig. 1.4 SPARQL query for Discovery phase

Type, RAM restrict (GB), variety of CPU Cores, Bandwidth (Gbps), Disk area (GB), and so on for the most part, the cloud specialist agencies have diverse utilitarian boundaries, which may also not be on hand with all the suppliers. Hence, some everyday boundaries are viewed here, for which the proposed Ontology is known as Unified Ontology. At that point, the purchaser needs to pick the nonpractical/QoS boundaries as per their need after which the purchaser desires to tap the Discover and Select capture to execute the question. The Semantic Query Processor is utilized to deal with the inquiry produced by way of the customer utilizing the SPARQL query language. SPARQL Query for revelation stage is given beneath (Fig. 1.4).

1.5 Experimental Results and Discussion

To Examine the result,the three performance measure experiment was conducted to evaluate the search efficiency of the proposed approach taking the same number of iterations. Three execution gauges in most cases utilized in data restoration [22, 23], for example, Accuracy, Recall, and F-Measure, are utilized to gauge the hunt productiveness throughout revelation and desire of cloud administrations. As there is no popular informational index available for semantic cloud administration revelation, a semantic cloud administration Repository used to be assembled, for example, the Cloud Ontology: Tripple Store making use of Protege v4.3 Ontology Editor that includes 51 IaaS Individuals (VM examples) with 16 boundaries as data properties gathered from the pinnacle IaaS cloud professional organization's entrances (for example, Amazon, Microsoft-Azure, GoGrid, IBM, Rackspace, and so on). During the disclosure cycle, the Precision, Recall, and F-measure esteems are registered through crossing a given rundown of cloud administrations, as per a query and proposed semantic matchmaking calculation. For a given inquiry Q, the Precision P is the extent of the pertinent cloud administrations recovered to all the recovered cloud benefits and is numerically communicated as

$$P = \frac{|\text{relevent services}| \cap |\text{retrieved services}|}{|\text{retrieved services}|} \tag{1.2}$$

In addition, the Recall R is the extent of important cloud administrations, which have been recovered to all the significant administrations and is numerically communicated as

$$R = \frac{|\text{relevent services}| \cap |\text{retrieved services}|}{|\text{relevent services}|} \tag{1.3}$$

At last, the F-Measure score have figured. The F-measure is the consonant amount of Precision and Recall, which gives the exactness of the methodology and is depicted as

$$F = 2 \times \frac{|P \times R|}{|P + R|} \tag{1.4}$$

The average for Recall, Precision, and F-measure have figured for both social and proposed philosophy based methodology as demonstrated in Table 1.2.

Table 1.2 Calculating the search effectiveness: ontology vs. relational method

Type of matching	Average precision (%)	Average recall (%)	Average F-measure (%)
Ontology method	56.01	17.95	27.08
Relational database method	52.48	13.27	20.53

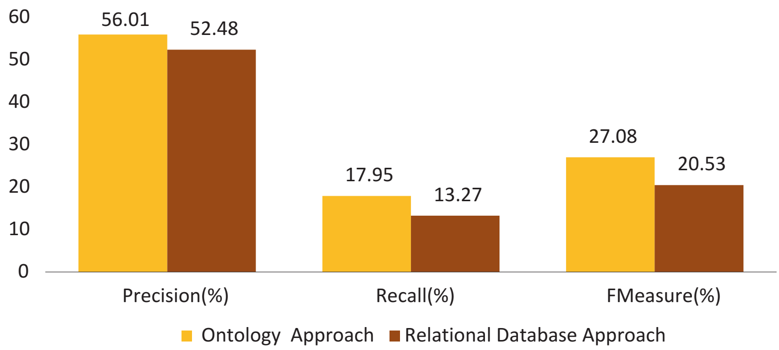


Fig. 1.5 Comparison of search effectiveness between ontology approach and relational approach

The evaluation end result as tested in Fig. 1.5 shows that search productivity of cosmology method is dazzling better as a way as Precision, Recall, and F-Measure than social methodology due to the fact of the proposed calculation and semantic principles. After investigation, there are two explanations at the back of low estimations of Precision, Recall, and F-measure.

- The philosophy in semantic cloud administration store is not always sufficiently massive.
- Inference guidelines are sufficiently not.

Therefore, this experiment is further extended by adding few pseudo cloud services to the cloud ontology for analysing the above two issues. The motive for making these pseudo administrations was once to display positive attributes to test the unbending nature of the proposed matchmaking calculation. At that point the quantity of IaaS clients are multiplied with a variety of data esteems and article residences in the proposed cloud cosmology. The Precision, Recall, and F-measure esteems appear in Table 3; what's more, the comparing plan is shown in Fig. 1.6.

Examination of the effects demonstrates that there is staggering enchantment in the presentation of the framework as a long way as Recall, Precision, and F-Measure esteems with the growth in the extent of classes, people, properties of cloud metaphysics, and the extent of surmising rules. This empowers the framework to find the hugest administrations as noted via the cloud administration customers.

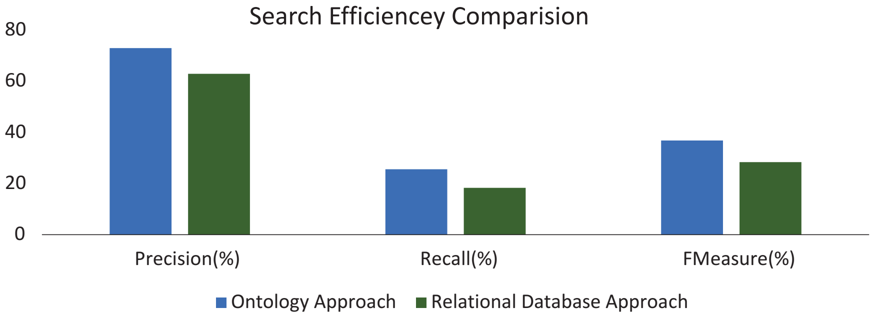


Fig. 1.6 Comparison of search efficiency between ontology approach and relational approach after ontology extension

Table 1.3 Measuring search efficiency after ontology extension: ontology's relational approach

Type of similar approach	Average precision (%)	Average recall (%)	Average F-measure (%)
Ontology approach	73	25.61	36.89
Relational database approach	62.9	18.37	28.37

1.6 Conclusion

The proposed ontology for classifying and representing the configuration data related to Cloud IaaS services is complete for the reason that it can seize each static configuration as well as dynamic QoS configuration. Further, the notion of ontology is integrated which enhances the efficiency and correctness of the proposed model. The model permits users to effectively search and choose the satisfactory feasible Cloud IaaS service provider with the most feasible accuracy level. A collection of experiments is conducted in this work and all the experiments show that the proposed strategy performs much better in terms of response time and accuracy measures such as Precision, Recall, and F-Measure.

References

1. Ali, A., Shamsuddin, S. M., & Eassa, F. E. (2014). Ontology-based cloud services representation. *Research Journal of Applied Sciences, Engineering and Technology, 8*(1), 83–94.
2. Ahamed, B. B., & Ramkumar, T. (2016). An intelligent web search framework for performing efficient retrieval of data. *Computers & Electrical Engineering, 56*, 289–299.

3. Ahamed, B., Najimaldeen, R. M., & Duraisamy, Y. (2020, April). Enhancement framework of semantic query expansion using mapped ontology. In 2020 international conference on computer science and software engineering (CSASE) (pp. 56–60). IEEE.
4. Alirezaei, E., & Parsa, S. (2020). Adaptable cross-organizational unstructured business processes via dynamic rule-based semantic network. *Information Systems Frontiers, 22*(3), 771–787.
5. Arul, U., & Prakash, S. (2020). Toward automatic web service composition based on multilevel workflow orchestration and semantic web service discovery. *International Journal of Business Information Systems, 34*(1), 128–156.
6. Badr Y., Abraham A., Biennier F. and Grosan C. (2008). Enhancing web service selection by user preferences of non-functional features. Proceeding of the 4th international conference on next generation web services practices (pp. 60–65). Seoul.
7. Hoefer, C. N., & Karagiannis, G. (2010). Taxonomy of cloud computing services. IEEE GLOBECOM workshop on enabling the future service oriented internet (pp. 1345–1350).
8. Chainbi W., Mezni H., & Ghedira K. (2010). PECoDiM: An agent based framework for autonomic web services. Proceedings of the IEEE 6th world congress on services, Miami, Florida (pp. 543–550).
9. Chang, Y. S., & Cheng, H. T. (2012). A scientific data extraction architecture using classified metadata. *Journal of Supercomputing, 60*(3), 338–359.
10. Dillon, T., Chen, Wu., & Chang, E. (2010). Cloud computing: Issues and challenges. Proceedings of the 24th IEEE international conference on advanced information networking and applications (AINA'10), 20–23 April (pp. 27–33).
11. Gu, T., Pung, H. K., & Yao, J. K. (2005). Towards a flexible service discovery. *Journal of Network and Computer Applications, 28*(3), 233–248.
12. Guns, R., Lioma, C., & Larsen, B. (2012). The tipping point: F-score as a function of the number of retrieved items. *Journal of Information Processing & Management, 48*(6), 1171–1180.
13. Hafsi, A., Gamha, Y., Njima, C. B., & Romdhane, L. B. (2020, June). BIG-SWSDM: BIpartite graph based social web service discovery model. In *International conference on business information systems* (pp. 307–318). Springer.
14. Lopes A. L., & Botelho L. M. (2007). Executing semantic web services with a context-aware service execution agent. Proceedings of the sixth international joint conference on autonomous agents and multiagent systems (SOCASE) (Vol. 4504, pp. 1–15). Springer.
15. Hogan, M., Liu, F., Sokol, A., & Tong, J. (2011). NIST cloud computing standards roadmap. [online]. Retrieved October 4, 2016, from https://www.nist.gov/sites/default/files/documents/itl/cloud/NIST_SP-500-291_Jul5A.pdf.
16. Majithia S., Ali A. S., Rana O. F., & Walker D. W. (2004). Reputation-based semantic service discovery. Proceedings of the 13th IEEE international workshops on enabling technologies: Infrastructure for collaborative enterprises (pp. 297–302).
17. Nam, S., & Kang, Y. (2008). XML schema design for web service quality management. Proceedings of the 2nd international conference on future generation communication and networking, Hainan Island (Vol. 2, pp. 99–102).
18. Paniagua, C., Eliasson, J., & Delsing, J. (2019). Efficient device-to-device service invocation using arrowhead orchestration. *IEEE Internet of Things Journal, 7*(1), 429–439.
19. Nagarajan, N., & Thirunavukarasu, R. (2020). Service-oriented broker for effective provisioning of cloud services–a survey. *International Journal of Computing and Digital Systems, 9*(5), 863–879.
20. Nagarajan, R., Thirunavukarasu, R., & Shanmugam, S. (2018). A cloud broker framework for infrastructure service discovery using semantic network. *International Journal of Intelligent Engineering and Systems, 11*(3), 11–19.
21. Zhou, A., Ren, K., Li, X., Zhang, W., Ren, X., & Deng, K. (2020). Semantic-based discovery method for high-performance computing resources in cyber-physical systems. *Microprocessors and Microsystems, 80*, 103328.

22. Zhou, J., & Niemela, E. (2006). Toward semantic qos aware web services: Issues, related studies and experience. Proceedings of the 2006 IEEE/WIC/ACM on web intelligence (pp. 553–557).
23. Zhou, J., & Niemela, E. et al. (2007) An integrated QoS-aware service development and management framework. Proceeding of the working IEEE/IFIP conference on software architecture (WICSA'07).

Chapter 2
Hybrid Machine Learning Models for Distributed Biological Data in Multi-Cloud Environment

K. Thenmozhi, M. Pyingkodi, and K. Ramesh

2.1 Introduction

Big data is an emergent field which increases more number of data in the fields like marketing, medical, biological research, transaction of data, and so on. Due to growing size of data, data retrieval is more complex. Big data is classified into three V's, that is, Volume, Velocity, and Variety of data [1]. Big data, which is a huge volume of data, is not only collected from computers but also from mobile phones, sensors in various filed, social media posts, and many other resources. Data retrieval, data analysis, quality and quantity measures of algorithm and data, and outlier detection are considered various issues in Big data [2].

Biological data is a collection of life science information, computational study, information of living organism, and high quantity of research knowledge. The progress of biological data information's collected from DNA, RNA, protein discovered [3, 4]. The types of biological data are incorporated from genomics, proteomics, microarray, metabolomics, gene expression, and ontology, and so on. The biological data is distinguished in different data format like image, sequence, structure, patterns, graph, text, geometric, and expression [5, 6].

K. Thenmozhi (✉)
Department of Computer Science, Kristu Jayanti College, Bangalore, India

M. Pyingkodi
Department of Computer Applications, Kongu Engineering College, Erode, India

K. Ramesh
Department of Computer Applications, Karpagam Academy of Higher Education, Coimbatore, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_2

The cell is the basic structure of every living organism. The nucleus is the heart of the cell with chromosomes which have a part called DNA. The four bases of DNA are Adenine (A), Cytosine(C), Guanine (G), and Thymine (T). DNA is transcribed into RNA which has the base pair of Adenine (A), Cytosine(C), Guanine (G), and Uracil (U) [7]. The base pair of RNA is similar to DNA except for Thymine. RNA use Uracil instead of Thymine. RNA is translated to protein. Proteins are formed by linking different amino acid or peptide bonds [8]. A protein is normally denoted as a sequence or string on an alphabet of 20 characters, except B, J, O, U, X, and Z.

Cloud computing is the main part of the research in bioinformatics for huge volume of biological data [9]. Distributed cloud computing is one of the main roles in cloud computing that simplifies the cloud location, progress, distribution of data, and application from various sites to achieve the necessities, hence improving the performance and reducing the idleness.

Machine learning denotes to design and assess the algorithms to enable the data mining models from raw data. Generally, machine learning facilitates the two learning mechanization, that is, supervised learning and unsupervised learning [10]. Supervised learning represents the classification and prediction of the members with known features based on class label of data. Unsupervised learning, otherwise called as clustering and outliers, collects similar data into one group and dissimilar data into another. Both learning mechanisms work well in biological research for biological data. The combination of machine learning and deep learning is quite complex for biological data. Machine learning hybrid with deep learning and cloud computing enhances the performance of the algorithm.

Distributed clustering is used to solve computational issues in distributed data. Generally, the data is classified into two forms: homogeneous and heterogeneous. Homogeneous data has similar dataset attributes, and heterogeneous has different dataset attributes. In Fig. 2.1, the distributed clustering is done in two levels such as local and global [11, 12].

2.1.1 Chapter Sections Overview

Chapter sections are organized as follows: Sect. 2.1, describes the introduction; Sect. 2.2 presents a detailed survey of previous studies, Sect. 2.3 explains about the hybrid models; Sect. 2.4, presents the results and discussion; and Sect. 2.5 presents the conclusion.

2.2 Literature Review

Bioinformatics is an emerging research area for storing and accessing a huge volume of data. Data access is a difficult task in the research field. The structure and function of protein based on the statistical metric based feature selection techniques,

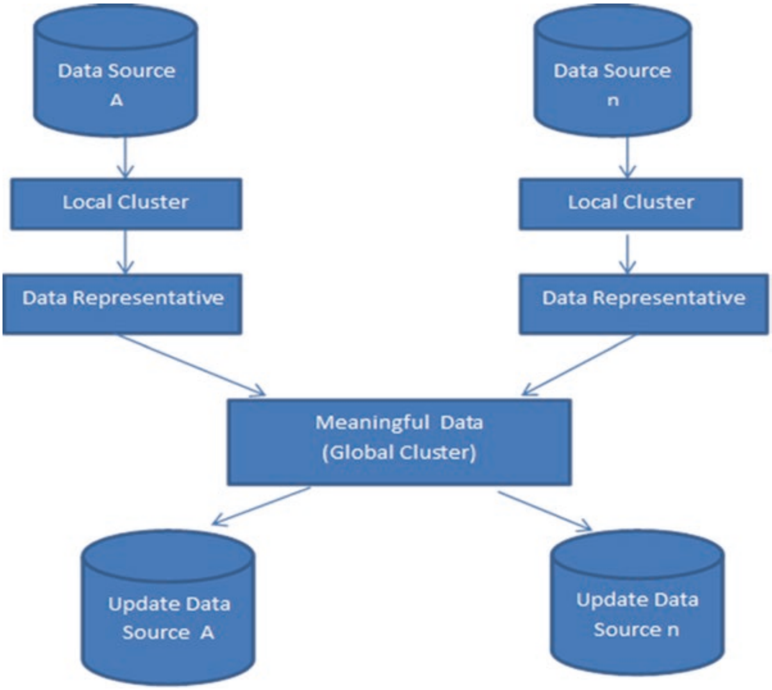


Fig. 2.1 Architecture of distributed clustering

which reduced the feature vector size for evaluate the growing biological data [13]. The neural network classifiers compared with other classifiers for improving the evaluation. The accuracy of classification is to exactly identify the changes of amino acid sequence. This feature selection proves a significant upgrade in performance in terms of accuracy, sensitivity, and F-measure. This selection technique fails to manage the time complexity for accessing the data.

The distribution-based spectral clustering and cuckoo search used for cancer identification with protein sequence data reduces time complexity. Invariant sequence identified based on the similarity index, which is identified by Jaccard similarity index. Fuzzy logic used to detect the membership value of protein sequence. Based on the similarity and membership value, the sequence is detected whether cancerous or non-cancerous. This distribution-based spectral clustering improves the accuracy and reduces the time but fails to detect the features-based detection [14, 15].

TRIBE-MCL is used for the family of protein to detect the information of sequence similarity. Protein family detection is one of the main goals of functional and structural genomics. Construct a protein–protein similarity graph for proteins.

Then, generate a weighted transition matrix for the constructed similarity graph by BLAST E-Values and finally transform, weight into transition probability for constructing a Markov matrix. This task is probably expensive to achieve a goal in a short period [16].

The deep learning algorithm exactly identifies the breast cancer using mammography image. Digital Database for Screening Mammography (CBIS-DDSM) test improves the sensitivity, specificity and reduces the false-positive and false-negative rates [17]. Deep learning method is highly suitable for heterogeneous mammography image, but it takes much time to produce the result of algorithm. Random forest and distributed techniques are rarely used in biological environment [18, 19].

2.3 Hybrid Models of Deep Learning and Machine Learning

The data is distributed among various places and size. If all the data collected into single site, it takes more execution time and memory for process the data. To avoid this contingency, the distributed approach is used to cluster the data locally and form a global data based on data representative. Local cluster is done by Distributed Spectral Clustering (DSC) technique such that construct a diagonal matrix for "n" number of protein data, then find the similarity using Jaccard similarity index, then compute the Laplacian function with the help of Eigen values and Eigen vectors. Then, run the Fuzzy C-Means (FCM) to separate an object. In normal spectral clustering, K-means is used to separate a data instead of FCM. Apply the statistical metric-based feature selection in global data. This selection is done based on the scoring and length of the sequence. In this model, machine learning algorithm of spectral clustering is used to split up the data based on the similarity and the deep learning-based feature selection acts to get final informative sequence. Table 2.1. represents the Pseudo code of Distributed Spectral Clustering with Feature Selection (DSCFS).

Table 2.1 Pseudo code of distributed spectral clustering with feature selection

Step 1: Construct diagonal matrix
Step 2: Build a similarity matrix by Jaccard similarity index
Step 3: Compute Laplacian function by Eigen values and vectors
Step 4: Update Laplacian function
Step 5: Minimize the objective function by fuzzy membership
Step 6: Apply the statistical based feature subset selection based on length and score of the sequence
Step 7: Get the final informative sequence

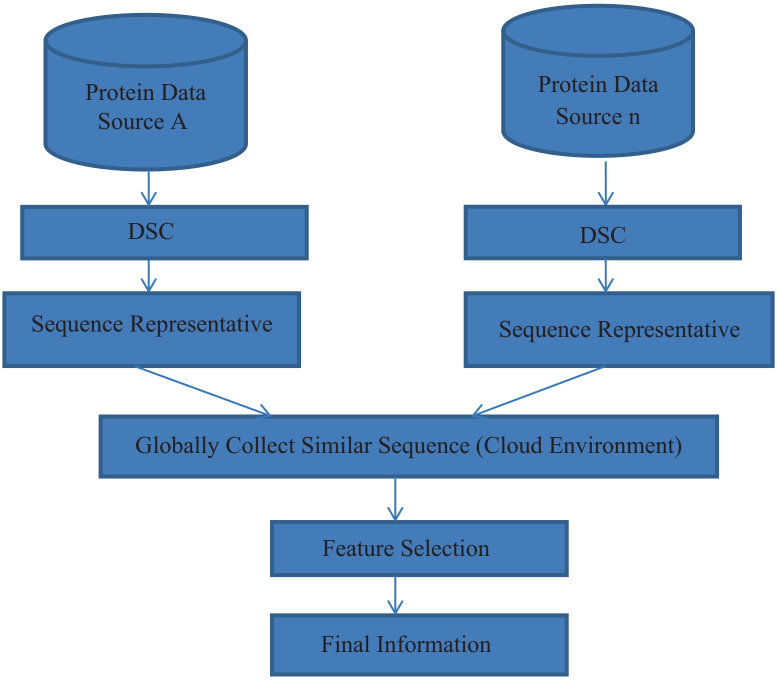


Fig. 2.2 Architecture of distributed spectral clustering with feature selection

Feature selection is done based on the length and score of amino acid. The standard 20 (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y) amino acid is used to stipulate the protein sequence of any length for any gene. Figure 2.2 represents the architecture of Distributed Spectral Clustering with Feature Selection.

2.4 Experimental Results and Discussion

The clustering measures are calculated by the following values: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) [20, 21].

2.4.1 Accuracy

Accuracy is refers to defined as correctly detect the cancerous sequence by the total number of sequence. It is measured in terms of percentage (%) (Table 2.2; Fig. 2.3).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2.1}$$

Table 2.2 Accuracy for TRIBE-MCL and DSCFS

Sequences	TRIBE-MCL	DSCFS
500	74	78
1000	79	82
1500	84	87
2000	88	93

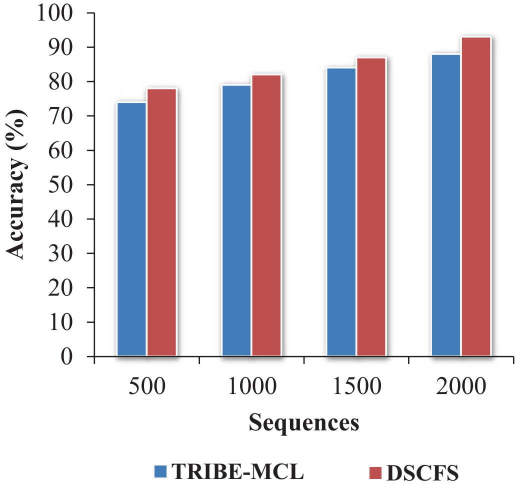


Fig. 2.3 Comparison of TRIBE-MCL and DSCFS in terms of accuracy

2.4.2 Precision/Specificity

Precision is referred to measure the quality of accuracy and it is the ratio of correctly identified sequences and the total number of sequences. It is also measured in terms of percentage (%) (Table 2.3; Fig. 2.4).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.2}$$

2.4.3 Recall/Sensitivity

Recall is referred to measure the quality of accuracy and it is defined as a fraction of correctly identified sequences and the total number of sequences. It is also measured in terms of percentage (%) (Table 2.4; Fig. 2.5).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.3}$$

Table 2.3 Precision for TRIBE-MCL and DSCF

Sequences	TRIBE-MCL	DSCFS
500	75	81
1000	77	84
1500	80	89
2000	83	91

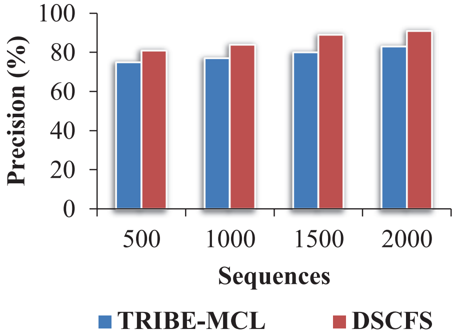


Fig. 2.4 Comparison of TRIBE-MCL and DSCFS in terms of precision

Table 2.4 Recall for TRIBE-MCL and DSCF

Sequences	TRIBE-MCL	DSCFS
500	71	76
1000	75	81
1500	78	85
2000	82	91

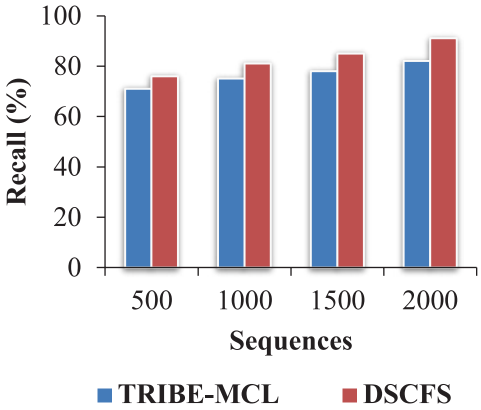


Fig. 2.5 Comparison of TRIBE-MCL and DSCFS in terms of recall

Table 2.5 F-measure for TRIBE-MCL and DSCF

Sequences	TRIBE-MCL	DSCFS
500	73	78
1000	76	82
1500	79	87
2000	82	91

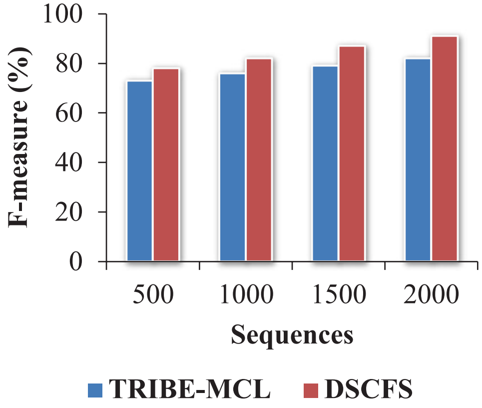


Fig. 2.6 Comparison of TRIBE-MCL and DSCFS in terms of F-measure

2.4.4 *F-Measure*

F-measure is referred to integrate the mean of precision and recall. It is also measured in terms of percentage (%) (Table 2.5; Fig. 2.6).

$$F = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{2.4}$$

2.4.5 *Time*

Time is referred to as starting and ending time of execution for the total number of sequence which is measured in terms of milliseconds (ms) (Table 2.6; Fig. 2.7).

2.4.6 *Motif for Normal Sequence (Fig. 2.8)*

Table 2.6 Times for TRIBE-MCL and DSCF

Sequences	TRIBE-MCL	DSCFS
500	47	24
1000	51	27
1500	54	31
2000	58	33

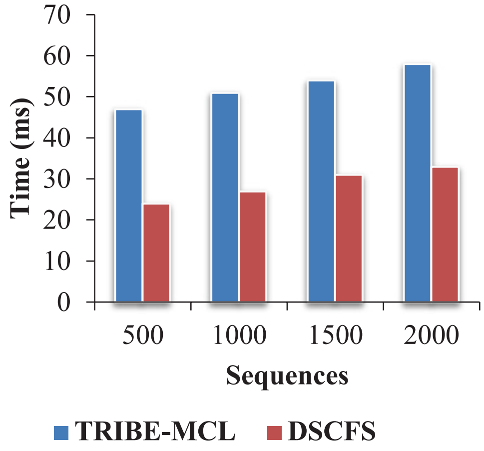


Fig. 2.7 Comparison of TRIBE-MCL and DSCFS in terms of time

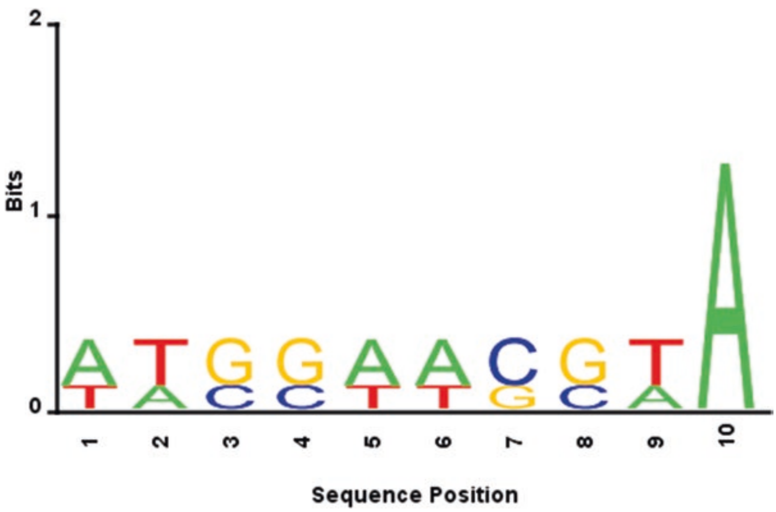


Fig. 2.8 Motif graph for normal sequence

2.4.7 Motif for Mutant Sequence (Fig. 2.9)

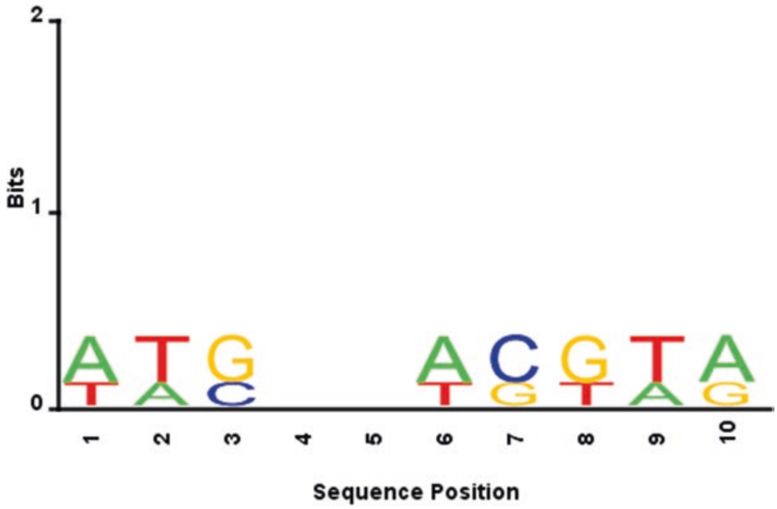


Fig. 2.9 Motif graph for mutant sequence

2.5 Conclusion

The Distributed Spectral Clustering with Feature Selection techniques is done in two models such as local and global models to reduce the time complexity, and feature selection is used to enhance the accuracy, precision, recall, and F-measures. Local model acts as a clustering and global model acts as Cloud, which provide most of the intelligent services like security, performance, productivity, reliability, scalability, speed, and accurate access. This method is mainly applicable for huge volume of distributed data. The results achieved are based on similarity, length, and score of the sequence. This novel technique is compared with TRIBE-MCL to show better performance to get mutant protein sequence. Every measure in this technique shows better performance than literature TRIBE-MCL method.

References

1. Driscoll, A., Daugelaite, J., & Sleator, R. D. (2013). Big data- hadoop and cloud computing in genomics. *Journal of Biomedical Informatics, 46*(5), 774–781. https://doi.org/10.1016/j.jbi.2013.07.001
2. Thenmozhi, K., Visalakshi, N. K., & Shanthi, S. (2017). Optimized data retrieval in big data environment using PPFC approach. *Asian Journal of Research in Social Sciences and Humanities, 7*(3), 683–690. https://doi.org/10.5958/2249-7315.2017.00198.8

3. Pan, T., & Uhlenbeck, O. C. (1993). Circularly permuted DNA, RNA and proteins - A review. *Gene, 125*(2), 111–114. https://doi.org/10.1016/0378-1119(93)90317-v
4. Konathala, G. K., Mandarapu, R., & Godi, S. (2017). Oncogenic mutations of PIK3CA and HRAS in carcinoma of cervix in south Indian women. *Journal of Oncological Sciences, 3*(3), 112–116. https://doi.org/10.1016/j.jons.2017.10.004
5. Diniz, W. J. S., & Canduri, F. (2017). Bioinformatics: An overview and its applications. *Genetics and Molecular Research, 16*(1), 1–21. https://doi.org/10.4238/gmr16019645
6. Zou, D., Ma, L., Yu, J., & Zhang, Z. (2015). Biological databases for human research. *Genomics, Proteomics & Bioinformatics, 13*(1), 55–63. https://doi.org/10.1016/j.gpb.2015.01.006
7. Rodrigues, M. J. F., & Mering, C. V. (2014). HPC-CLUST: Distributed hierarchical clustering for large sets of nucleotide sequences. *Bioinformatics, 30*(2), 287–288. https://doi.org/10.1093/bioinformatics/btt657
8. Abdul, M., Safdar, A., Mubashar, I., & Nabeela, K. (2014). Prediction of human breast and colon cancers from imbalanced data using nearest neighbor and support vector machines. *Computer Methods and Programs in Biomedicine, 113*(3), 792–808. https://doi.org/10.1016/j.cmpb.2014.01.001
9. Shakil, K, & Alam, M. (2018). Cloud computing in bioinformatics and big data analytics: Current status and future research. Advances in Intelligent Systems and Computing (pp. 629–654). https://doi.org/10.1007/978-981-10-6620-7.
10. Xu, C., & Jackson, S. A. (2019). Machine learning and complex biological data. *Genome Biology, 20*, 76. https://doi.org/10.1186/s13059-019-1689-0
11. Januzaj E, Kriegel H-P, & Pfeifle M (2003). Towards effective and efficient distributed clustering. Workshop on clustering large data sets (ICDM2003), Melbourne (Vol. 1, pp. 1–10).
12. Bendechache, M., & Kechadi, T. (2015). Distributed clustering algorithm for spatial data mining. Second IEEE international conference on spatial data mining and geographical knowledge services, China (Vol. 15, pp. 60–65). https://doi.org/10.1109/ICSDM.2015.7298026.
13. Iqbal, M. J., Faye, I., Samir, B. B., & Said, A. M. (2014). Efficient feature selection and classification of protein sequence data in bioinformatics. Hindawi Publishing Corporation, Scientific World Journal (pp. 1–12). https://doi.org/10.1155/2014/173869.
14. Thenmozhi, K., Visalakshi, N. K., & Shanthi, S. (2018). Distribution based fuzzy estimate spectral clustering for Cancer detection with protein sequence and structural motifs. *Asian Pacific Journal of Cancer Prevention, 19*(7), 1935–1940. https://doi.org/10.22034/APJCP.2018.19.7.1935
15. Thenmozhi, K., Visalakshi, N. K., & Shanthi, S. (2018). Distributed ICSA clustering approach for large scale protein sequences and Cancer diagnosis. *Asian Pacific Journal of Cancer Prevention, 19*(11), 3105–3109. https://doi.org/10.31557/APJCP.2018.19.11.3105
16. Enright, A. J. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research, 30*(7), 1575–1584. https://doi.org/10.1093/nar/30.7.1575
17. Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., & Sieh, W. (2019). Deep learning to improve breast Cancer detection on screening. *Scientific Reports, 9*(1), 12495. https://doi.org/10.1038/s41598-019-48995-4
18. Ramesh, K., Vinitha, A., Dhamodharan, M., & Shanmuga, V. M. (2020). An improved random Forest algorithm for effective stock market prediction trending towards machine learning. *International Journal of Grid and Distributed Computing, 13*(1), 873–881.
19. Thenmozhi, K., Pyingkodi, M., & Kumaravel, S. (2018). Tapered assessment on distributed clustering vital in protein sequence environment. *International Journal of Recent Trends in Engineering & Research (IJRTER), 4*(7) https://doi.org/10.23883/IJRTER.2018.4350.D8R7D.
20. Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006). Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. Advances in Artificial Intelligence (pp. 1015–1021). https://doi.org/10.1007/11941439_114.
21. Huda, S., Yearwood, J., Jelinek, H. F., Hassan, M. M., Fortino, G., & Buckland, M. (2016). A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis. *IEEE Access, 4*, 9145–9154. https://doi.org/10.1109/ACCESS.2016.2647238

Chapter 3
Multi-Cloud Path Planning of Unmanned Aerial Vehicles with Multi-Criteria Decision Making: A Literature Review

K. Santhi, B. Valarmathi, and T. Chellatamilan

3.1 Introduction

UAV denotes a pilotless aircraft which functions via a combination of technologies such as artificial intelligence, computer vision, object avoidance technologies, and what not. A UAV operates on ample levels of independency via remote control by a manual operation or a pre-programmed operation on board computers. Having experience significant levels of achievements on autonomous technologies, UAVs take a big chunk on wide versatility of applications, such as national security, talking of emergency situation, humanitarian aid and disaster management, conservation of resources, disease control and prevention, agriculture and forming, weather forecasting, urbanization, retail, manufactural establishment, nourishment of inventories, and upbringing of economies.

3.2 Classification of UAV

There is no one criterion when it arrives to the classification of UAV. They are categorized by aerodynamics, landing, weight, and range as shown in Fig. 3.1.

K. Santhi (✉)
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: santhikrishnan@vit.ac.in

B. Valarmathi · T. Chellatamilan
School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: valarmathi.b@vit.ac.in; chellatamilan.t@vit.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_3

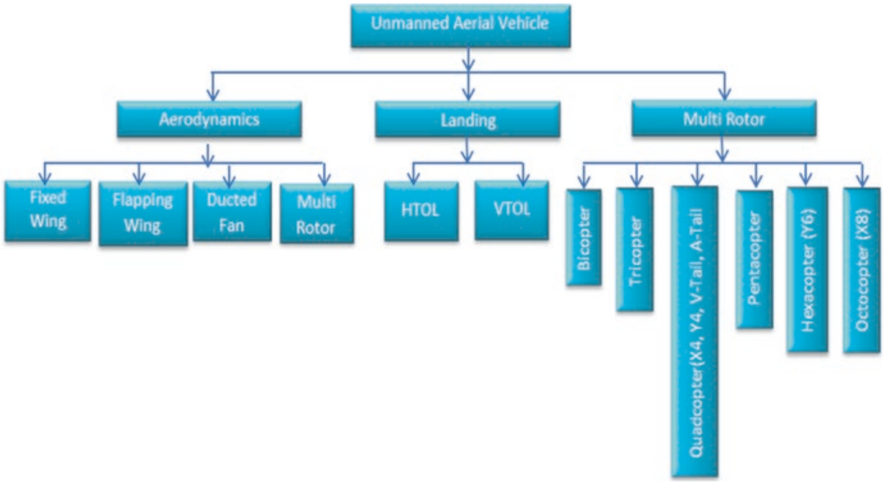


Fig. 3.1 Classification of UAV based on aerodynamics, landing, and multi-rotor

3.2.1 Based on Aerodynamics

Wide varieties of UAV systems have been worked upon, which includes the fixed wing aircraft [1], chopper [2], multi-copter [3], vertical aeration and piloting [4], details on various part in the making of UAVs, and getting the UAVs into the market.

Fixed wings are the key uplifting components which work on accelerating the current phase ahead. Lifts generated are systematically influenced by angulation of the UAVs along with its initial phase of aeration which influenced on fixed wings. A thrust-to-load ratio lower than one and higher is essential to get the most out of fixed wing drones when launching the flight [5]. It must be noted that fixed wing drones are higher in sophistication and does utilize appreciable amount of power when compared to multi-rotor on some of payload [6]. Controlling this aircraft does require roll, pitch, and yaw maneuvers. Direction, aeration, and positioning are taken cared by roll, pitch, and yaw, respectively. Yaw, roll, and pitch are angulated by rudder, ailerons, and elevators on the surface of aircraft shown in Fig. 3.2.

Given that fixed wing drones are getting higher popularity due to its sophistication and compatibility with larger lift-to-drag ratio (L/D ratio) and with sophisticated Reynolds number, they cannot aerate on low speed and also hover still. Being well observed the fact that the lift-drug points on lift generated by a trailing wing are counter generated, fixed wing drones are less valued for L/D. Small hummingbirds to large dragonflies [7] have been an inspiration for developing flapping wing drones. Not only the birds on the whole have been driven inspiration from but also the feathers of the above further helped in designing lightweight and flexible wings in aerodynamics. Although the inspiration came easy from the birds, it is a big technological challenge to work on accounting to the complexity in aerodynamic for creating flexible and weightless flaps [8].

Fig. 3.2 Pitch, roll, and yaw in an aircraft (NASA official: Tom Benson)

In a multirotor, main rotor sharp edge delivers a powerful push helping on aeration and piloting. Multirotor unmanned aerial vehicles aid **vertical** take-off and **landing** (VTOL) and also as a special feature of hovering still which unfortunately fixed wing aircraft are not capable of. Furthermore, the practice of flexible root bars as an alternative of universal joints significantly reduces the machinery complexity. Instead, the wing twist modulation model requires a specific wing design, whose lift force differs almost linearly with the root bar deformation [9]. The sole draw back with multirotor is they require ample amount of power to function. Abbott equations are used to numerate power and thrust requirements [10] where RPM (Revolutions per minute)

$$\text{Power } [w] = \text{Pitch} * (\text{Diameter})^3 * (\text{RPM})^2 * (10^{-10}) \tag{3.1}$$

$$\text{Thrust } [oz] = \text{Pitch} * (\text{Diameter})^3 * (\text{RPM})^2 * (10^{-10}) \tag{3.2}$$

Alteration in acceleration and deceleration of thrusting propeller/motor elements are used in functioning of multicopter. It is categorized into various classes keeping number and positioning of motors as sole components; every peculiar mission is specialized by a particular class. Because of which many number of configuration take birth, such as Bicopter, Tricopter (Y3, T3), Quadcopter (X4, Y4, V-Tail, A-Tail), Pentacopter, Hexacopter (Y6), and Octocopter (X8) [10].

3.2.2 Based on Landing

Horizontal take-off and landing (HTOL) are worked upon further from fixed wing aircraft as they happen to have a well-appreciated phase in its cruise and a smooth landing to end the aeration. Vertical take-off and landing (VTOL) cruise speed is a main detaining factor for vertical take-off and landing (VTOL) drones even though they are well versed in flying, landing, and vertical hovering [10]. Various automatic landing systems based on the GPS (global positioning system), INS (inertial navigation system), ILS (instrument landing system), and tracking radar may not be applicable due to the complexity and operational environment costs/limitations. The vision-based landing proved to be attractive as it is passive and inexpensive and does not require special equipment other than a camera and on-board vision processing unit. An uncalibrated camera is utilized to generate high-precision position information for automatic landing on the runway with a speed sensor because airspeed is very important during landing. The superior distinctive of this method is that neither a calibrated camera is required nor the recognition of special points of the track and its 3D location must be experimented.

3.2.3 Based on Weight and Range

Engineers find weight and range as a solid parameter for classifying drones as listed in Table 3.1. Top 10 best value drones under 7100 ₹ in India 2020 are shown in Fig. 3.3. The best drones for 2020 are shown in Fig. 3.4.

3.3 Hardware Design and Challenges

Unmanned aerial system design involves a link between UAV and user and vice versa, and also components such as stations for ground controlling are involved. The design of UAV has head to tail programs for aerial vehicle as it can start from vehicle framing to aeration of the vehicle. The crucial task begins in picking up of elements such as airframe, controller, propellers motor, and power supply. The need of in-depth knowledge on mathematical designs to program a UAV for a particular operation is vital. Figure 3.5 labels the subsystems and also modules for programming an UAS.

(1) Aircraft design: The challenge begins when the complexity of the type of application used limits the reporting area phase and climbing rate. The key strings of an aircraft subsystem are assessment of inertia, motors, airframe, propellers, central processing unit, and receiver [10]. Alloys, aluminum, and titanium are the most

Table 3.1 Classification of UAV based on weight and range

Type	Range (km)	Weight (kg)
Nano-sized	<1	<0.025
Micro-sized	<10	<5
Mini	<10	<20
Close range	10–30	25–150
Short range	30–70	50–250
Medium range	70–200	150–500
Minimal altitude low endurance	>250	250–2500
Minimal altitude high endurance	>500	15–25
Marginal altitude low endurance	>500	1000–1500
Maximal altitude low endurance	>2000	2500–5000

Fig. 3.3 Top 10 best value drones under 7100 ₹ in India 2020

Best Drones	DJI Mavic 2 Pro	DJI Inspire 2	DJI Mavic Air 2	Autel Robotics EVO	DJI Mavic 2 Zoom	Parrot Anafi	DJI Spark	Ryze Tello	DJI Mavic Mini
Dimensions	3.3 by 3.6 by 8.4 inches	12.5 by 16.7 by 16.7 inches	3.3 by 3.8 by 7.1 inches	5.5 by 5.5 by 10 inches	3.3 by 3.6 by 8.4 inches	2.5 by 2.6 by 9.6 inches	2.2 by 5.6 by 5.6 inches	1.6 by 3.9 by 3.6 inches	2.2 x 3.2 x 5.5 inches
Weight	2 lb	7.3 lb	1.3 lb	1.9 lb	2 lb	11.3 oz	10.6 oz	2.8 oz	8.8 oz
Rotors	4	4	4	4	4	4	4	4	4
Obstacle Detection	✓	✓	✓	-	✓	-	✓	-	-
Integrated Camera	Integrated with Gimbal	Integrated with Gimbal	Integrated with Gimbal	Integrated with Gimbal	Integrated with Gimbal	Integrated with Gimbal	Integrated with Gimbal	Integrated without Gimbal	Integrated with Gimbal
Video Resolution	4K	5.2K	4K	4K	4K	4K	1080p	720p	2.7K
Megapixels	20 MP	20 MP	48 MP	12 MP	12 MP	21 MP	12 MP	5 MP	12 MP
Media Format	microSDXC	microSDXC	Internal, microSDXC	microSDXC	microSDXC	microSD, microSDHC, microSDXC	microSDXC	None	microSDXC
Remote	Dedicated with App	Dedicated with App	Dedicated with App	Dedicated with LCD	Dedicated with App	Dedicated with App	Smartphone/Tablet App, Dedicated with App	Smartphone/Tablet App	Dedicated with App
Live Video Feed	1080p	720p	1080p	720p	1080p	Yes	Yes	Yes	Yes

Fig. 3.4 The best drones for 2020 (Jim Fisher from PCMag India)

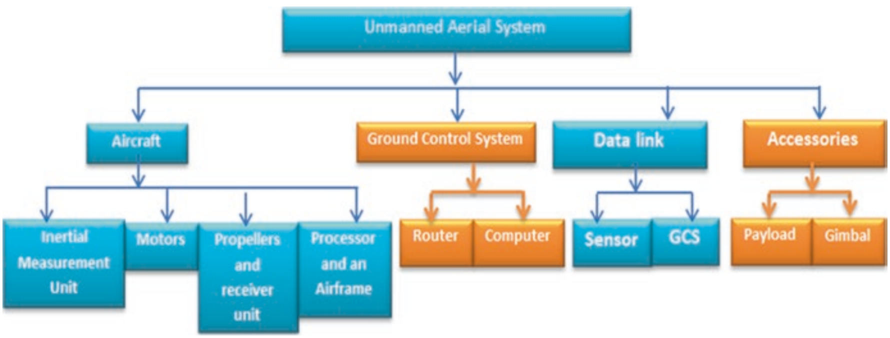


Fig. 3.5 Unmanned aerial system subsystems

common metals in manufacturing an aircraft. On the other hand, non-metallic substances such as transparent and recycled plastic are utilized. Electronic speed controller keeps itself useful by changing the force provided via motor. They can be in the air in a particular direction and also set their height by using signals based on the results of assessment of inertia. (2) Ground Control System: Computerized wireless router, by controlling ground system, could unlock a wide range of possibilities such as capturing, data processing, and data visualization. Ground control station is basically made into a standard to be able to act compatible with a variety of platform and also open system architecture [11]. (3) Data Link: Aircraft sensors and ground control station (GCS) have wide integration for running the computerized wireless router smoothly. The IEEE 802.11 wireless link is picked up to establish a link from the aircraft CPU and ground controller and vice versa. Routers should be planted with antennas which could scoop omnidirectional tracking for high gain which are utilized in lowering the path loss. Currently, modern antennas work on 2.4 GHz and a minimal gain of 12dBi. (4) Accessories: Multispectral, thermal, hyper-spectral, digital camera, and film imaging units are made compatible to UAV. The above are utilized to carry out photogrammetry, film shooting, and field mapping. Thermal and hyperspectral cameras seem to be appreciated more in remote sensing. Mining, oil, and gas industries use drones equipped with thermal sensors. Table 3.2 tabulates the achievements in wide variety of spectrum on sensors for UAV.

Table 3.2 Developments in multispectral sensors for UAV

Model	System summary	Image size	Optics	Weight (g)
MCAW	Embedded Linux computer system with sync'd capture interface to multiple snap shutter sensors	6 images of 1280 × 1024 pixels 8 or 10 bits/pixel	9.6 mm fixed lens	550
Micro-MCA + tau	6 Mpel configurable multi-spectral camera	7 images of 1280 × 1024 pixels 8 or 10 bits/pixel	9.6 mm fixed lens	900
ADC lite	3.2 Mpel multi-spectral R-G-NIR system	1280 × 1536 pixels 8 or 10 bits/pixel	8.0 mm user changeable lens	200
ADC micro	3.2 Mpel multi-spectral R-G-NIR system	1280 × 1536 pixels 8 or 10 bits/pixel	8.43 mm fixed lens	90
ADC snap	1.3 Mpel multi-spectral R-G-NIR snap shutter system	1280 × 1536 pixels 8 or 10 bits/pixel	8.43 mm fixed lens	90
Micro-MCA	5.2–15.6 Mpel configurable multi-spectral camera	4.6 or 12 images of 1280 × 1024 pixels 8 or 10 bits/pixel	9.6 mm fixed lens	uMCA-4500 uMCA-6530 uMCA-121000

3.4 Path Planning Overview and Issues

Having a path planning technique is imperative for computing safe path with minimal expenditure of time to reach the final desired destination. It consists of motion planning, trajectory planning, navigation, global path planning, and local navigation. It is prominent that in path planning in complex condition, the 2D (two-dimensional) strategies are not proficient to perceive the odds and other computerized wireless objected when compared with 3D (three-dimensional) path planning techniques, In such manner that it is sophisticated, a path planning for 3D (D^3) location with static *obsts* $O = \{O_1, O_2, \ldots O_n\} \subset D^3$, from P_{start} to P_{target}, is as shown in Fig. 3.6. This situation includes a robot traveling through six dividers with minor fixes. The robot is bigger than the gaps and needs to twist to produce a crash free way the underlying setup to the last design [12].

At that point, the problem on path planning for D^3 location is defined as (P_{start}, P_{target}, W_{free}) which comprises the following functions: [0, T]→D^3 is clearly expressed in the bounded region where T is characterized as a time. At that point, the following holds (0) = P_{start} → at beginning time (T) = P_{target} → at target time W_{free} → workspace without obstacles there exists, $\varnothing = \delta(\beta) \in W_{free}$ for all, β in [0,T]. Then, Ø is entitled path planning of UAVs. A perfect path is clearly expressed as $\delta'(c, t, e)$ = optimum of $\delta(c, t, e)$, where δ is the function of set of all feasible path and δ' is an optimal path computation function. At that point, the following holds where c → cost (c), t → time (t), e → energy (e) should be minimized. The energy communication of UAVs base-station could be brought to a minimal amount on decreasing the power used in transmission. In the same way, there is a need for minimized mechanical support and resources. The energy-efficient consumption model in UAVs [13, 14] is shown in Eq. 3.1.

$$E = \left(P_{\min} + \alpha \mathrm{h}\right) + \left(P_{\max}\right)\left(\mathrm{h}/\mathrm{s}\right) \tag{3.3}$$

whereas t → the operating time, h → the height, and s →the speed of the UAVs. P_{min} → minimum power needed to start the UAVs and α → motor speed multiplier. P_{min} relies upon weight and engine characteristics. Hence, the total communication cost (T_{com}) to limit the time and cost in UAVs correspondence framework is shown in Eq. 3.2 (Shubhani Aggarwa, 2020) [14].

$$T_{com} = t_{startup} + \left(t_{overhead} + t_{hop}\right) l \tag{3.4}$$

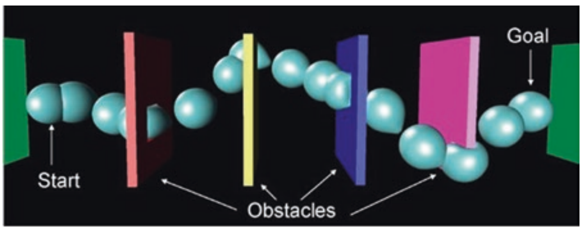


Fig. 3.6 Spherical robot through walls [12]

whereas $t_{startup} \rightarrow$ UAVs start-up time, $t_{overhead} \rightarrow$ overhead time, $t_{hop} \rightarrow$ UAVs per-hop time and $l \rightarrow$ source and target communication links. Aside from these constraints, robustness, completeness, and collision avoidance aspects need to be measured for finding the optimality in path planning of UAVs.

3.4.1 Steps in the Path Planning

Path planning of the UAVs is spoken to as '*U*' comprises of two stages as follows. The main stage is the pre-preparing stage. In this stage, hubs and edges are drawn on the workspace 'W' with obstructions '*O*'. At that point, the idea of the configuration space (c-space) to portray *U* and *O* on *W* is applied [13]. For generating the graph maps, representation techniques are applied [15]. Identification of UAVs is done with help of the query phase. Graph-based search algorithms Dijkstra's algorithm [16], regular chain of segments algorithm (RCS) [17], Floyd algorithm, and lazy counting-based splay tree algorithm [18] to name a few are used in query stage. Probabilistic models, Q-learning [19], mixed integer linear programming [20], and bio-inspired models like intelligent water drops [21] can also be utilized for path planning of UAVs. For representing the C-space on workspace *W*, there are an ample amount of path planning methods such as potential fields [22], cell decomposition [23], and roadmaps [24].

3.4.2 Challenges in Path Planning

Ample research recommendations are being talked about in the yester years to take care of path planning difficulties on UAVs [25]. For instance, Marina Torresa et al. [26] defined the regions by different sweep direction to discover an optimal path. Additionally, Torres et al. [26] investigated to lessen the distance between sub-areas by the back and forth pattern. Also, Balampanis et al. [27] projected the hybrid and approximate decomposition technique. Along these lines, Acevedo et al. [28] suggested spiral-like pattern in complex coverage areas. Path planning algorithms obtain an available drone path with hazard avoidance autonomously. Many parameters such as getting the path completed, optimizing the path, minimizing the length of the path, making it cost and energy efficient, and reducing the odds have been noted in path planning techniques.

3.4.2.1 Multi-Cloud Path Planning with Multi-criteria Decision Making

The multi-robot system is established on cloud technology with a high level of autonomy, which is set to play an increasing role in responding to the COVID-19 pandemic. The multi-cloud approach shifts agent computing load to the cloud and provides influential processing skills to the multi-robot system. In order to improve the efficiency of trajectory path planning, the implementation of multi-criteria decision-making (MCDM) while using a full consistency method (FUCOM) is utilized [29]. Mustafa Hamurcu et al. [30] promises to determine the weight of factors influencing robot movement, with regard to mission specificity that involves managing multiple risks from diverse sources, thus optimizing the global cost map.

3.4.2.2 Internet of Drone-Based Data Analytics in Multi-Cloud

The latest technology behind Internet of Drone (IOD) safe exploitation on commercial and public use presents communication and computational challenges in aspects of the real world. Assingning individual tasks to multiple drones and using the data center values on the side, data are transferred to the cloud for technically balancing the data from an inaccessible area [31]. Methods for analyzing drone data with great efficiency in the areas of progress monitoring, inspections, and surveys to analyze the data to make key decisions are identified. The processing of drone data has just extended into the cloud with bandwidth management for data processing, and on the basis of image detection, an effective decision will be taken to protect human life and property [31].

3.4.3 Path Planning Techniques in UAVs

Ample number of time was invested in UAVs as it shows great promise. The versatile stages for path planning endeavor are (1) programing 3D environment and (2) graphing for the programmed 3D ecosystem. For starters, representation technique is based on configuration (Geraerts, 2010) [23], roadmaps (Ryan DuToit) [32] (Kwangjin Yang, 2008) [33], and potential field (Alex Nash) [34]. Furthermore, the next in line are integrating swarm intelligence algorithms (Hrabar, 2008) [35], simulated annealing method, to name a few (Shubhani Aggarwa, 2020) [14]. Optimization problems would be easily solved by path planning algorithms. The exact behavior of UAVs can be easily studied by the above algorithms.

3.4.3.1 Representation Techniques

Getting the UAVs projected into the 3D environment is a phase maker, and it is done using AI and sampling; the naming of those techniques is as shown in the Fig. 3.7.

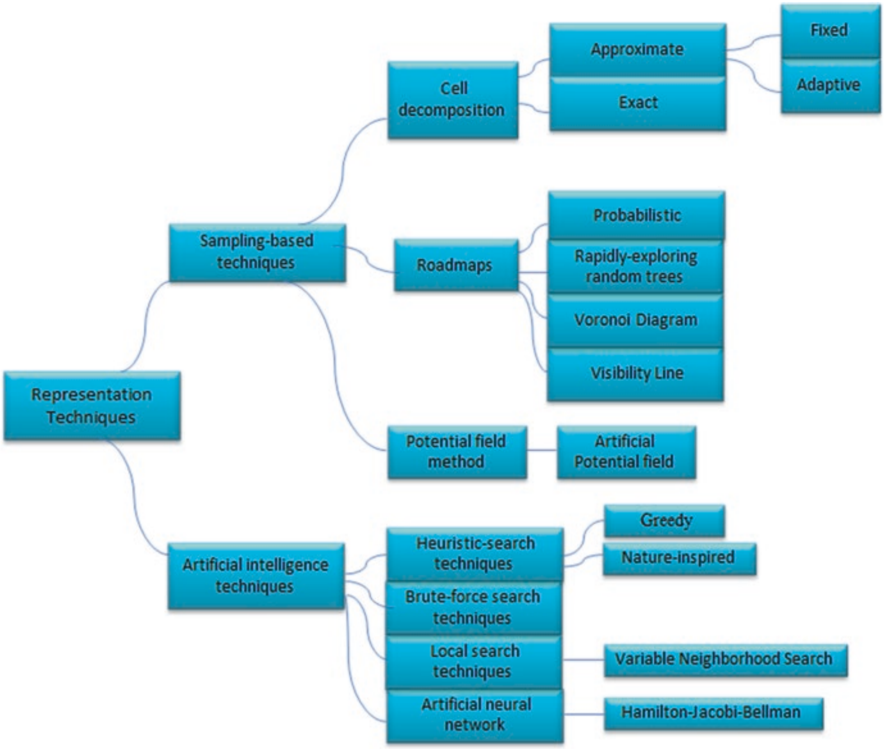


Fig. 3.7 The representation techniques

Sampling-Based Techniques

These techniques need the programming 3D environment that has already been preprogrammed. Division into nodes and making a map out of it are achieved via algorithm that undertakes direction and path of the UAVs. Rapid-exploring random trees (RRT) [32], RRTstar (RRT*), A-star (A*), probabilistic roadmaps (PRMs) with D* Lite (PRM) [36], particle swarm optimization (PSO), and improved intelligent water drop algorithm (IIWD) [21] are some of the versatile methods for sampling mechanisms.

Cell Decomposition

Creation of a safe path for intra- and inter-cell path is taken cared by cell decomposition. Types of cells, space, various strategies, and calculations are the key factors for basing the cellular organization in UAV path planning. The exact cellular decomposition works by dividing the area of interest and works into varied amount of sub-areas. In this way, reduction of the coverage path problem (CPP) could be achieved to motion planning [37]. Decomposed cells could be swept to the site of need. Adjacency graph takes care of the decomposition, and a search can be performed to get to the connected path for picking up the nodes. Trapezoidal decomposition and boustrophedon decomposition help in the process. Formation of trapezoidal shaped cells is how the trapezoidal decomposition works. The other now creates non-convex cells. The boustrophedon decomposition decreases the number of trapezoidal cells for covering the distance of the path of travel and its path, which in turn is better than trapezoidal decomposition.

An approximate cellular decomposition is a fine-grid-based depiction of the free space [38]. Here, the cells are all the identical size and shape, but the combination of the cells only approximates the target area. This idea was pioneered by Elfes [39] and Moravac [40]. These regular cells normally take up many forms. Grid-based depiction does help in the formation of paths [41]. For example, Franklin Samaniego [37] worked on interpreting path planning by comparing the RRT, PRM, and ECD-PRM. The exact and approximate cell decomposition on a roadmap is interpreted by them as shown in Figs. 3.8 and 3.9 [42]. End result brings RRT, PRM, and ECD-PRM better than in MACD.

These algorithms aid in cell decomposition. Similarly, authors in [43] worked on the A*, RRT, and PSO algorithms and published RRT algorithm which in turn performs far better when compared to A* and PSO. The total cost is calculated by adding heuristics expenses to the expense of the travelled path by a UAV. The lowest expense is always preferred. Hence, there is no need to visit all nodes which are depicted in Fig. 3.10 and Table 3.3. The mapping used by A* in distance calculation is as follows:

$$f(n) = g(n) + h(n) \tag{3.5}$$

in the calculation 5, $f(n)$: a heuristic function that calculates, $g(n)$: the cost of access from the start node to the current node, and $h(n)$: the distance of the path to be travelled from the start node to the destination node is the estimated distance. The heuristic values are S ->5,A ->4,B ->5, and C ->0.

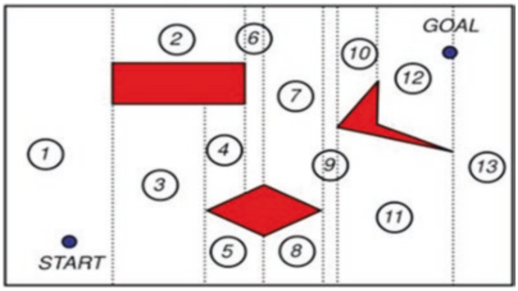


Fig. 3.8 Exact cell decomposition

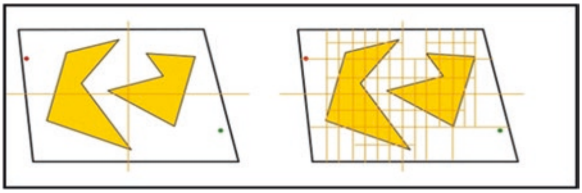


Fig. 3.9 Approximate cell decomposition (Nourbakhsh, 2004) [42]

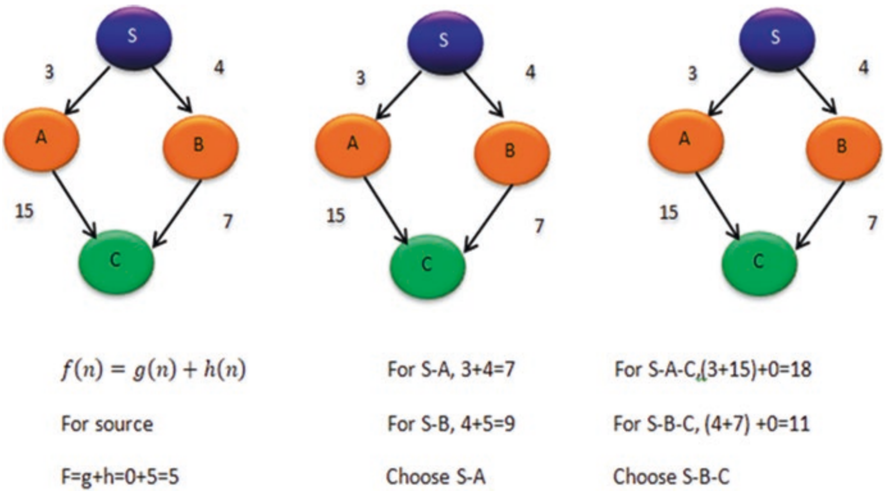


Fig. 3.10 Sample nodes for A * algorithm

Roadmaps

(1) Construction and (2) query are components in roadmaps. 3D environment is used in construction stage and then comes the query stage. The concatenation of curves aids in path planning. The third stage makes the path planning even more sufficient. The computation does not take time as it worked up during processing. As-Rapidly-Exploring Random Tree (RRT), RRT*, Zammit, and E. Van Kampen [44] introduced ways to estimate the concert of the A* and RRT in making the path planning even more sophisticated. It is inferred that A* is better than the RRT in many aspects. But either can be used for path planning.

1. Probabilistic roadmap method (PRM) aids in trajectories that are very efficient and reasonable in UAV path planning. Similarly, Zhuang et al. [45] considered this probabilistic roadmap ideology for path planning in many innovative places like nuclear facilities as they have a radioactive surrounding. The issue of defining a path from the start configuration to the end configuration is wonderfully taken cared by probabilistic roadmap planner. The thick line in Fig. 3.11a picturizes the pathway. For instance, Mansard et al. [46] proposed on Kino dynamic probabilistic roadmap.

Table 3.3 A * algorithm

```
Create an open_list of only the start node.

Create a closed list blank.

 while (the goal node has not been made):

         Consider the node with the lowest fscore in the open list.

    if (this node is our target node): we are done.

    if not:

            Place the current node in the closed list and watch all its neighbors.

       for (each adjacent to the running node):

            if (the neighbour has a value g smaller than the current value and is in the
            closed list):
                  change the neighbor to the new lower g-value.
                  The current node is right away the neighbor's parent
            else if (the current g-value is lower and this neighbour is in the open list):
                  change the neighbour to the new lower g-value.
                  change the neighbor's parent to our current node
             else if this neighbor is not on either lists:
                 add it to the current list and adjust its g.
```

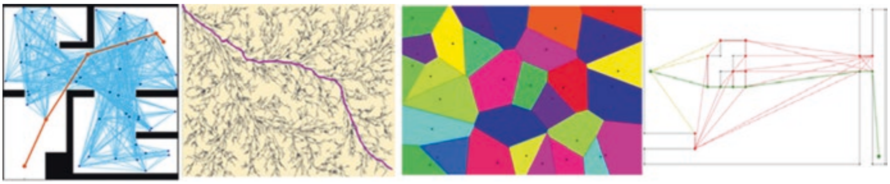


Fig. 3.11 (**a**) Probabilistic RM (**b**) RRT (**c**)Voronoi diagram (**d**) Visibility line

2. Rapid-exploring random trees: Rapid-exploring random trees (RRT) are expressly intended to deal with no holonomic constraints (including dynamics) and high degrees of autonomy. It is iteratively reached output on control feedbacks that drive the framework somewhat toward arbitrarily selected focuses, as inverse to expecting point-to-point convergence [47] as shown in Fig. 3.11b. An advancement of RRT, i.e., RRT-Connect [48], developed a much more versatile method that is now used for getting an accident free path. For instance, Zhang et al. [49] proposed RRT-Connect and combined it with an area that has a promising potential artificially which contributes an optimal path for UAVs. It has the haphazardness and efficient planning methods, which alters the planned way of coverage nearer to the efficient travel path than the path of the single procedure.

For instance, in order to make the UAVs more sophisticated, Yang et al. [50] introduced environmental potential field-based RRT (EPF-RRT), as natural potential field is incorporated, target points produce virtual gravitational power and hindrances are repulsive, and the subsequent energy drives the RRT to dismiss into from the deterrent and close by to the goal. In this way, the productive arrangment of the path is done using the RRT. Zu et al. [48] were in urge to program a wide number of directions and path for the drones progressively, from beginning areas to objective areas in the occurrence of unknown pop-up interruptions. They also suggested that when given a path with obstacle, the UAVs would easily find another odd free path with the help of this algorithm. Qinpeng Sun et al. [49] introduced another improvement, for example, bidirectional-RRT; an improved variant of RRT contributes an extraordinary achievement phase and effectiveness in UAV path planning shown in Table 3.4.

O. Adiyatov et al. [50] utilized sparse-based RRT* algorithm for ques. about the odds as a binary variable (RRT*K) or with a quadrant-based representation (RRT*Q) to the database of the node to make it optimize the path. The above helps in getting a faster result and converges to the optimal result with fewer numbers of places of nodes.

According to Iram Noreen et al. [51], RRT*-adjustable bound (RRT*-AB) is a sampling-based planner which improved time and space requirements having quick convergence rate than RRT*. These points of interest of memory-efficient A* (MEA*) mark it reasonable for off-line requests utilizing trivial robots with well-ordered power and memory resources. But, RRT*-AB will overtake MEA* in high-dimensional problems for having developed with ensembling capability.

3. Voronoi diagram: Voronoi diagram (VD) is a diagram which divides the plane into smaller paths having close to each of a given set of objects as shown in Fig. 3.11c. The sophistication of the paths is achieved by Waypoint Path Planner (WPP) HanTong [52] by taking the initial path which was created by VD. For

Table 3.4 RRT algorithm

T = (V, E) ← RRT(z_{init}) // initial state of the tree
T ← TreeInitialize();
T ← NodeInsert(Ø, zinit, T); // adds a node z_{new} to get z_{min}
for i=0 to i=N do
z_{rand} ← ConfigurationSample(i);// random state from configuration space Z
$z_{Nodenearest}$ ← NodeNearest(T, z_{rand}); // nearest node from T to z_{rand}
(z_{new}, U_{new}) ← Steer ($z_{Nodenearest}$, z_{rand}); //control input U
if FreeObstacle(znew) then
T ← NodeInsert(zmin, znew, T);
return T

example (Mengxiao Song et al. [53]), based on VD and watershed segmentation procedure, global seam-line network generation technique for mosaicking UAV orthoimages is developed. Xin Feng et al. [54] utilized heterogeneous Voronoi graph for a path planning which involves emergency drone delivery. Shen et al. [55] suggested an enhanced kind of VD to divide the sharing space which is from a very low attitude on many different areas. Taking help from Automatic Dependent Surveillance-Broadcast (ADSB), track method is set up to regulate and screen drones and reduce the chance of collision and ensure security between manned and remote-controlled aircraft systems.

4. Visibility Line: Visibility line is utilized to discover a result of autonomous mobile robot path planning in an unknown obstacle location [56]. The VL was built steadily which is shown in Fig. 3.11d. A learning component was consolidated so as to build the VL. Moreover, a sensor with constrained range was utilized to acquire data around the odds in the specified location. Notwithstanding, the produced path was not enough by reason of the inaccessibility of whole data about the specified environment. Rao [57] recommended a broad-spectrum framework of robot navigation that could be functional to any new location including mobile robots where an appropriate steering could be commenced using the Restricted Visibility Graph (RVG).Wooden and Egerstedt [58] derived oriented visibility graph (OVG) strategy which fundamentally decreased roadmap for unstructured polygonal situations suitable for real-time path planning application of outdoors robots. Thus, as to heighten the performance over runs, the intermediate graphs were kept between runs, and dynamic update instructions were specified. Huang Sunan et al. [59] recommended a novel technique founded on visibility graphs by utilizing the path with no odds and minimal distance of coverage in all 2D path planning and then collecting the paths that get a minimal distance to cover to reach the destination in the complete 3D environment.

Potential field method (PFM) denotes the location to particulate an odd. Its path of travel is hugely influenced by areas of potential around the c-space. The path of the drones is well determined basing on the areas of reluctant from the initial point to the destination point. Notwithstanding, the regular PFM experiences the local minima making the UAVs stuck before it arrives at the objective. For example, Budiyanto et al. [60] recommended the hierarchical PFM by incorporating some rotational force between the UAVs to resolve the problem of inaccessible destinations and collision challenges among the UAVs.

So as to upgrade the PFM for multi-UAVs, Bai et al. [22] presented a longitudinal random factor, and B-spline interpolation is utilized to resolve the problem of dropping into local minimum and to smooth the planned route respectively. Authors in [22, 60, 61] indicated that there are numerous techniques to overpower the restrictions of PFM and utilize for collision avoidance in path planning of UAVs. For example, Abeywickrama et al. [62] proposed an improved variety of PFM, which is utilized to plan the path and to draw the UAVs for the preferred goal configurations. It helps the drones to steer clear from the odds thus avoiding accidents.

Artificial Intelligence Techniques

It is a disposition to program a wireless device with could function as a robot that can be unsurprisingly composed by the software plug-ins and meditates cleverly similarly an astute human thinks. It is a part of computer science, pointing on building machines and programming with knowledge like people so they can perform comparative reasoning, thinking, dynamic, critical thinking, and natural language processing like human. It is an approach to utilize and arrange the information proficiently and viably with the goal that it tends to be utilized much of the time. It is an innovation dependent on the advancement of technologies like engineering, mathematics, and biology.

Heuristic-Search Techniques

These techniques are utilized in UAVs for finding sophisticated path by minimizing the expanse using cost estimation programming methods. Revenues from AI market worldwide from the beginning point to the end point are collected and optimized in the cost estimation method. For example, Park et al. [63] recommended the DroneNetX framework for network by accompanying aerial wireless links into the quarantined ground network by means of UAVs. It does searching of connection between the computers from scratch and discovers critical locations where the path is sophisticated.

Essentially, Carpin et al. [64] proposed a procedure which, dependent on greedy technique, is the most uncovered first (MUFT) that conquers the problem of "cooperative multirobots observation of multiple moving targets" (CMOMMT) by utilizing many UAVs. Arantes et al. [65] proposed genetic with greedy technologies to evaluate the hardware glitches and path re-planning of UAVs. Decision-making algorithms are soulfully done using in-fly awareness (INF) security system. For instance, Li et al. [66] presented a genetic algorithm for searching and rescue. They have applied this algorithm to optimize the priority field to help the smart energy cycling and improve the performance of UAVs. The smart energy cycling has been appreciated more by using energy-efficient trajectories. In Zhou et al. [67], point was made energy efficient. Specifically, UAV-aided mobile crowd sensing MCS system has anytime and anywhere accessibility, yet it lacks the good battery; thus, Gale–Shapely algorithm and an energy-efficient perspective approach are used to improve the battery.

Xixia et al. [21] suggested using multiple swarms (multiswarm) IWD (LMIWD) algorithm for optimization of path planning. LMIWD algorithm is depended upon for cooperation and competition among UAVs. It is inferred that this method outperforms random search, greedy search, and particle swarm optimization (PSO) algorithm.

Brute-Force Search Techniques

Brute-force techniques do require a lot of theory as they work on small states, yet they do expect valid operators and initial sates to work with for building the description. Breadth first search and depth-first search are two of the most vital components in brute force techniques. Sharma et al. [68] worked on depth first search programming. Farid et al. [69] worked on waypoints-based trajectory generation to improve the quality of UAVs.

Local Search Techniques

Hard problems are solved using local search techniques. Optimization of values is done using these techniques which uses hill climbing and TSP as the main components. For example, Huang et al. [70] helped in finding a short path panning ideas for the UAVs. Cluster algorithm and PSO algorithms used for doing so small coverage of the area by UAVs are addressed using local search techniques. Similarly, Wang et al. [71] further helped his method by giving a 2-tie search algorithm for lost UAVs. According to Perazzo et al. [72], location finding and identification in UAV path is sorted using TSP algorithm and LocalizerBee. It is observed that the heuristic-based genetic algorithm is basically used in getting a path for UAVs. Bit-Monnot et al. [73] proposed variable neighborhood search (VNS) swiftly to produce planning with multiple UAVs that are fine grained even though they span over large spaces and long periods.

Artificial Neural Networks

Artificial neural network (ANN) is utilized to locate an optimal path planning for UAVs. For instance, Kurdi et al. [74] tackled the route and position issues of UAVs by utilizing neural network. In this scheme, ANN takes response from the Global Positioning System (GPS), Robot Vision System (RVS), and quad-copter vision system (QVS) and gives out the optimized path localization to the UAVs. And then also, Zhang et al. [75] used a self-adjustable integral line-of-sight (LOS) guidance to make the UAVs to act decisively in unfavorable conditions. They have additionally introduced control method for path planning to locate an optimized path for UAVs. Hamid Shiri et al. [76] proposed Hamilton-Jacobi-Bellman equation (HJB) in real time, yielding the control decisions even when the connection is lost.

3.4.3.2 Cooperative Techniques

Cooperative techniques are substantial to identify ample number learning approaches that can be investigated to relate from many classifications like problem-solving and graphic organizers. They consist of machine learning, multi-objective optimization, and mathematical and bio-inspired models that are utilized in getting a direction for the drone to reach the target. The taxonomy of the cooperative techniques is presented in Fig. 3.12.

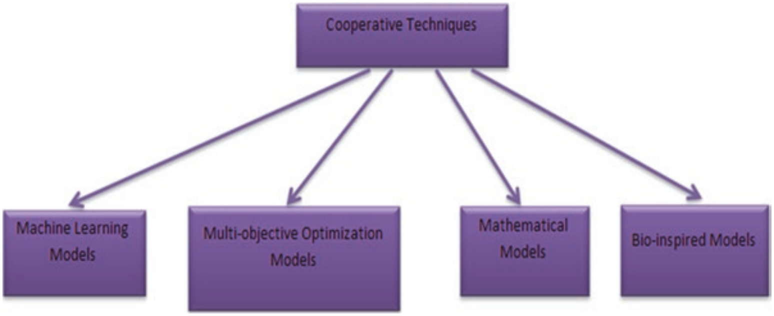


Fig. 3.12 Taxonomy of cooperative techniques in UAV path planning

Machine Learning Models

A variety of computer systems responds to the situations automatically without a need for getting explicitly programmed. They are divided into supervised, unsupervised, and reinforcement learning categories which are utilized in getting a sophisticated direction for the path of travel to reach the destination for the drones. First, identify and collect the relevant data from several resources, then select the suitable machine learning algorithm, and finally different data sets are utilized to train the model; based on that, the optimized path is envisioned.

In supervised learning, an algorithm for path planning procedure is trained yet unable to figure out the exact role. It constructs the model where dependencies occur between the input factors and expected the target value like Gaussian filter (GF) and Kalman filter (KF). Non-linear least-square method was suggested by Jiang and Liang [77] to estimate UAV path planning in a threat atmosphere.

Likewise, Kang et al. [78] suggested the Kalman filter (KF) algorithm to resolve noise and deliver the protected flight path to UAVs in a challenging atmosphere. Also, Wu et al. [79] commended KF algorithm to resolve many challenges like noise in the air, collision probability, cluster state approximation, and track planning of UAVs. They have validated that the collision probability among the UAVs is just 0.2% by using KF, but rate of error does not reach an acceptable level. Thus, to tackle the error, Yoo et al. [80] suggested another filter algorithm Gaussian in UAVs. To compute the posterior density in UAV path planning, the KF update equations are applied directly [81].

In unsupervised learning, descriptive modeling and pattern recognition methods are used. Clustering algorithms like k-mean clustering are utilized in UAV path planning [82]. For instance, Farmani et al. [83] used clustering algorithm for tracking the multiple targets and KF to identify the precise position of the target in UAV path planning which shows the reduced time complexity. Similarly, Tartaglione and Ariola [84] presented quality threshold clustering based on an obstacle avoidance strategy for searching landmarks.

Reinforcement learning is a constant process and always studies from the environments in an iterative way. The reinforcement learning [85], deep reinforcement learning [86], and deep Q-network [87] are utilized for the optimized path planning of UAVs. For instance, S. Luan et al. [88] proposed the G-Learning procedure; simultaneous computation and recognition is achieved by cost matrix which also aids in location-based information for accident free path traveling in concurrently and reorganized based on the distance of geometric and information. Zhang et al. [89] proposed the Q-learning algorithm for extracting an optimal or suboptimal path for UAVs. Similarly, Yijing et al. [90] put forth this algorithm for creative smooth usage without any prior knowledge and disturbing the environment by taking adaptive and random exploration into account. Bouhamed et al. [91] created a Deep Deterministic Policy Gradient (DDPG) for navigation of UAVs to travel without any abstinence.

Multi-objective Optimization Models

This model takes into account the convex optimization [92] and two-echelon optimization [93]. C. Yin et al. [94] utilized safety index amps for optimization for taking a picture of obstacles in the path ahead. Luo et al. [95] suggested two-echelon optimization to do so. In Angley [96], proposal of large number of targets is better understood using optimization of the research and other vital strategies. To conclude, they also found that it increases the results by 10 percent. Koohifar et al. [97] proposed the steepest descent posterior Cramer–Rao lower bound for getting in hand with ratio frequencies. Ti and Li [98] showed some work on as-joint task, resource allocation, and computation offloading. They improved convex optimization method. Similarly, Zeng et al. [99] reduced the time of travel to 50% by working on the algorithm. Y. Zeng et al. [100] used a rotary wing to connect with multiple ground nodes (GNs) to be environmental protective. Jeong et al. [101] connected mobile and UAVs for optimization. Lee and Yu [102] tried to optimize using the concept of gravitation and energy.

Mathematical Models

It takes in arithmetical stuff like Lyapunov function [103], Bezier curve [104], and Ergodic Exploration [105] which are used for working on optimizing the path. They do involve many algorithm and models to optimize the path. For example, Mathew et al. [106] worked on traveling salesman problem and recharging the programs. Likewise, De Waen et al. [107] made trajectory planning more worthwhile and forceful and also worked on scaling. Control theory includes non-linear predictive models like model predictive control method, Lyapunov function, and Dubin algorithm for UAV path planning. Keyu Wu et al. [79] observed that RRT programming goes well with PSO for getting the drone a clear and smooth path. Similarly, Luo's

[108] proposal is based on the work on algebraic equation to have a clear agricultural reliability.

Ergodic explorations of distributed information (EEDI) are used for creating low-cost path derivations [105]. Lyapunov creates a stability and control on versatile environment [109]. Darbari et al. [110] proposed Markov decision process (MDP) to develop an optimal direction for the drones to reach the destination with the decision process. Similarly, Yu et al. [111] proposed to use Bayesian filter and observable Markov decision process (POMDP) to develop an optimal path for UAVs. Eaton et al. [112] suggested the use of POMDP to make the UAVs less fragile in handling tough situation where it can handle reaching the target even in a cumbersome path which in turn is a much needed development as UAVs often fail to reach the targets. Q. Yang [113] look a step further and suggested adding Kalman filter (UKF) with POMDP to make the UAVs reach the cumbersome target. Alessandretti and Aguiar [114] traveled totally in a different path to make the UAVs reach the target via its path by using model predictive control (MPC) and angular B. Sun et al. [115] observed that the use of MPC made sense taking it a bit higher level to design the path. The cost function considering load swing angle and the distance between obstacle and UAV are designed and generate an optimal trajectory. Yel et al. [116] proposed a self-triggered framework for adding feathers to the path.

Non-linear model is based on at least one independent variable. It is a framework where change of output is not relative to the difference in input. Hausman et al. [117] proposed on updating for self-calibration applications by non-linear applications. Li et al. [118] proposed optimizing the path using parallel search options. Moustris [119] proposed a feedback linearization method for pointing out the location by coordinates which in turn aids to the complete state space and also in demonstration of the fact that there is an information change with the end goal that the dynamical conditions stay invariant.

Tian et al. [120] idealized the improved artificial bee colony (IABC) for helping the drones in escaping the cumbersome situations. Cheng and Li [121] called in for a genetic algorithm as it decreases the cost and gives out a pretty optimal path. Xixia et al. [21] proposed IWD to work on water population. However, it is significant that this technique could not totally avoid cumbersome paths. Likewise, Popovic et al. [122] model finds the presence of weeds on farmland to give an optimal path by getting data. Mostafa et al. [123] worked on getting chi-square interference.

Similarly, Ji et al. [124] suggested the 2-opt algorithm to enhance the operations like search and rescue of UAVs. The simulation outcomes illustrate that the 2-OptACO technique ensures a quicker phase of convergence than the GA and ACO. Similarly, Yang et al. [125] proposed Gaussian process (GP) regression method to finish the consignment drop mission with an optimal flight-time and reduced errors of landing in UAVs. In Marija Popović [81], in area monitoring of UAVs, the multi-resolution mapping is used.

Bio-Inspired Models

These models helps in solving the complications due to ecological factors like neurodynamics and physiological biology, Sun B et al. [126]. P. Fazio et al. [127] designated networks flying ad hoc NETwork (FANET) to aid in developed and theologized agricultural areas. Evolutionary techniques modified shuffled frog leaping algorithm for fighting of UAV path planning [128, 129]. It is a living mass determined meta-heuristic algorithm with variety of mathematical functions. In Jia Song [130], tackling of UAVs in 3D spaces is done using biogeography-based optimization (BBO). G. Tian et al. [120] used adaptive multi-objective evolution programing in the field of UAVs. Aditya A. Paranjape et al. [131] proposed using waypoint algorithm compared to birds and UAVs.

Z. Sun [132]took geosynchronous synthetic aperture radar (SAR) into consideration for transmitting signals to sophisticate the bio-inspired models. Similarly, Liu et al. [133] suggested the binocular vision-based technique for getting the path in all the environments.

S. Ren [134] aids in avoiding collision and reaching the destination faster. Yang et al. [113] used multiobjective programming for getting the right time to fly, mapping, safety, and much more. Kamel [135] used hierarchical fuzzy logic controller (HFLC) for getting UAVs some insane features. Shikai Shaoa [136] used comprehensively improved particle swarm optimization (PSO) for mapping and helps in optimality. Similarly, J. Chen [137] solved traveling salesman problems by using improved genetic algorithm in UAVs. They also proved that it helps in convergence and optimization of UAVs.

3.4.3.3 Non-cooperative Techniques

This acts autonomously, and one must be mindful of one another's standard and guidelines to determine the direction of travel for the drones. Search-based algorithms like circular digraph, Floyd method, and flood-fill graph techniques are utilized for path planning of UAVs as shown in Fig. 3.13.

Chen et al. [138] proposed ACO for path planning and greedy algorithm for task allocation in multi-UAV path planning and task allocations. It is necessary to decompose the task into sub-tasks and to decide which UAV should execute which

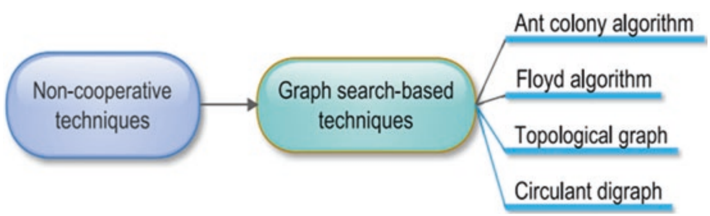


Fig. 3.13 Taxonomy of non-cooperative techniques (Shubhani Aggarwa, 2020) [14]

sub-task in the optimal time. Razzaq et al. [139] recommended the algorithm based on graph for providing deconfliction of UAVs. Morita et al. [140] proposed flood-fill and greedy 2-opt algorithms in annulled zones to extract the sub-routes for radiation dose mapping. Based on the results, the model outperforms if there should arise an occurrence of many hindrances. Yang et al. [141] explore the Floyd algorithm to judge the initial location of start, and Push Forward Insertion Heuristic algorithm (PFIH) is utilized for getting then optimized path. Du and Cowlagi [142] worked on an idea that although the UAVs can fly distance, 3D analysis and coordination could help the UAVs more. It is also observed that the repair cost decreases by 38.8%. Bogdanowicz [143] suggested the 360° mapping for army security, monitoring, and checking for distinctive arrangement of zones. The above strategy delivers a save and troubled path and maximizes the coverage area throughout search operations.

Coverage and Connectivity

Unmanned aerial vehicles are becoming increasingly important in a variety of applications, such as defense and security, emergency response and recovery, humanitarian and disaster relief aid, conservation of energy, disease control in humanitarian emergencies, healthcare and biomedical devices, agriculture and inspection of agricultural fields, weather forecasting and climate change, maritime and logistics management, waste management and sustainable development, energy and environment, mining and metallurgy, urban planning and transportation, telecommunications, game consoles and controllers, and space and universe. The connection between the drones and ground control stations (GCSs) happens to be vital for saving the information and sending information for their efficient functioning. For instance, Lee and Batsoyol [144] introduced wireless ad hoc network (WANET) to cover the area fully with minimum number of UAVs by utilizing Dijkstra's algorithm shown in Table 3.5 and Fig. 3.14, which delivers an optimal and collision-free path for UAVs in an urban location.

In Zouaoui H et al. [19], quality of experience (QoE) is improved by utilizing Q-learning reinforcement learning algorithm in the process of UAV path planning and prevention of service interruption avoidance between drones that is more than one drone serving the same field. In Challita et al. [145], reinforcement learning echo state network (ESN)-based procedure has been suggested to decide on path and resource allocation optimally.

Various authors like Bouzid et al. [146] suggested the Rapidly Exploring Random Tree Fixed Nodes (RRT*FN), Xiaojing et al. [147] proposed Biased Sampling Potentially Guided Intelligent Bidirectional RRT* (BPIB-RRT*), and Cabreira et al. [148] introduced the energy-aware spiral coverage and back and forth (E-spiral and E-BF) algorithms for UAV's path planning.

Table 3.5 Dijkstra's algorithm

```
1.  Dista[source]←0  //distance to source vertex is zero
2.  for all vert ∈ V-{source} //set all other vertex distance to infinity
3.  do Dista[vert] ← ∞
4.  S← ∅  // S, the set of visited vertices is initially empty
5.  Q←V //(Q, the queue initially contain all vertices)
6.   while Q≠ ∅  do //while the queue is not empty
7.  u←minDista(Q,Dista)  // select the element of Q with the min.distance
8.  S←S∪{u} // add u to list of visited vertices
9.  for all vert ∈ neighbours[u]  // if new shortest path found)
10.   if Dista[vert] > Dista[u]+weight (u, vert) then
11. do Dista[vert]←Dista[u] +weight(u,vert) // set new value of shortest
    path
12. return Dista
```

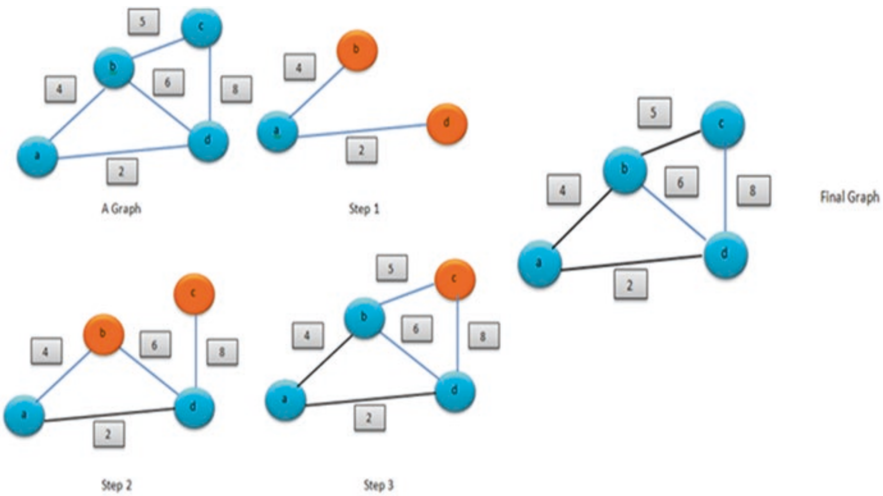


Fig. 3.14 Sample nodes for Dijkstra algorithm

3.4.3.4 Multi-Cloud Security in UAV Path Planning

Z. Zheng et al. [149] explored the planning of 3D trajectories for UAV in 3D cloud environments. A 3D-oriented trajectory algorithm uses point clouds directly to derive optimized trajectories for a drone in path palnning. This approach investigates unobstructed, low-cost, smooth, and dynamically achievable pathways by analyzing a point cloud of the target environment, using a modified RRT with the k-d tree. In perusing direction for the destination in a drone, security and privacy is a significant concern, and most of the authors are truck on the security odds in the drones. They primarily concentrated on how the communication of drones can be forbidden from the odds due to lack of safety. Talking for instance, Lin et al. [150] deliberated the requirements of security and privacy in UAV network architecture. Likewise, Javaid et al. [151] analyzed the security vulnerabilities and origins of loss of control and delivered the mitigation and deterrent solutions of UAVs.

Fu et al. [152] proposed an improved artificial potential field to empower multi-UAV collision resistant. For instance, Challita et al. [153] proposed ANN-based solution structures to address the wireless and security challenges that emerge with regard to UAVs such as distribution frameworks, transportation frameworks, and casting of digital data. These methodologies empower a gateway for the UAVs to function at par with wired devices even though its wireless. Kharchenko and Torianyk [154] used Internet of Drones for digital data securities and analysis of cyber security. Similarly, Bin Li et al. [155] suggested addition of 5G to give an optimized energy transmission. It gives null postponements, improved safety, and sophisticated usage of UAVs. Liang et al. [156] shared an idea of using blockchain to share data. Furthermore, this has shown incredible amount of promise as it enables transaction safety and user privacy.

3.5 Conclusion

This paper is written on an aim to see the sophistication of UAVs with the help of survey. It is a must to know that the path panning in UAVs is of three main divisions. They are representative, coordination, and non-coordination methods. Working ahead on the above methods, it has been found that UVAs could be more sophisticated when incorporated with the above methods by increasing the distance under range, networking in UAVs. Learning of safekeeping mechanisms of UAVs has been fascinating. UAVs on top of having more research papers based on them, they still got more bugs and glitches. Higher distance to cover, good networking, overtaking a cumbersome path, optimization of paths, user friendliness, developments, and cost-effectiveness are still a nightmare in UAVs, but thanks to loT systems and multi-cloud security algorithms for making it progress toward the development of UAVs. To hit on the specifics, vivid and vide number of papers are further needed to address the network treats in UAVs.

References

1. Beard, R. W., & D. K. (2005). Autonomous vehicle technologies for small fixed-wing UAVs. *Journal of Aerospace Computing, Information, and Communication, 2*, 92–108.
2. Candiago, S. A. (2015). Evaluating multispectral images and vegetation indices for precision farming applications from UAV images. *Remote Sensing, 7*(4), 4026–4047. https://doi.org/10.3390/rs70404026
3. Tsouros, D., Bibi, S., & Sarigiannidis, P. (2019). A review on UAV-based applications for precision agriculture. *Information, 10*(11), 1–26. https://doi.org/10.3390/info10110349. International Conference on Precision Agriculture.

4. Ozdemir, U. (2014, April). Design of a Commercial Hybrid VTOL UAV system. *Journal of Intelligent & Robotic Systems, 74*(12). https://doi.org/10.1007/s10846-013-9900-0
5. Hassanalian, M. A. (2017, April). Design, manufacturing, and flight testing of a fixed wing micro air vehicle with Zimmerman planform. *Meccanica, 52*, 1265–1282. https://doi.org/10.1007/s11012-016-0475-2
6. Singhal, G., Bansod, B., & Mathew, L. (2018). Unmanned aerial vehicle classification, applications and challenges: A review. Preprints 2018110601. https://doi.org/10.20944/preprints201811.0601.v1.
7. Michael, A. A., & Fenelon, T. F. (2010, February). Design of an active flapping wing mechanism and a micro aerial vehicle using a rotary actuator. *Mechanism and Machine Theory, 45*(2), 137–146. https://doi.org/10.1016/j.mechmachtheory.2009.01.007
8. Shyy, W., & Aono, H.-k. (2013). *An introduction to flapping wing aerodynamics* (p. 322). Cambridge University Press. 978-1-107-64035-1.
9. Gabriel Torres, T. J. (2015, May). Wing analysis of a flapping wing unmanned aerial vehicle using CFD. *International Journal of Advance Engineering and Research Development, 2*(5), 216–221. https://doi.org/10.21090/ijaerd.020530
10. Austin, R. (2011). *Unmanned aircraft systems: UAVS design, development and deployment* (Vol. 54). John Wiley &Sons. https://doi.org/10.1002/9780470664797
11. Haitao Xiang, L. T. (2011, February). Development of a low-cost agricultural remote sensing system based on an autonomous unmanned aerial vehicle (UAV). *Biosystems Engineering, 108*(2), 87–194. https://doi.org/10.1016/j.biosystemseng.2010.11.010
12. Russell Gayle, P. S. (2005). Path planning for deformable robots in complex environments. In *Robotics: Science and systems I, June 8-11, 2005* (pp. 1–8). Massachusetts Institute of Technology. https://doi.org/10.15607/RSS.2005.I.030
13. Fotouhi, A. Q.-R. (2019). Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. *IEEE Communications Surveys & Tutorials, 21*(4), 3417–3442. https://doi.org/10.1109/COMST.2019.2906228
14. Aggarwa, S., & l. K. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications, 149*, 270–299.
15. Santhi, K., & Zayaraz, G. (2015). Trade-off analysis of crosscutting functionalities using lazy counting-based splay tree in aspect oriented programming. *Research Journal of Applied Sciences, Engineering and Technology, 9*, 396–408. https://doi.org/10.19026/rjaset.9.1419
16. Cormen, T. H., & C. E. (2009). *Introduction to algorithms* (3rd ed.). The MIT Press.
17. Divkoti, M. R. R., & Nouri-Baygi, M. (2019). RCS: a fast path planning algorithm for Unmanned Aerial Vehicles. *Robotics*, 1–17. arxiv.org/abs/1904.12283
18. Santhi, K., & G. Z. (2014). Resolving aspect dependencies for composition of aspects. *Arabian Journal for Science and Engineering, 40*(2), 475–486. https://doi.org/10.1007/s13369-014-1454-3
19. Zouaoui, H., & F. S. (2019). Energy and quality aware multi-UAV flight path design through Q-learning algorithms. In *WWIC 2019: Wired/wireless internet communications* (Vol. 11618, pp. 246–257). Springer. https://doi.org/10.1007/978-3-030-30523-9_20
20. Bellingham, J., & Tillerson, M. (2003). Multi-task allocation and path planning for cooperating UAVs. Cooperative control: Models, applications and algorithms (pp. 23–41). https://doi.org/10.1007/978-1-4757-3758-5_2.
21. Xixia Sun, C. C. (2019, December)). A cooperative target search method based on intelligent water drops algorithm. *Computers & Electrical Engineering, 80*, 1–14. https://doi.org/10.1016/j.compeleceng.2019.106494
22. Bai, W., Wu, X., Xie, Y., Wang, Y., Zhao, H., Chen, K., … Hao, Y. (2018). A cooperative route planning method for multi-UAVs based-on the fusion of artificial potential field and B-spline interpolation. In *37th Chinese control conference (CCC)* (pp. 6733–6738). IEEE. https://doi.org/10.23919/ChiCC.2018.8483665
23. Geraerts, R. (2010). Planning short paths with clearance using explicit corridors. IEEE international conference on robotics and automation (pp. 1997–2004). Anchorage, AK. https://doi.org/10.1109/ROBOT.2010.5509263.

24. Biaz, R. C. (2013). UAV collision avoidance using RRT* and LOS maximization. #CSSE12 - 03.
25. Mademlis, I., & Mygdalis, V. (2018). Challenges in autonomous UAV cinematography: An overview. In *IEEE international conference on multimedia and expo (ICME)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICME.2018.8486586
26. MarinaTorresa, D. A. (2016). Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. Expert systems with applications (pp. 441–451). https://doi.org/10.1016/j.eswa.2016.02.007.
27. Balampanis, F. M. (2017, April). Area partition for coastal regions with multiple UAS. *Journal of Intelligent & Robotic Systems, 88*, 751–766. https://doi.org/10.1007/s10846-017-0559-9
28. Acevedo, J. A. (2013, April). Cooperative large area surveillance with a team of aerial Mobile robots for long endurance missions. *Journal of Intelligent & Robotic Systems, 70*, 329–345. https://doi.org/10.1007/s10846-012-9716-3
29. Zagradjanin, N., Pamucar, D., & Jovanovic, K. (2019). Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method. *Symmetry, 11*(10), 1–15. https://doi.org/10.3390/sym11101241
30. Mustafa Hamurcu, T. E. (2020). Selection of unmanned aerial vehicles by using multi-criteria decision-making for defence. *Journal of Mathematics, 2020*, 1–11. https://doi.org/10.1155/2020/4308756
31. Bijolin Edwin, E., & RoshniThanka, M. (2019). An internet of drone (IOD) based data analytics in cloud for emergency services. *International Journal of Recent Technology and Engineering (IJRTE), 7*(5S2), 1–5. Retrieved from https://www.ijrte.org/wp-content/uploads/papers/v7i5s2/ES2044017519.pdf.
32. Ryan DuToit, M. L. (n.d.). UAV collision avoidance using RRT* and LOS maximization. Technical report #CSSE12 - 03.
33. Kwangjin Yang, S. S. (2008). 3D smooth path planning for a UAV in cluttered natural environments. In *RSJ international conference on intelligent robots and systems*. IEEE. https://doi.org/10.1109/IROS.2008.4650637
34. Alex Nash, S. K. (2010). Lazy Theta*: Any-angle path planning and path length analysis in 3D. Twenty-Fourth AAAI Conference on Artificial Intelligence. AAAI Publications.
35. Hrabar, S. (2008). *3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs* (pp. 807–814). IEEE/RSJ International Conference on Intelligent Robots and Systems. https://doi.org/10.1109/IROS.2008.4650775
36. Sven Koenig, M. L. (2005). Fast replanning for navigation in unknown terrain. *IEEE Transactions on Robotics, 21*(3), 354–363. https://doi.org/10.1109/TRO.2004.838026
37. Franklin Samaniego, J. S.-N. (2017). UAV motion planning and obstacle avoidance based on adaptive 3D cell decomposition: Continuous space vs discrete space. (pp. 1–6). Salinas, Ecuador: IEEE second Ecuador technical chapters meeting (ETCM). https://doi.org/10.1109/ETCM.2017.8247533.
38. Choset, H. C. (2001). Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence, 31*, 113–126. https://doi.org/10.1023/A:1016639210559
39. Elfes, A. (June 1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation, 3*(3), 249–265. https://doi.org/10.1109/JRA.1987.1087096
40. Moravec, H. P., & Elfes, A. (1985). High resolution maps from wide angle sonar. International Conference on Robotics and Automation (Vol. 2, pp. 116–121). IEEE. https://doi.org/10.1109/ROBOT.1985.1087316.
41. Enric Galceran, M. C. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems, 61*(12), 1258–1276. https://doi.org/10.1016/j.robot.2013.09.004
42. Nourbakhsh, R. S. (2004). *Introduction to autonomous Mobile robots*. The MIT Press.
43. Lim, M. K. (2017). High performance and fast object detection in road environments. In *Seventh international conference on image processing theory, tools and applications (IPTA)* (pp. 1–6). IEEE. https://doi.org/10.1109/IPTA.2017.8310148

44. Zammit, C., & van Kampen, E. J. (2020). Comparison of A* and RRT in real–time 3D path planning of UAVs (pp. 1–25). Orlando, FL: American Institute of Aeronautics and Astronautics Inc. (AIAA). https://doi.org/10.2514/6.2020-0861.
45. Zhuang Wang, J. C. (2018, November). Probabilistic roadmap method for path-planning in radioactive environment of nuclear facilities. *Progress in Nuclear Energy, 109*, 113120. https://doi.org/10.1016/j.pnucene.2018.08.006j
46. Mansard, N., & del Prete, A. (2018). Using a memory of motion to efficiently warm-start a nonlinear predictive controller. In *International conference on robotics and automation (ICRA)* (pp. 2986–2993). IEEE. https://doi.org/10.1109/ICRA.2018.8463154
47. Lavalle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning.
48. Zu, W., & Fan, G. (2018). Multi-UAVs cooperative path planning method based on improved RRT algorithm. In *IEEE international conference on mechatronics and automation(ICMA)* (pp. 1563–1567). IEEE. https://doi.org/10.1109/ICMA.2018.8484400
49. Sun, Q., & Li, M. (2018). UAV path planning based on improved rapidly-exploring random tree. In *Chinese control and decision conference (CCDC)* (pp. 6420–6424). IEEE. https://doi.org/10.1109/CCDC.2018.8408258
50. Adiyatov, O., & Sultanov, K. (2017). Sparse tree heuristics for RRT* family motion planners. In *IEEE international conference on advanced intelligent mechatronics (AIM)* (pp. 1447–1452). IEEE. https://doi.org/10.1109/AIM.2017.8014222
51. Noreen, I., Khan, A., Asghar, K., & Habib, Z. (2019). A path-planning performance comparison of RRT*-AB with MEA* in a 2-dimensional environment. *Symmetry, 11*(7), 945. https://doi.org/10.3390/sym11070945
52. HanTong, W. W. (2012). Path planning of UAV based on Voronoi diagram and DPSO. *Procedia Engineering, 29*, 4198–4203. https://doi.org/10.1016/j.proeng.2012.01.643
53. Mengxiao Song, Z. J. (2018). Mosaicking UAV orthoimages using bounded Voronoi diagrams and watersheds. *International Journal of Remote Sensing, 39*(15–16), 4960–4979. https://doi.org/10.1080/01431161.2017.1350309
54. Xin Feng, A. T. (2018). Allocation using a heterogeneous space Voronoi diagram. *Journal of Geographical Systems*, 207–226. https://doi.org/10.1007/s10109-018-0274-5
55. Chen, X., Li, G. Y., & Chen, X. M. (2017). Path planning and cooperative control for multiple UAVs based on consistency theory and Voronoi diagram. 29th Chinese control and decision conference (CCDC), (pp. 881–886). Chongqing. https://doi.org/10.1109/CCDC.2017.7978644.
56. Oommen, B., Iyengar, S., Rao, N., & Kashyap, R. (1987, December). Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case. *IEEE Journal on Robotics and Automation, 3*(6), 672–681. https://doi.org/10.1109/JRA.1987.1087133
57. Rao, N. (1989, June). Algorithmic framework for learned robot navigation in unknown terrains. *Computer, 22*(6), 37–43. https://doi.org/10.1109/2.30719
58. Egerstedt, D. W. (2006). Oriented visibility graphs: Low-complexity planning in real-time environments. In *Proceedings 2006 IEEE international conference on robotics and automation, ICRA* (pp. 2354–2359). IEEE. https://doi.org/10.1109/ROBOT.2006.1642054
59. Huang, S., & Teo, R. S. (2019). Computationally efficient visibility graph-based generation of 3D shortest collision-free path among polyhedral obstacles for unmanned aerial vehicles. In *International conference on unmanned aircraft systems (ICUAS)* (pp. 1218–1223). IEEE. https://doi.org/10.1109/ICUAS.2019.8798322
60. Budiyanto, A., Cahyadi, A., Adji, T. B., & Wahyunggoro, O. (2015). UAV obstacle avoidance using potential field under dynamic environment. In *International conference on control, electronics, renewable energy and communications (ICCEREC)* (pp. 187–192). IEEE. https://doi.org/10.1109/ICCEREC.2015.7337041
61. Chen, S., & Yang, Z. (2017). An improved artificial potential field based path planning algorithm for unmanned aerial vehicle in dynamic environments. In *International conference on security, pattern analysis, and cybernetics (SPAC)* (pp. 591–596). IEEE. https://doi.org/10.1109/SPAC.2017.8304346

62. Abeywickrama, H. V., Jayawickrama, B. A., He, Y., & Dutkiewicz, E. (2018). Potential field based inter-UAV collision avoidance using virtual target relocation. In *IEEE 87th vehicular technology conference (VTC Spring)* (pp. 1–5). IEEE. https://doi.org/10.1109/VTCSpring.2018.8417773.D. Zhang, Y. X. (2018). An improved path planning algorithm for unmanned aerial vehicle based on RRT-connect. 37th Chinese control conference (CCC) (pp. 4854–4858). Wuhan: IEEE. doi:10.23919/ChiCC.2018.8483405.
63. Park, S., & Shin, C. (2018). DroneNetX: Network reconstruction through connectivity probing and relay deployment by multiple UAVs in ad hoc networks. *IEEE Transactions on Vehicular Technology, 67*(11), 11192–11207. https://doi.org/10.1109/TVT.2018.2870397
64. Carpin, A. K. (2006). Multirobot cooperation for surveillance of multiple moving targets - a new behavioral approach. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (pp. 1311–1316). IEEE. https://doi.org/10.1109/ROBOT.2006.1641890
65. Da Silva Arantes, J., & Arantes, M. D. S. (2017). Evaluating hardware platforms and path re-planning strategies for the UAV emergency landing problem. In *IEEE 29th international conference on tools with artificial intelligence (ICTAI)* (pp. 937–944). IEEE. https://doi.org/10.1109/ICTAI.2017.00144
66. Li, B., & Patankar, S. (2018). Planning large-scale search and rescue using team of UAVs and charging stations. In *2018 IEEE international symposium on safety, security, and rescue robotics (SSRR)* (pp. 1–8). IEEE. https://doi.org/10.1109/SSRR.2018.8468631
67. Zhou, Z., Feng, J., Gu, B., Ai, B., Mumtaz, S., Rodriguez, J., & Gui, M. (2018). When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning. *IEEE Transactions on Communications, 66*, 5526–5538. https://doi.org/10.1109/TCOMM.2018.2857461
68. Sharma, H., & T. S. (2017). An efficient backtracking-based approach to turn-constrained path planning for aerial mobile robots. In *European conference on Mobile robots (ECMR)* (pp. 1–8). IEEE. https://doi.org/10.1109/ECMR.2017.8098712
69. Farid, G., & Mo, H. (2018). Computationally efficient algorithm to generate a waypoints-based trajectory for a quadrotor UAV. In *Chinese control and decision conference(CCDC)* (p. 44144419). IEEE. https://doi.org/10.1109/CCDC.2018.8407894
70. Huang, D., & Zhao, D.. (2017). A new method of the shortest path planning for unmanned aerial vehicles. 2017 6th Data Driven Control and Learning Systems (DDCLS) (pp. 599–605). Chongqing, China. doi:https://doi.org/10.1109/DDCLS.2017.8068140
71. Wang, J., Chen, W.-B., & Temu, V. (2018). Multi-vehicle motion planning for search and tracking. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)* (pp. 352–355). IEEE. https://doi.org/10.1109/MIPR.2018.00078
72. Perazzo, P., & Sorbelli, F. B. (2017, September). Drone path planning for secure positioning and secure position verification. *IEEE Transactions on Mobile Computing, 16*(9), 2478–2493. https://doi.org/10.1109/TMC.2016.2627552
73. Bit-Monnot, R. B.-R. (2018). A local search approach to observation. International conference on automated planning and scheduling. Delft, Netherlands. Retrieved from https://hal.archives-ouvertes.fr/hal-01730655.
74. Kurdi, M. M., & Dadykin, A. K. (2018). Proposed system of artificial Neural Network for positioning and navigation of UAV-UGV. 2018 Electric electronics, computer science, Biomedical engineerings' meeting (EBBT) (p. 16). doi:https://doi.org/10.1109/EBBT.2018.8391459.
75. Zhang, Y., Zhang, Y., Liu, Z., Yu, Z., & Qu, Y. (2018). Line-of-sight path following control on UAV with sideslip estimation and compensation. In *37th Chinese control conference (CCC)* (pp. 4711–4716). IEEE. https://doi.org/10.23919/ChiCC.2018.8483606
76. Hamid Shiri, J. P. (2019). Remote UAV online path planning via neural network based opportunistic control. Networking and internet architecture (pp. 1–9). Retrieved from https://arxiv.org/abs/1910.04969.
77. Liang, H. J. (2018). Online path planning of autonomous UAVs for bearing-only standoff multi-target following in threat environment. *IEEE Access, 6*, 22531–22544. https://doi.org/10.1109/ACCESS.2018.2824849

78. Kang, M., & Liu, Y. (2017). A threat modeling method based on kalman filter for UAV path planning. 29th Chinese control and decision conference (CCDC) (pp. 3823–3828). Chongqing, China. https://doi.org/10.1109/CCDC.2017.7979170.
79. Wu, K., & Xi, T. (2017). Real-time three-dimensional smooth path planning for unmanned aerial vehicles in completely unknown cluttered environments. TENCON 2017–2017 IEEE region 10 conference, Penang, 2017 (pp. 2017–2022). Penang: IEEE. https://doi.org/10.1109/TENCON.2017.8228192.
80. Yoo, Q. Y. (2018). Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms. *IEEE Access, 6*, 13671–13684. https://doi.org/10.1109/ACCESS.2018.2812896
81. Marija Popović, T. V.-C. (2020). An informative path planning framework for UAV-based terrain monitoring. *Autonomous Robots, 44*, 889–911. https://doi.org/10.1007/s10514-020-09903-2
82. Faigl, J., & Váňa, P. (2018, April)). Surveillance planning with Bézier curves. *IEEE Robotics and Automation Letters, 3*(2), 750–757. https://doi.org/10.1109/LRA.2018.2789844
83. Farmani, N., & Sun, L. (2017). A scalable multitarget tracking system for cooperative unmanned aerial vehicles. *IEEE Transactions on Aerospace and Electronic Systems, 53*(4), 1947–1961. https://doi.org/10.1109/TAES.2017.2677746
84. Tartaglione, G. (2018). Obstacle avoidance via landmark clustering in a path-planning algorithm. Annual American control conference (ACC) (pp. 2776–2781). https://doi.org/10.23919/ACC.2018.8430891.
85. Sascha Lange, M. R. (2012). Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 international joint conference on neural networks (IJCNN)* (p. 18). IEEE. https://doi.org/10.1109/IJCNN.2012.6252823
86. Chao Yan, X. X. (2020). Towards real-time path planning through deep reinforcement learning for a UAV in dynamic environments. *Journal of Intelligent & Robotic Systems, 98*, 297–309.
87. Mnih, V. K. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529–533. https://doi.org/10.1038/nature14236
88. Luan, S., & Yang, Y. (2017). 3D G-learning in UAVs. In *12th IEEE conference on industrial electronics and applications (ICIEA)* (pp. 953–957). IEEE. https://doi.org/10.1109/ICIEA.2017.8282976
89. Zhang, T., & Huo, X. (2018). Hybrid path planning of A quadrotor UAV based on Q-learning algorithm. In *2018 37th Chinese control conference (CCC)* (pp. 5415–5419). IEEE. https://doi.org/10.23919/ChiCC.2018.8482604
90. Yijing, Z., & Zheng, Z. (2017). Q learning algorithm based UAV path learning and obstacle avoidance approach. In *2017 36th Chinese control conference (CCC)* (pp. 3397–3402). IEEE. https://doi.org/10.23919/ChiCC.2017.8027884
91. Bouhamed, O. G. (2020). Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach. IEEE international symposium on circuits and systems (ISCAS'200) (pp. 1–5). Seville, Spain: arXiv preprint. Doi:arXiv:2003.10923.
92. Stephen Boyd, L. V. (2009). Convex optimization. New York: United States of America by Cambridge University Press. https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf.
93. Yong Wang, K. A. (2018). Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Systems with Applications, 104*, 244–260. https://doi.org/10.1016/j.eswa.2018.03.018
94. Yin, C., & Xiao, Z. (2018, April). Offline and online search: UAV multiobjective path planning under dynamic urban environment. *IEEE Internet of Things Journal, 5*(2), 546–558. https://doi.org/10.1109/JIOT.2017.2717078
95. Luo, Z., & Liu, Z. (2018). The mathematical modeling of the two-echelon ground vehicle and its mounted unmanned aerial vehicle cooperated routing problem. 2018 IEEE intelligent vehicles symposium (IV), (pp. 1163–1170). Changshu. doi:https://doi.org/10.1109/IVS.2018.8500391.

96. Angley, D. R. (2019, January). Search for targets in a risky environment using multi-objective optimisation. *IET Radar, Sonar & Navigation, 13*(1), 123–127. https://doi.org/10.1049/iet-rsn.2018.5184
97. Koohifar, F., & Guvenc, I. (2018). Autonomous tracking of intermittent RF source using a UAV swarm. *IEEE Access, 6*, 15884–15897. https://doi.org/10.1109/ACCESS.2018.2810599
98. Le, N. T. (2018). Joint resource allocation, computation offloading, and path planning for UAV based hierarchical fog-cloud mobile systems. IEEE seventh international conference on communications and electronics (ICCE) (pp. 373–378). Hue. https://doi.org/10.1109/CCE.2018.8465572.
99. Zeng, Y., & Xiu, X. (2018, April). Trajectory design for completion time minimization in UAV-enabled multicasting. *EEE Transactions on Wireless Communications, 17*(4), 2233–2246. https://doi.org/10.1109/TWC.2018.2790401
100. Zeng, Y., & Xu, J. (2019, April). Energy minimization for wireless communication with rotary-wing UAV. *IEEE Transactions on Wireless Communications, 18*(4), 2329–2345. https://doi.org/10.1109/TWC.2019.2902559
101. Jeong, S., & O. S. (2018, March). Mobile edge computing via a UAV-mounted cloudlet: Optimization of Bit allocation and path planning. *IEEE Transactions on Vehicular Technology, 67*(3), 2049–2063. https://doi.org/10.1109/TVT.2017.2706308
102. Yu, J. L. (2017, June). Optimal path planning of solar-powered UAV using gravitational potential energy. *IEEE Transactions on Aerospace and Electronic Systems, 53*(3), 1442–1451. https://doi.org/10.1109/TAES.2017.2671522
103. Zamani, A., & Galloway, J. D. (2019). Feedback motion planning of legged robots by composing orbital Lyapunov functions using rapidly-exploring random trees. 2019 international conference on robotics and automation (ICRA) (pp. 1410–1416). https://doi.org/10.1109/ICRA.2019.8793578.
104. Elhoseny, M. (2017, August). Bezier curve based path planning in a dynamic field using modified genetic algorithm. *Journal of Computational Science*, 1–41. doi:https://doi.org/10.1016/j.jocs.2017.08.004
105. Miller, L. M. (2015). Ergodic exploration of distributed information. *IEEE Transactions on Robotics, 32*(1), 36–52. https://doi.org/10.1109/TRO.2015.2500441
106. Mathew, N., & Smith, S. L. (2015). Multirobot rendezvous planning for recharging in persistent tasks. *IEEE Transactions on Robotics, 31*(1), 128142. https://doi.org/10.1109/TRO.2014.2380593
107. De Waen, J., & Dinh, H. T. (2017). *Scalable multirotor UAV trajectory planning using mixed integer linear programming* (pp. 1–6). IEEE. https://doi.org/10.1109/ECMR.2017.8098706
108. Luo, X. (2017). Optimal path planning for UAV based inspection system of large-scale photovoltaic farm. In *2017 Chinese automation congress (CAC)* (pp. 4495–4500). IEEE. https://doi.org/10.1109/CAC.2017.8243572
109. Zhang, J., & Yan, J. (2015). Design and information architectures for an unmanned aerial vehicle cooperative formation tracking controller. *IEEE Access, 6*, 45821–45833. https://doi.org/10.1109/TRO.2015.2500441
110. Darbari, V., & Gupta, S. (2017). Dynamic motion planning for aerial surveillance on a fixed-wing UAV. 2017 international conference on unmanned aircraft systems (ICUAS), (pp. 488–497). Miami, FL. https://doi.org/10.1109/ICUAS.2017.7991463.
111. Yu, H., & Meier, K. (2015, April). Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles. *IEEE/ASME Transactions on Mechatronics, 20*(2), 541–552. https://doi.org/10.1109/TMECH.2014.2301459
112. Eaton, C. M., & Chong, E. K. P. (2017). Robust UAV path planning using POMDP with limited FOV sensor. 2017 IEEE conference on control technology and applications (CCTA) (pp. 1530–1535). Kohala Coast, Hawai'i, USA.
113. Yang, Q., & Zhang, J. (2018). Path planning for unmanned aerial vehicle passive detection under the framework of partially observable markov decision process. In *2018 Chinese control and decision conference (CCDC)* (pp. 3896–3903). IEEE. https://doi.org/10.1109/CCDC.2018.8407800

114. Alessandretti, A. (2017). A planar path-following model predictive controller for fixed-wing unmanned aerial vehicles. 2017 11th international workshop on robot motion and control (RoMoCo) (pp. 59–64). IEEE. https://doi.org/10.1109/RoMoCo.2017.8003893.
115. Sun, B., & Hu, C.. (2018). Trajectory planning of quadrotor UAV with suspended payload based on predictive control. 2018 37th Chinese control conference (CCC) (pp. 10049–10054). Wuhan. https://doi.org/10.23919/ChiCC.2018.8484159.
116. Yel, E., & Lin, T. X. (2018). Self-triggered adaptive planning and scheduling of UAV operations. 2018 IEEE international conference on robotics and automation (ICRA) (pp. 7518–7524). Brisbane, QLD. https://doi.org/10.1109/ICRA.2018.8463205.
117. Hausman, K., & J. P. (2017, July)). Observability-aware trajectory optimization for self-calibration with application to UAVs. *IEEE Robotics and Automation Letters, 2*(3), 1770–1777. https://doi.org/10.1109/LRA.2017.2647799
118. Li, J., & Li, X. (2018). Multi-UAV cooperative coverage path planning in plateau and mountain environment. In *2018 33rd youth academic annual conference of Chinese association of automation (YAC)* (pp. 820–824). IEEE. https://doi.org/10.1109/YAC.2018.8406484
119. Moustris, P. K. (2019, Jan). Terrain following for fixed-wing unmanned aerial vehicles using feedback equivalence. *IEEE Control Systems Letters, 3*(1), 150–155. https://doi.org/10.1109/LCSYS.2018.2854239
120. Tian, G., & Zhang, L. (2018). Real-time dynamic track planning of multi-UAV formation based on improved artificial bee colony algorithm. 2018 37th Chinese control conference (CCC) (pp. 10055–10060). Wuhan. https://doi.org/10.23919/ChiCC.2018.8482622.
121. Li, Z. C. (2018). Improved GASA algorithm for mutation strategy UAV path planning. In *2018 10th international conference on communication software and networks (ICCSN)* (pp. 506–510). IEEE. https://doi.org/10.1109/ICCSN.2018.8488319
122. Popović, M., & Hitz, G. (2017). Online informative path planning for active classification using UAVs. In *2017 IEEE international conference on robotics and automation (ICRA)* (pp. 5753–5758). Singapore. https://doi.org/10.1109/ICRA.2017.7989676
123. Mostafa, S. A., & Mustapha, A. (2018). A real-time autonomous flight navigation trajectory assessment for unmanned aerial vehicles. 2018 international symposium on agent, multi-agent systems and robotics (ISAMSR) (pp. 1–6). Putrajaya. https://doi.org/10.1109/ISAMSR.2018.8540544.
124. Ji, X., Hua, Q., Li, C., & Tang, J. (2017). 2-OptACO: An improvement of ant Colony optimization for UAV path in disaster rescue. 2017 international conference on networking and network applications (NaNA) (pp. 225–231). Kathmandu. https://doi.org/10.1109/NaNA.2017.16.
125. Yang, S., & Wei, N. (2017). Real-time optimal path planning and wind estimation using Gaussian process regression for precision airdrop. In *2017 American control conference (ACC)* (pp. 2582–2587). IEEE. https://doi.org/10.23919/ACC.2017.7963341
126. Sun, B. Z. (2013). A novel tracking control approach for unmanned underwater vehicles based on bio-inspired neurodynamics. *Journal of Marine Science and Technology, 18*, 63–74. https://doi.org/10.1007/s00773-012-0188-8
127. Fazio, P., & Rango, F. D. (2019). Scalable and ligthway bio-inspired coordination protocol for FANET in precision agriculture applications. *Computers & Electrical Engineering, 74*, 305–318. https://doi.org/10.1016/j.compeleceng.2019.01.018
128. Hasircioglu, I., & Topcuoglu, H. R. (2008). 3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms (pp. 1499–1506). https://doi.org/10.1145/1389095.1389386.
129. Pu, H. Z. (2011). Modified shuffled frog leaping algorithm for optimization of UAV flight controller. *International Journal of Intelligent Computing and Cybernetics, 4*(1), 25–39. https://doi.org/10.1108/17563781111115778
130. Jia Song, M. Z. (2019). The high-speed rotorcraft unmanned aerial vehicle path planning based on the biogeography-based optimization algorithm. *Advances in Mechanical Engineering, 11*, 1–12. https://doi.org/10.1177/1687814019847863

131. Paranjape, A. A., & S. C. (2018). Robotic herding of a flock of birds using an unmanned aerial vehicle. *IEEE Transactions on Robotics, 34*(4), 901–915. https://doi.org/10.1109/TRO.2018.2853610
132. Sun, Z., & Wu, J. (2016). Path planning for GEO-UAV Bistatic SAR using constrained adaptive multiobjective differential evolution. *IEEE Transactions on Geoscience and Remote Sensing, 54*(11), 6444–6457. https://doi.org/10.1109/TGRS.2016.2585184
133. Liu, Y., & Zhuang, Y. (2018). Binocular vision-based autonomous path planning for UAVs in unknown outdoor scenes. In *018 eighth international conference on information science and technology (ICIST)* (pp. 492–498). IEEE. https://doi.org/10.1109/ICIST.2018.8426133
134. Ren, S., & Chen, Y. (2018). Path planning for the marsupial double-UAVs system in air-ground collaborative application. 2018 37th Chinese Control Conference (CCC), (pp. 5420–5425). Wuhan. https://doi.org/10.23919/ChiCC.2018.8483087.
135. Kamel, A. T. (2019). Cooperative task assignment and trajectory planning of unmanned systems via HFLC and PSO. *World Scientific, 7*(2), 65–81. https://doi.org/10.1142/S2301385019500018
136. Shikai Shaoa, Y. P. (2020). Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Transactions, 97*, 415–430. https://doi.org/10.1016/j.isatra.2019.08.018
137. Chen, J., & Ye, F. (2017). Travelling salesman problem for UAV path planning with two parallel optimization algorithms. In *2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)* (pp. 832–837). IEEE. https://doi.org/10.1109/PIERS-FALL.2017.8293250
138. Chen, X., & Xu, R. (2017). Multi-objective route planning for UAV. In *2017 4th international conference on information science and control engineering (ICISCE)* (pp. 1023–1027). IEEE. https://doi.org/10.1109/ICISCE.2017.215
139. Razzaq, S., & C. X. (2018). Three-dimensional UAV routing with Deconfliction. *IEEE Access, 6*, 21536–21551. https://doi.org/10.1109/ACCESS.2018.2824558
140. Morita, T., & Oyama, K. (2018). Decision making support of UAV path planning for efficient sensing in radiation dose mapping. 2018 IEEE 42nd annual computer software and applications conference (COMPSAC) (pp. 333–338). Tokyo. https://doi.org/10.1109/COMPSAC.2018.00053.
141. Yang, J., & Xi, J. (2018). Multi-base multi-UAV cooperative patrol route planning novel method. 2018 33rd youth academic annual conference of chinese association of automation (YAC) (pp. 688–693). Nanjing. https://doi.org/10.1109/YAC.2018.8406460.
142. Cowlagi, R. D. (2017). Interactive sensing and path-planning with incremental 3D path repair for a quadrotor UAV in cluttered and partially known environments. In *2017 IEEE 56th annual conference on decision and control (CDC)* (pp. 933–938). IEEE. https://doi.org/10.1109/CDC.2017.8263778
143. Bogdanowicz, Z. R. (2017, December). Flying swarm of drones over Circulant digraph. *IEEE Transactions on Aerospace and Electronic Systems, 53*(6), 2662–2670. https://doi.org/10.1109/TAES.2017.2709858
144. Batsoyol, N., & Lee, H. J. (2018). Towards self-organizing UAV ad-hoc networks through collaborative sensing and deployment. In *2018 IEEE global communications conference (GLOBECOM)* (pp. 1–7). IEEE. https://doi.org/10.1109/GLOCOM.2018.8647985
145. Challita, U., & Saad, W. (2018). Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs. 2018 IEEE international conference on communications (ICC) (pp. 1–7). Kansas City, MO. https://doi.org/10.1109/ICC.2018.8422706.
146. Bouzid, Y., & Bestaoui, Y. (2017). Quadrotor-UAV optimal coverage path planning in cluttered environment with a limited onboard energy. 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 979–984). Vancouver, BC. https://doi.org/10.1109/IROS.2017.8202264.
147. Wu, X., & Xu, L. (2019). Biased sampling potentially guided intelligent bidirectional RRT* algorithm for UAV path planning in 3D environment. *Mathematical Problems in Engineering, 2019*, 1–12. Retrieved from https://doi.org/10.1155/2019/5157403.

148. Cabreira, T. M., & Franco, C. D. (2018, October). Energy-aware spiral coverage path planning for UAV photogrammetric applications. *IEEE Robotics and Automation Letters, 3*(4), 3662–3668. https://doi.org/10.1109/LRA.2018.2854967
149. Zheng, Z., & Bewley, T. R. (2020). Point cloud-based target-oriented 3D path planning for UAVs. International conference on unmanned aircraft systems (ICUAS) (pp. 790–798). https://doi.org/10.1109/ICUAS48674.2020.9213894.
150. Lin, C., & He, D. (2018, Jan). Security and privacy for the internet of drones: Challenges and solutions. *IEEE Communications Magazine, 56*(1), 64–69. https://doi.org/10.1109/MCOM.2017.1700390
151. Javaid, A. Y., & Sun, W. (2012). Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In *2012 IEEE conference on technologies for homeland security (HST)* (pp. 585–590). IEEE. https://doi.org/10.1109/THS.2012.6459914
152. Fu, Z., & Mao, Y. (2019). Secure multi-UAV collaborative task allocation. *IEEE Access, 7*, 35579–35587. https://doi.org/10.1109/ACCESS.2019.2902221
153. Challita, U., & Ferdowsi, A. (2019, February). Machine learning for wireless connectivity and security of cellular-connected UAVs. *IEEE Wireless Communications, 26*(1), 28–35. https://doi.org/10.1109/MWC.2018.1800155
154. Kharchenko, V. (2018). Cybersecurity of the internet of drones: Vulnerabilities analysis and IMECA based assessment. In *2018 IEEE 9th international conference on dependable systems, services and technologies (DESSERT)* (pp. 364–369). IEEE. https://doi.org/10.1109/DESSERT.2018.8409160
155. Li, B., & Fei, Z. (2019, October). Secure UAV communication networks over 5G. *IEEE Wireless Communications, 26*(5), 114–120. https://doi.org/10.1109/MWC.2019.1800458
156. Qiu, J., & Grace, D. (2020, January). Blockchain-based secure Spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: An Operator's perspective. *IEEE Internet of Things Journal, 7*(1), 451–466. https://doi.org/10.1109/JIOT.2019.2944213

Chapter 4
Estimation of Sharing Dependencies in Personal Storage Clouds Using Ensemble Learning Approaches

S. Poonkuntran and J. Manessa

4.1 Dynamic Provisioning of Cloud Resources in Multi-Cloud Environment

Cloud computing is an emerging trend in all kinds of industries starting from data collection to prepare detailed analysis and conclusions. It becomes a preference for industries in different ways. They include no need to invest in infrastructure, an infinite amount of resource capacity, and no need to wait for several months to build infrastructure to start the business. The cloud provides complete virtualization where user can hire their required resources at competitive prices. They need to pay only for the resource they are using. The users can acquire their resources dynamically, and the same can elastically be customized based on the changing needs at different times [1].

Many architectures have been proposed and practiced by companies. Multi-cloud is one particular architecture where the services will be availed from different vendors for single network architecture. It is different from the hybrid cloud that provides a heterogeneous environment composed of different infrastructure environments such as the public and private cloud. There are three main reasons for choosing a multi-cloud environment.

Choice: The cloud service vendors are chosen by the companies for their needs, and they have flexibility in choosing the vendors. It helps the companies to avoid vendor lock-in.

S. Poonkuntran (✉)
School of Computer Science and Engineering, VIT Bhopal University, Bhopal, India
e-mail: poonkuntran.s@vitbhopal.ac.in

J. Manessa
Freelance Trainer, Madurai, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_4

Disaster Avoidance: The multi-cloud environment provides 100% availability of resources for computation and storage if any loss happens due to human errors or disaster. It helps the companies to avoid downtime.

Compliance: The multi-cloud environments help the companies to achieve their goals for governance, risk management, and compliance regulations.

The dynamic and elasticity are important features of multi-cloud service for which users no need to do anything from their side. The service provider of cloud computing needs to take care provisioning of the resources to the different users based on their changing needs over different times. This cannot be simply done and needs to have a good mechanism to exactly predict the number of resources required to execute the computation, improve utilization of resources, and minimize the cost.

The resource provisioning includes the selection of appropriate hardware resources (Processor, RAM, Storage, Networking), deployment, OS requirements, creating a runtime environment, and providing necessary software and its management to ensure the guaranteed performance of the application as per service level agreement (SLA). Every service will be offered through an SLA executed between provider and consumer of the cloud.

The SLA will provide QoS parameters which include availability, reliability, response time, throughput, security, and performance. The provisioning of resources needs to be done without affecting the SLA and QoS. The resource provisioning will have four different styles. They are static, dynamic, static/dynamic, and user self-service. In static, the resource provisioning will be fixed irrespective of workload and requirements. It is suffered from underprovisioning and overprovisioning. In underprovisioning, few jobs will never be executed, since no resources are provisioned for them. In overprovisioning, few jobs will have more resources than required, and it becomes a waste. Both situations will lead to ineffective utilization of resources. The dynamic provisioning will allocate resources based on the demands, and it is not fixed. It is decided based on different parameters which include demand, events, and popularity. Here, the allocation of resources will be varied over time based on requirements. Predicting the exact requirement will be a challenge in dynamic provisioning. Sometimes, the prediction will not be possible due to frequent requirement changes. Hence, the third style static/dynamic is used. It is a hybrid style where static provisioning will be done in cases where dynamic provisioning is not feasible. The fourth style is user self-service where the user will decide the requirements and purchase the resources [1, 2].

The dynamic resource provisioning targeted response time, minimization of the cost of the services, maximization of company profit, fault tolerance, reducing SLA violation, and efficient power consumption [3]. The prediction of the pattern of resource consumption in cloud computing is very crucial and important in the dynamic provisioning of resources. It can only serve as a base for the provisioning. It raises the demand for the prediction of workload patterns in cloud computing. The workload parameters are different from one application to another running on the cloud.

The social networking applications which require websites to be run on the cloud need higher bandwidth and storage requirements. The scientific application which requires computing infrastructure to be run on cloud needs high computing powers. The e-Commerce applications require their application to be run on cloud needs current load parameters. Similarly, the different parameters will be used to measure workload parameters based on different applications.

Many companies are presently shifting their business to the multi-cloud to avail the benefits of elasticity and pay as you go cost molds. Such an environment will have several public clouds and private clouds based on the company's business needs. Such an environment demands storage solutions in the cloud. Thereby, the storage cloud becomes popular today [4–19].

4.1.1 Storage Cloud

The storage cloud provides storage solutions to the user where the user can store their data on the cloud through the Internet and access them being anywhere. It provides anytime anywhere access to the user's data. The cloud storage provider delivers your data on demand, just in time fashion and cheaper cost. It makes users avoid buying their own storage devices. The cloud storage vendors need to have a well scalable infrastructure to provide storage services to the users on-demand and pay as you go model. They need to manage their capacity, security, and durability to provide access to your data [4–19].

The storage cloud takes many advantages, and the major three advantages are listed below.

1. No need to invest in hardware infrastructures to create our storage facilities. Users need to avail of the entire storage solution on the cloud, and they can pay as per their usage. Usually, the two parameters will be used to measure the consumption of storage resources in the cloud. They are the amount of storage purchased and how frequently the storage is accessed for storage and retrieval. This model brings different cost metrics based on the data access frequency. If a particular data is created and it is not frequently accessed, this will come to the lesser cost models.
2. The storage cloud provides on-time deployment of resources where any additional resources can be purchased in time without worries about installation and deployment. It can quickly be added to your subscription. We can find additional resources dynamically on the go, no need to fix it in advance. However, it requires well prediction techniques to forecast future needs.
3. The cloud storage provides centralized data management where all the data is stored centrally in the cloud, and it can be easily accessed, shared, and migrated online. It can also be done anywhere and anytime.

Cloud storage supports different types of formats in which the data can be stored and retrieved. It includes object storage, file storage, and block storage. The object storage stores the objects that are metadata of the data to be stored. It allows us to build our storage solution from scratch. The Amazon Simple Storage Service (S3) is an example of object storage. It can be widely used to import, backup, and archive our data.

The second type is file storage where the application needs a file system through which it will share the files between users. Usually, file storage is linked to Network Attached Storage (NAS). File storage is widely used by enterprises and users to store their information as files and storing media files. The Amazon Elastic File System (EFS) is an example of file storage.

The third type is block storage where the application will have low latency and variable size storage for all the users connected to it. For example, the ERP systems will need individual storage to work with databases. Usually, this storage is facilitated through Storage Area Networks (SAN). The Amazon Elastic Block Store (EBS) is an example of block storage.

From the storage cloud revolution, personal cloud storage has come up now. Personal cloud storage provides cloud storage solutions to individuals for storing their files, photos, and videos. As in cloud storage, it enables individuals to access their files at any time, anywhere [4–19].

4.1.2 Dynamic Provisioning of Personal Storage Cloud

The dynamic provisioning of personal storage cloud includes volume creation, ensuring the availability of files in time, preparing the necessary number of copies of the same file for parallel processing, ensuring the security of the contents, protecting ownership rights, and measuring sharing dependency. The sharing dependency is a key parameter in the storage cloud by which how files are being shared among the number of users. This will be an important parameter for predicting the workload patterns in personal storage clouds. The majority of the user who uses the personal cloud accounts is mainly for sharing their contents with their dears and nears.

The sharing dependency helps the cloud service provider to estimate the resources and provisioning them for sharing. However, it is not easy to estimate the sharing dependency of the files on cloud storage. The pattern of sharing will change over time and requires intensive machine learning applications to predict them [4–19].

It also helps the companies to check the utilization of the spaces availed from the cloud and to estimate the future trends in storage requirements.

4.2 Machine Learning in Cloud Resource Provisioning

The elasticity is the main feature of cloud computing which automatically increases and shrinks the resources for the applications based on the workload. The prediction of the workload usually is done through resource indicators such as processor usages, storage usage, and traffic parameters. It is well suited for non-complex applications where the workload patterns are having periodical changes. However, it is very difficult to fix the indicator values, changes, and obtained thresholds.

When applications become complex, the indicators will be at a low level and limited. The changes in workload will be ad hoc, and it requires techniques apart from the resource indicators. In general, all the cloud services will be offered as best of service where no reservation of resources will be done by the vendors for the applications. Based on the demand raised by the applications, the resources are being provisioned. No guarantee is given by the vendor.

At the same time, cloud vendors can also provide advanced reservations for the applications. In this case, the SLA will have guaranteed QoS parameters to provide reservation. Such reservations need to be supported by the machine learning utilities to compute the current level of resource utilization and estimate future trends [20].

Machine learning is being applied in cloud resource provisioning, especially for pro-active management and auto-scaling. Many kinds of research have witnessed the role of machine learning in auto-scaling and the proactive management of clouds. VMware's Distributed Resource Scheduler (DRS) is proposed in [21] and uses an auto-scaling. A middleware called "Haizea" is introduced in [22] and offers reservation facility through leases. The Haizea can be called anytime and anywhere. The metric-based scaling technique called "Amazon Cloud Watch" is proposed in [23] that does auto-scaling on demand. No reservation is supported by the Amazon Cloud Watch. In [24], a system is proposed that reserves the resources initially and later adjusts based on the demands of the applications. In [25], the Aneka system is proposed that collects and releases the resources based on the completion time of requests. The auto-scaling is governed by the thresholds that must meet the requirements of completion of applications. In [26], a system is proposed for auto-scaling that uses past workload statistics to estimate future needs.

In [27], statistical machine learning was employed in single tier applications on the cloud to estimate the system performance parameters. The CPU and bandwidth utilization of VMs for single-tier applications were measured in Amazon EC2 using machine learning [28]. The reinforcement learning approach was used to find the best optimal server configurations in [29]. The same reinforcement learning was used to allocate the resources automatically in [30] for single-tier applications. The work in [31] uses non-linear regression techniques to learn the patterns of the performance of web applications hosted on the cloud, and a trained model is then used to predict future needs. The queuing networks were used in the research for analyzing the behaviors of each tier in multi-tier web applications [32]. The capacity of multi-tier web applications was estimated online using machine learning in [33]. The k-means clustering algorithm was used to model the dynamic workloads of multi-tier applications. This work uses queuing theory and service rates as base parameters for the clustering. Another work in [34] uses the learning models to do resource provisioning for homogeneous performances. In [35], machine learning is employed to automatically configuring web applications based on CPU utilization, number of requests, and network utilization. A rule-based learning model was used in [36] for identifying sudden changes in requirements.

From these works, it is concluded that workload volume is a key parameter that is being identified differently and used to estimate the future needs of the requirements. The machine learning techniques were good in predicting the workload patterns in the clouds, especially for dynamic provisioning. The machine learning models need to be designed based on the application for which the estimation is carried out [27–37].

The sharing dependency is a vital relation in identifying the workload volumes of storage clouds. This chapter takes NEC personal cloud data and models the sharing dependency using ensemble learning for the auto-provisioning of resources.

4.3 Prediction of Sharing Dependency Using Ensemble Learning

4.3.1 Ensemble Learning

Ensemble learning is a branch of machine learning where multiple model outputs are combined to yield improved performance. Bias and variance are two important factors in machine learning.

The bias is referring to the average difference between actual values and predicted values. The trained model bias is high; it will underperform and miss important details in the models. Hence, we need to have a lower bias. When models have become complex to train, the bias will reduce and it is up to some point. After that, the variance of the model will increase.

The variance is another factor that defines how the prediction is done on the same data different from each other. The high variance will overfit your data and the model will be failed. However, the bias and variance will be the tradeoff as shown in Fig. 4.1. To resolve this, ensemble learning is used.

The bias and variance will play an important role in machine learning. The low bias and low variance model will be good, and it will classify all the data into the respective classes. If the model has low bias and high variance, it will classify the data around the respective classes, not exactly on the classes, and each data will be far away from each other. If the model has high bias and low variance, it will miss the data and will not classify it into respective classes. However, all these data will be near to each other because of the low variance. If the model has high bias and high variance, it will miss all the data and will not classify into the respective classes. All these data will be far away from each other because of the high variance.

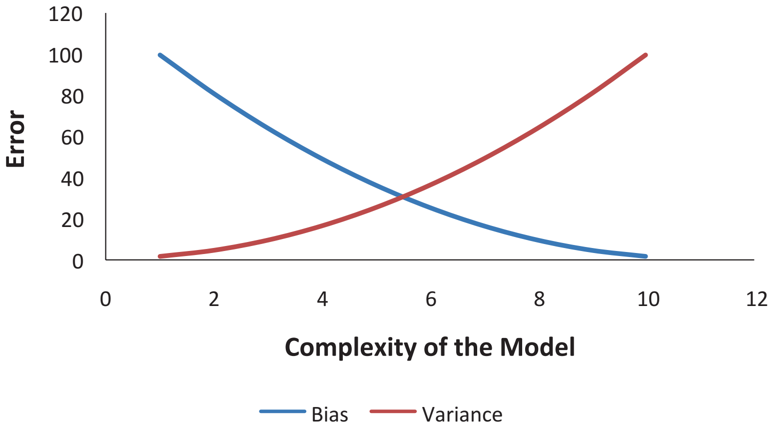


Fig. 4.1 Bias vs. variance in machine learning

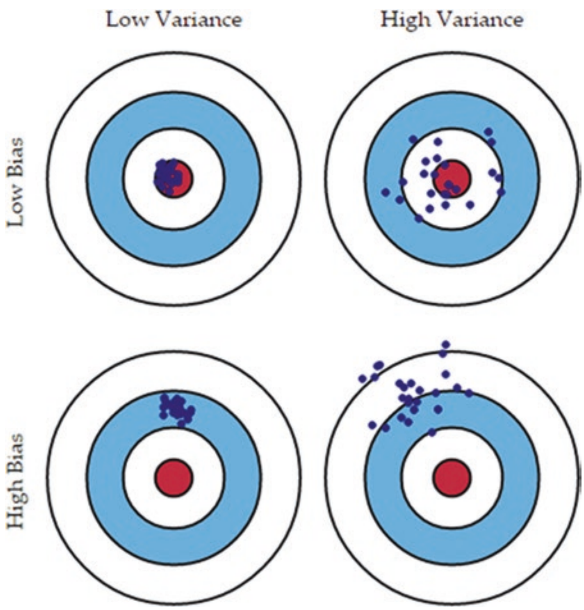


Fig. 4.2 The bulls eye diagram for bias vs. variance in machine learning. Source: http://scott.fortmann-roe.com/docs/BiasVariance.html

Hence, the main objective of any machine learning model is to reduce bias and variance to get the best model. It is shown in Fig. 4.2. The red color circle is a class to which the data points (blue colored dots) to be classified.

The ensemble learning can be applied in three different styles. They are bagging, boosting, and stacking. The bagging style will divide your input data sets into small, multiple datasets. The appropriate, different machine learning models will be chosen for every subset of the data and applied independently. The results of all the subsets are aggregated through an average to give the results of the original dataset.

The boosting style will not divide the datasets. It collects all the appropriate models and starts applying them one by one on the dataset. It adjusts the weight values of the model based on the previous results. Thereby, it reduces the bias and makes the model the best fit for the data.

The stacking is a different style where different model outputs are combined. The combined operation targets reducing bias or variance. Thereby, the best model is identified through stacking.

4.3.2 Dataset

The NEC Personal Cloud Data set is used in this chapter [38]. This dataset is an original dataset collected from the vendors with limitations. The limitation includes no location information is collected for privacy. All the data are collected as traces and it contains log information. The data are given by two trace files. The File Trace (CS_FileTraces.txt) contains details about the file, and Sharing Traces (CS_SharingTraces.txt) contains sharing details of the files. The data collected is real data from 7th March 2013 to 9th September 2015, and the total duration of the data is 2 years and 6 months. This data set contains two levels of information. They are storage and sharing interactions.

From the storage level, file traces were collected directly from the provider through SQL Server and Open Stack Swift. The file traces are log files containing file id, user id, file size in Bytes, and Account/Container ID in which the file is located. This information is used to analyze the file-related details at the storage layer. The statistics of the file traces are listed in Table 4.1.

From the sharing interaction, the sharing traces were collected. The sharing traces are log files that contain interaction among users and their sharing file information. The data is collected for 2 years and 6 months duration as mentioned above. This file contains 75,041 interaction details of file sharing. The fields include user 1, user 2, file id, folder ID, and account ID. User 1 is the owner of the file, and it is located in the folder that is linked to that user in the cloud. The user 2 is sharing the files of the user1. The statistics of the sharing interaction are listed in Table 4.2.

Table 4.1 CS_FileTraces – statistics

Total number records	76,554 file records
Total number of unique users	7721
Total number of unique accounts/containers	9215
Total number of unique files	74,723
Total number of unique file formats available in the dataset	418
Total number of unique file size present in the dataset	63,430

Table 4.2 CS_SharingTraces – statistics

Total number records	75,041 sharing interaction records
Total number of unique owners (User 1)	7015
Total number of unique users sharing the files (User 2)	8314

4.3.3 Preprocessing

The given datasets cannot be directly used for machine learning. To prepare the extracted data to be used for machine learning, preprocessing is carried out. All the details given in the file traces are categorical types except file size which is a numerical type. Similarly, the details in the sharing traces are also in categorical types. The preprocessing analyzes the dataset thoroughly. The file traces contain the details about each file whose owner details are given by the user id. The file sizes are given in bytes. The account/container id is the location id where the file is kept in the cloud. The sharing traces contain files whose owners are in user 1 and sharing users are in user 2. However, the sharing traces recorded each interaction separately. These interactions are raw and not labeled or grouped. The sharing interactions are in different styles which include a single file shared by many users, many files shared by a single user, a single file shared by a single user, and many files shared by a single user. To label the interactions, the number of files of each owner that is shared by others is calculated. The distribution of the number of files of each owner is illustrated in Fig. 4.3.

From Fig. 4.3, it is found that few owners do not have any files in their accounts that are being accessed by others. The number of files held by the owner is around 500 for the majority of the owners. Few of them are having a huge number of files. The minimum and the maximum number of files held by the owner are 0 and 9369, respectively.

Next, the file sizes are computed for each owner. The sharing traces contain only the file ID that is being shared. The size of each file is given in file traces. To calculate the file sizes, all the files shared by an owner are taken from sharing traces, whose size is referred from file traces, and the total file sizes are computed by sum. This is taken as a folder size since it contains many files. The folder sizes are computed in Bytes and converted into MegaBytes (MB). The distribution of folder size is as shown in Fig. 4.4. It is clearly shown that few owners do not have any files in their folder and it makes folder size zero. Many owner's folder size is around 5000 MB and a few of them having folder size above 5000 MB. The minimum and maximum folder sizes are 0 MB and 241830.1 MB, respectively.

Next, the frequency of folder access for each owner is computed. It is directly calculated from the sharing traces file by doing self-references of owners in the user1 field. The reason is that sharing traces contain interaction details of each file that is linked to an owner.

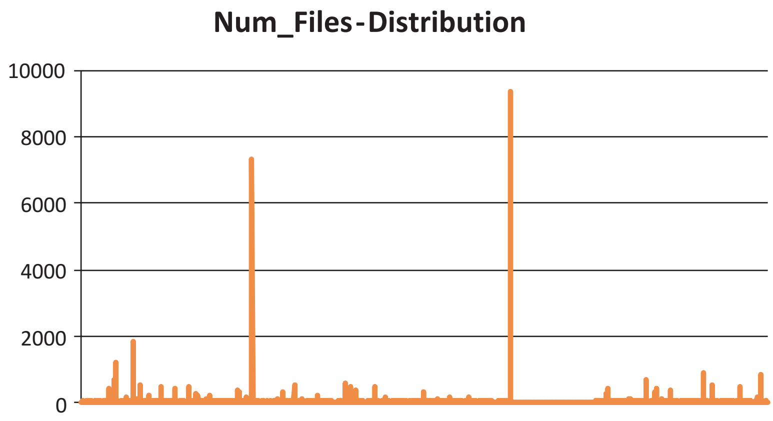


Fig. 4.3 The distribution of the number of files for each owner that is shared by others

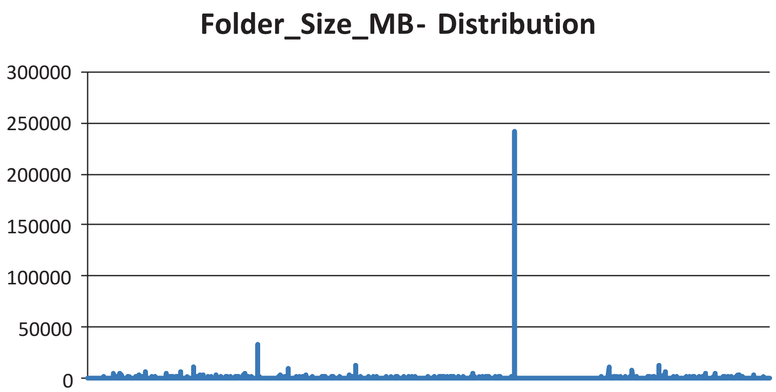


Fig. 4.4 The distribution of the folder size of each owner

The distribution of frequency access to the folder is shown in Fig. 4.5. It is clearly shown that the minimum frequency is zero. It means that the folder is not accessed by anyone. The maximum frequency is 242. It means that the folder is shared by 242 users. Thus the preprocessing extracted four different parameters from the data-set. The user ID is mainly for identification, and it is excluded from machine learning. The other three parameters number of files, folder size, and frequency access are taken for machine learning.

4.3.4 Ensemble Classifiers

The five different ensemble classifiers are used in the experiment. They are bagged tree, boosted tree, subspace discriminant, subspace kNN, and RUSboosted tree.

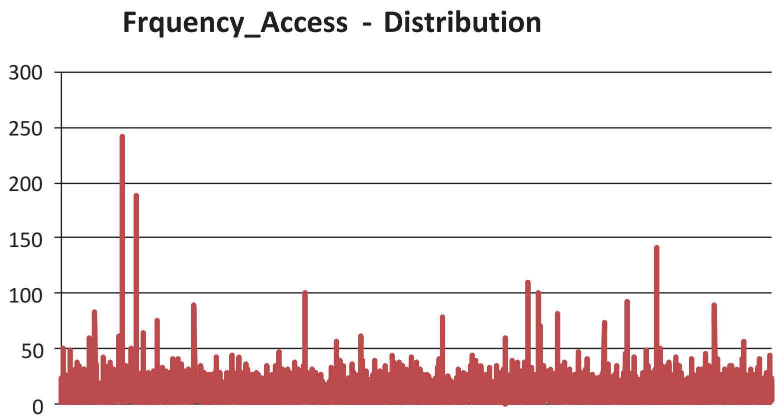


Fig. 4.5 The distribution of the frequency of access to the owner's folder

The bagged tree uses a random forest algorithm. It divides the given input into *n* number of parts using bootstrap sampling. Then, it constructs n number of classification trees and combines them into a single tree to yield the final classification [39–41].

The ADA boosting is used in the boosted tree which is one of the first boosting algorithms that combines multiple weak learner's outputs into single strong learners by using different weights. The ADA boost first divides the input into number splits and then applies a decision tree on them individually. A single split with a decision tree is called decision stumps. It analyzes all the stumps carefully and assigns higher weightage to the stumps which is difficult to classify and low weightage to the stumps which can classify well [42].

The subspace discrimination uses discriminant learners in subspace. The learners use Gaussian mixture models. The subspaces are generated using random subspace methods. The random subspace method uses three parameters. The first parameter is the actual dimension (D1) of the data (it means the number of columns in the given data) and next is the dimensions of the data (D2) to be used in the samples of each learner in the ensemble. The value of D2 is to be calculated statistically. The last parameter is the number of learners (N) used in the ensemble.

The random subspace method identifies the random subset of D2 variables from the D1 possible variables. Then, the methods iteratively train the week learners of discriminant analysis using these D2 variables, until there are N weak learners. The final prediction is carried out from the highest average of weak learners. Thus, the subspace discrimination works. Similarly, the subspace kNN works in subspace with nearest neighbors' leaners rather than discriminant analysis [42–47].

The RUSboosted tree uses ADA boosting algorithm in under-sampling space. This method is well suited for imbalanced data where few classes will have only a few members in the training data compared to other classes. The under-sampling is a key concept in this method where the number of members for each class in the training data is computed first and selects the minimum value. The chosen minimum value will be the sample size for each class to which the undersampling is done for the classes whose members are above the minimum. Finally, the boosting procedure will be done based on ADA Boost [42–47].

4.3.5 Data Cleaning

As in Sects. 3.3.2 and 3.3.3, the NEC data set is a log file that contains two sets of information for sharing dependency in the personal clouds. From this, we have extracted three vital pieces of information, namely, the number of files, folder size in MB, and frequency of access of the files for each owner. This is the data to be used in machine learning. This data is collected for 7015 owners in the NEC personal cloud. Thereby, the data set contains 7015 records and the dimension is three. The number of files and folder size is chosen as predictors, and the frequency of access is taken as a response.

First, the outliers analysis is carried out for from the variables through the box plot. By reviewing the box plot of the number of files as shown in Fig. 4.6, it is clear that the majority of the values are below 1000 and few values are above 1000. The number of the file contains zero values for 897 records. It means that 897 owners have accounts with no files in them. The median of the number of files is six. By analyzing the box plot for the number of files, it is found that the data is not uniformly distributed.

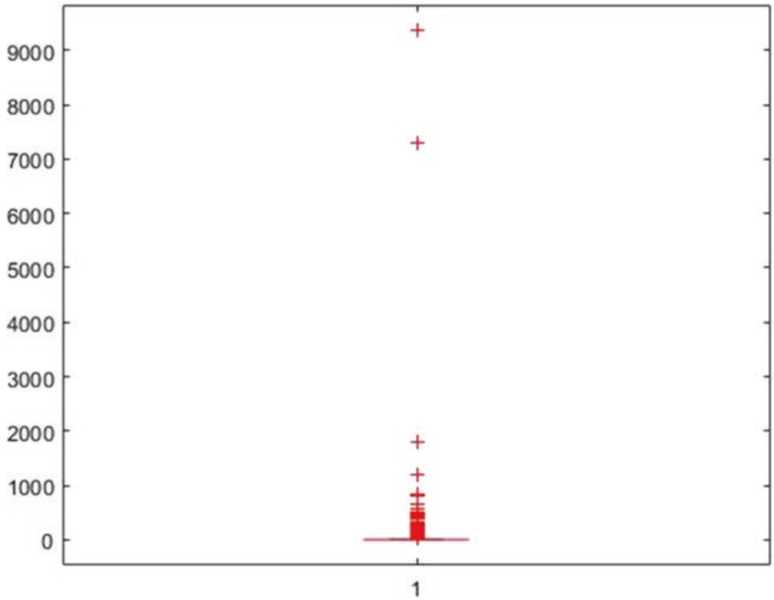


Fig. 4.6 The box plot for number of files

The box plot for the folder size is shown in Fig. 4.7. The minimum folder size is 0 MB and the maximum is 241,830 MB. The accounts which are not having any files will have a 0 MB folder size. The median value is 166 MB. The folder size is also not uniformly distributed. The response variable frequency of access is also reviewed by its box plot as shown in Fig. 4.8.

The frequency of access is having a minimum of 0 and a maximum of 242. The median is 9. and the majority of the values are within 100 and few values are above 100. It is also observed that accounts not having any files are accessed many times. The frequency of access to 897 records (which are not having any files, empty folders) is exemplified in Fig. 4.9. This information will not provide any scope in machine learning for the identification of shared dependency. The main aim of the work is to predict sharing dependency for the dynamic provisioning of cloud resources. The empty folder's access will not require any provisioning of resources. Such folders would have been accessed wrongly or the requested file is no longer available. Hence, these 897 records are filtered out from the experiment. In addition to this, the folders have few empty files whose sizes are in few kilobytes (KB). However, these files are accessed many times, and such information will be useful in learning. Hence, those records are retained. Thereby, the dataset contains 7015 records initially reduced to 6118 records after cleaning.

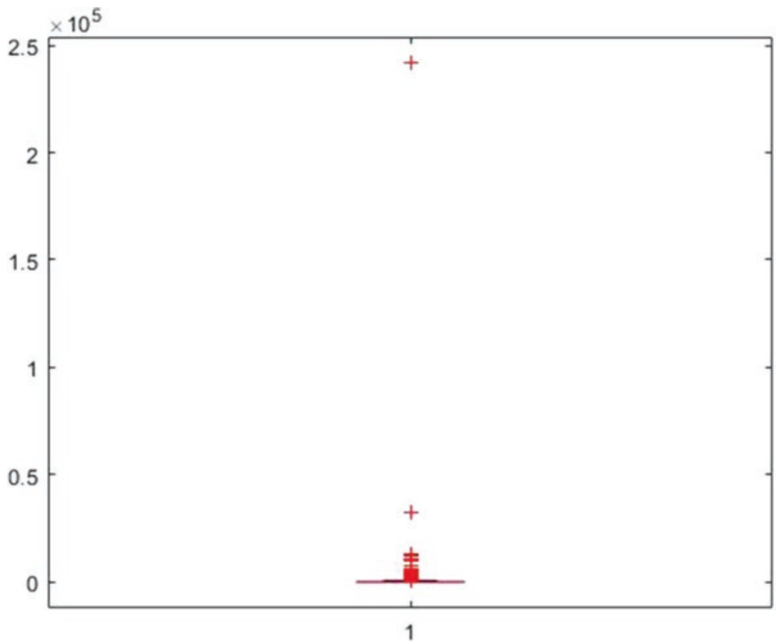


Fig. 4.7 The box plot for folder size

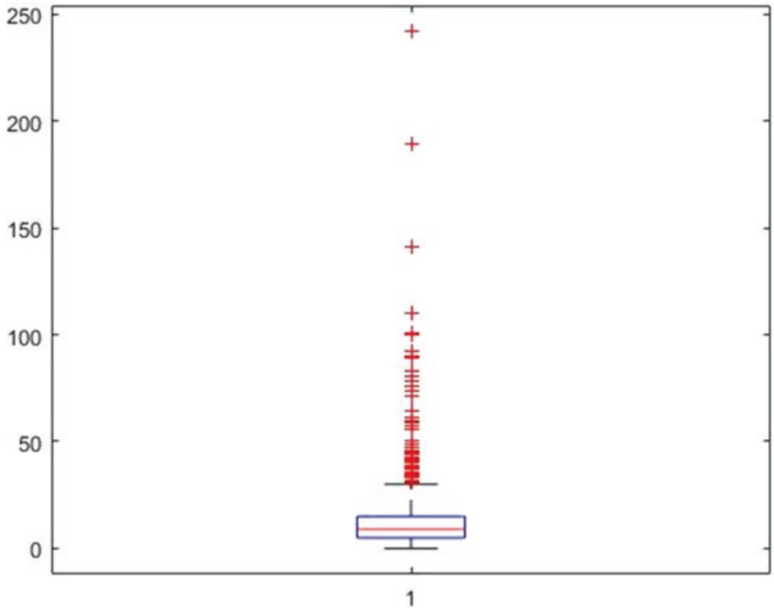


Fig. 4.8 The box plot for frequency of access

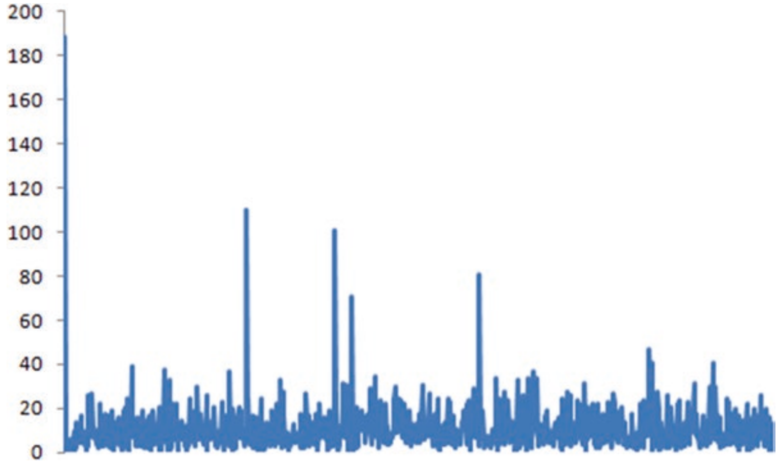


Fig. 4.9 The frequency of access to empty folders

4.3.6 Model Training and Analysis

After the cleaning, the experiment is set up with the number of files, and folder size is as predictors and frequency of access as a response. The experiment is carried out on Matlab R2016b using the classification leaners app. The response variable contains 65 classes for which the predictors are trained in the model. The scatter plot of predictors is shown in Fig. 4.10. From the scatter plot, it is clear that the predictors are not linearly related and cannot be trained using any linear models. The relationship of predictors is non-linear and it requires non-linear models. The nature of the tasks in the experiment is to predict the frequency of access from a given number of files and folder size. The decision tree and clustering models are more suitable for such situations. Thereby, the chapter uses ensemble learning for the training.

The predictors are fed to the classification learner app with the default configuration of all the ensemble learners. The results are summarized in Table 4.3. The boosted trees yield the highest accuracy of 7.1%, and all other models are providing an accuracy below 7.1%. The ROC curves of all the models are shown in Fig. 4.10. All the models provide an AUC value of around 0.5, and around 50% of the data is properly mapped to positive and negative classes as shown in Fig. 4.11.

It is also understood that the response variable has 65 classes of the frequency of access not well mapped by the number of files and folder sizes. To fine-tune the results, we have created a new class variable based on the number of files, folder size, and frequency of access. Here, these three variables are taken as the predictors. The response variable is prepared by using quartiles of each variable. Each variable has four quartiles, and all these four quartiles of each variable are combined to quartile values of other variables. Thereby, 64 quartile combinations are created and the same is modeled as response classes for the data. Then, for every sample, the

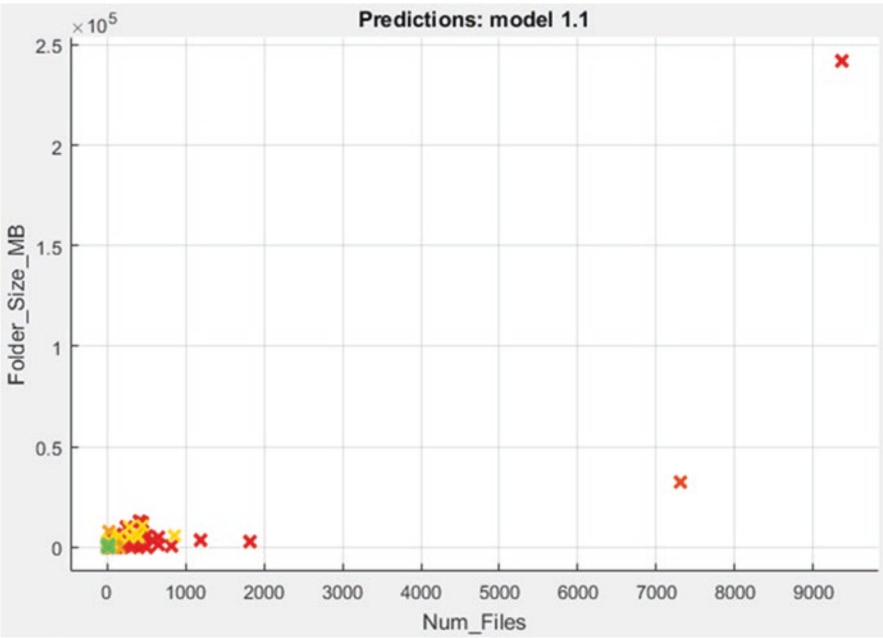


Fig. 4.10 The scatter plot between Num_Files and Folder_Size_MB

Table 4.3 Ensemble learners – results (number of predictors is two)

Ensemble learner	Accuracy (%)
Model 1.1 boosted trees	7.1
Model 1.2 bagged trees	5.1
Model 1.3 subspace discriminant	6.9
Model 1.4 subspace kNN	5.9
Model 1.5 RUSboosted trees	4.0

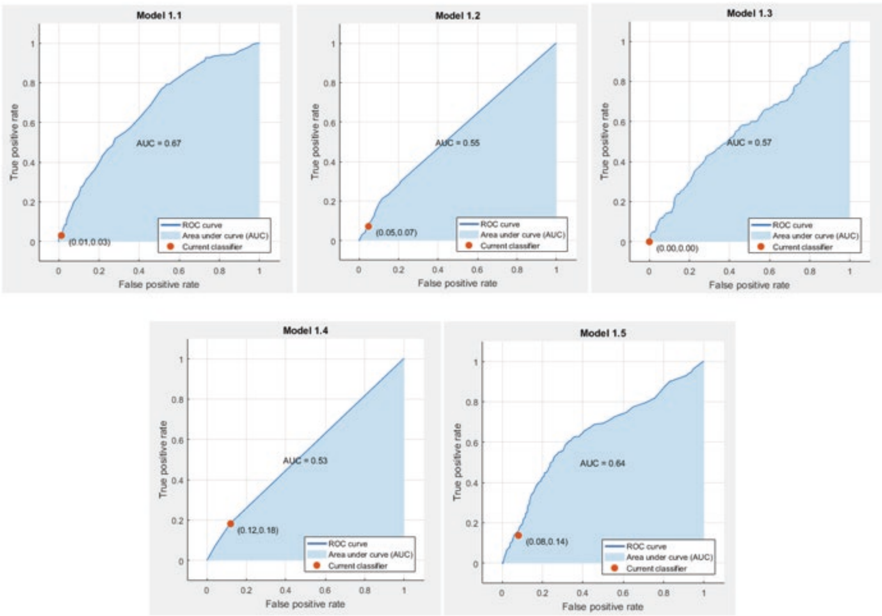


Fig. 4.11 ROC curve for the ensemble learners – number of predictors is two

corresponding quartile in which the variable is present and quartile combination of all three variables are computed.

Table 4.4 shows the quartile values of all the predictors used in the experiment. The response class is computed as follows. For example, if the sample contains values of Num_Files as 12, Folder_Size_MB as 684.58, and Frequency _Access as 18, then the quartile for Num_Files will be 4, Folder_Size_Mb will be 4, and Frequency_Access as 4. The response class for this sample will be 444. Thus, the response class for each sample is computed. Now, the model is trained using these three variables as a predictor and combined quartile values of variables as the response variable.

Table 4.5 shows the improvement in accuracy compared to the results in Table 4.4. The bagged trees provide the highest accuracy of 99.8%, and the boosted tree yields the second highest accuracy of 88.1%. All other models provide an

Table 4.4 Quartile values of predictors

Parameters	Num_Files	Folder_Size_MB	Frequency_Access
Min	1	0	1
Q1	4	91.8675	5
Q2	6	203.725	9
Q3	8	362.2425	15
Q4	9369	241830.1	242
Max	9369	241830.1	242

Table 4.5 Ensemble learners – results (number of predictors is three)

Ensemble learner	Accuracy (%)
Model 1.1 boosted trees	88.1
Model 1.2 bagged trees	99.8
Model 1.3 subspace discriminant	14.8
Model 1.4 subspace kNN	38.5
Model 1.5 RUSboosted trees	39.1

accuracy of below 50% and improved the accuracy compared to the results in Table 4.4. The second approach improved the overall accuracy of ensemble classifiers by 50% at an average. The individual model wise improvements are 81% in boosted trees, 94.70% in bagged tress, 7.90% in subspace discriminant, 32.60% in subspace kNN, and 35.10% in RUSboosted trees.

Figure 4.12 shows the performance of each model through ROC curves, and all the models can improve the AUC values compared to the previous results in Fig. 4.11. Now, the models are providing AUC values between 0.9 and 1.

4.3.7 Comparative Results

We have experimented with two different methods. First, we have taken a dataset with two predictors (Num_Files and Folder_Size_MB) and Frequency_Access as Response. The training results gave a maximum accuracy of 7.1% and an AUC of 0.67. This is achieved by the boosted tree. Second, we have taken all three variables as predictors (Num_Files, Folder_Size_MB, and Frequency_Access) and the newly computed quartile combination as a response. The training results for this method yield a maximum accuracy of 99.8% and an AUC of 1. Comparatively, the second method was found best for the NEC personal cloud dataset. This method can predict the demands of the quartile combinations using Num_Files and Folder_Size_MB and Frequency_Access. Method 2 improved the overall accuracy of all the models by 50% at an average and AUC by 45% at an average. The bagged tree model was found as the best model for the NEC cloud dataset.

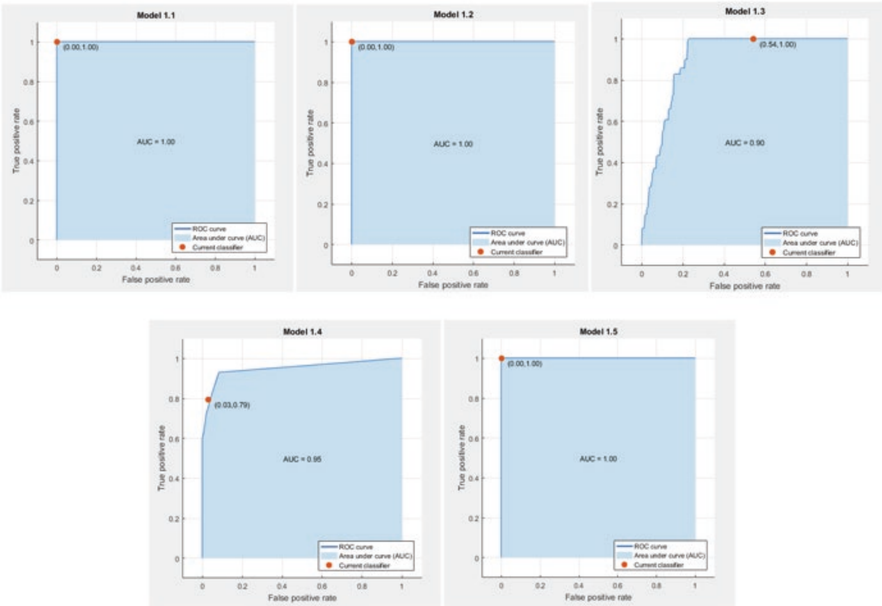


Fig. 4.12 ROC curve for the ensemble learners – number of predictors is three

4.4 Conclusions

This chapter presents ensemble classifier analysis to the NEC personal cloud dataset that contains two files File Traces and Sharing Traces for 2 years and 6 months from 7th March 2013 to 9th September 2015. From the dataset, we have extracted three 4 parameters, Owner ID, Number of Files, Folder Size in MB, and Frequency of Access. The owner ID is omitted, and the other three parameters are taken for learning. The ensemble learning contains five models boosted trees, bagged trees, subspace discriminant, subspace kNN, and RUSboosted trees. The training is done by two methods. The first method attempted to map the relationship between the number of files and folder size to the frequency of access. The second method attempted to find the relationship between all these three parameters to newly computed quartile combinations. The experimental results show that method 2 is best for the NEC data set and the Bagged Tree model can excel in the prediction that provides an accuracy of 99.8% and AUC of 1. Method 2 improves the accuracy of all the models by 50% and AUC by 45% compared to method 1. Thus, the bagged tree model with three predictors outperforms all other models of ensemble classifiers, and it is found as the best model for NCE personal cloud dataset in predicting sharing dependency. It is concluded that the bagged tree model is the best in estimating the sharing dependency of personal clouds in multi-cloud environments.

References

1. Vecchiola, C., Calheiros, R. N., Karunamoorthya, D., & Buyyaa, R. (2012). Deadline-driven provisioning of resources for scientific applications in hybrid clouds with Aneka. *Elsevier Journal of Future Generation Computer Systems, 28*, 58–65. https://doi.org/10.1016/j.future.2011.05.008
2. Islam, S., Keunga, J., Lee, K., & Liu, A. (2012). Empirical prediction models for adaptive resource provisioning in the cloud. *Elsevier Journal of Future Generation Computer Systems, 28*, 155–162. https://doi.org/10.1016/j.future.2011.05.027
3. Guruprasad, B. (2014). Resource provisioning techniques in cloud computing environment: A survey. *International Journal of Research in Computer and Communication Technology, 3*, 395–401.
4. Jeyarani, R., Nagaveni, N., & Vasanth Ramc, R. (2012). Design and implementation of adaptive power-aware virtual machine provisioner (APA-VMP) using swarm intelligence. *Elsevier Journal of Future Generation Computer Systems, 28*, 811–821. https://doi.org/10.1016/j.future.2011.06.002
5. Kousiouris, G., Menychtasa, A., Kyriazis, D., Gogouvitis, S., & Varvarigou, T. (2014). Dynamic, behavioral-based estimation of resource provisioning based on high-level application terms in cloud platforms. *Elsevier Journal of Future Generation Computer Systems, 32*, 27–40. https://doi.org/10.1016/j.future.2012.05.009
6. Li, C., & Li, L. Y. (2012). Optimal resource provisioning for cloud computing environment. *SpringerLink Journal of Supercomputing, 62*, 989–1022. https://doi.org/10.1007/s11227-012-0775-9
7. Javadi, B., Thulasiraman, P., & Buyya, R. (2013). Enhancing performance of failure-prone clusters by adaptive provisioning of cloud resources. *SpringerLink Journal of Supercomputing, 63*, 467–489. https://doi.org/10.1007/s11227-012-0826-2
8. Halder, K, Bellur, U, & Kulkarni, P (2012). Risk aware provisioning and resource aggregation based consolidation of virtual machines. IEEE fifth international conference on cloud computing (pp. 598–605).
9. Wang, C, Chen, J, Zhou, B. B., & Zomaya, A. Y. (2012) Just satisfactory resource provisioning for parallel applications in the cloud. IEEE eighth world congress on services (pp. 285–292). https://doi.org/10.1109/SERVICES.2012.38.
10. Ramachandran, L., Narendra, N. C., & Ponnalagu, K. (2012). Dynamic provisioning in multi-tenant service clouds. *SpringerLink Journal of Service-Oriented Computing and Applications, 6*, 283–302. https://doi.org/10.1007/s11761-012-0116-0
11. Zaman, S, & Grosu, D. (2011). Combinatorial auction-based mechanisms for VM provisioning and allocation in clouds**.** IEEE third international conference on cloud computing technology and science. https://doi.org/10.1109/CloudCom.2011.24.
12. Zhu, Q., & Agrawal, G. (2012). Resource provisioning with budget constraints for adaptive applications in cloud environments. *IEEE Transactions on Services Computing, 5*, 497–511. https://doi.org/10.1109/TSC.2011.61
13. Verma, A., Cherkasova, L., & Campbell, R. H. (2011). Resource provisioning framework for MapReduce jobs with performance goals. *SpringerLink Lecture Notes in Computer Science, 7049*, 165–186. https://doi.org/10.1007/978-3-642-25821-3_9
14. He, S., Guo, L., Guo, Y., Wu, C, Ghanem, M, & Han, R. (2012). Elastic application container: A lightweight approach for cloud resource provisioning. IEEE international conference on advanced information networking and applications. https://doi.org/10.1109/AINA.2012.74.
15. Nelson, V., & Uma, V. (2012). Semantic-based resource provisioning and scheduling in inter-cloud environment. IEEE xplore recent trends in information technology (pp. 250–254). https://doi.org/10.1109/ICRTIT.2012.6206823.
16. Pawar, C. S., & Wagh, R. B. (2013). Priority based dynamic resource allocation in cloud computing. International conference on intelligent systems and signal processing. https://doi.org/10.1109/ISSP.2013.6526925.

17. Zhanga, T, Dua, Z., Chen, Y., Xiang J. C, & Wang, X (2011) Typical virtual appliances: An optimized mechanism for virtual appliances provisioning and management. *Elsevier Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2010.11.925.
18. Javadi, B., Abawajy, J., & Buyya, R. (2012). Failure-aware resource provisioning for hybrid cloud infrastructure. *Elsevier Journal of Parallel and Distributed Computing, 72*, 1318–1331. https://doi.org/10.1016/j.jpdc.2012.06.012
19. Wang, Y., Nakao, A., Vasilakos, A. V., & Ma, J. (2011). On the effectiveness of service differentiation based resource provision incentive mechanisms in a dynamic and autonomous P2P network. *Elsevier Journal of Computer Networks, 55*, 3811–3831. https://doi.org/10.1016/j.comnet.2011.07.011
20. Biswas, S. Majumdar, B. Nandy, A. & El-Haraki (2014) Automatic resource provisioning: A machine learning-based proactive approach. IEEE 6th international conference on cloud computing technology and science (pp. 168–173). https://doi.org/10.1109/CloudCom.2014.147.
21. Gulati, A., Shanmuganathan, G., Holler, A., & Ahmad, I. (2011). Cloud-scale resource management: challenges and techniques. Proceedings of the 3rd USENIX conference on hot topics in cloud computing. https://dl.acm.org/doi/10.5555/2170444.2170447.
22. Sotomayor, B., Montero, R. S., Llorente, I. M., & Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. *IEEE Computer Society, 13*, 14–22. https://doi.org/10.1109/MIC.2009.119
23. Amazon CloudWatch. http://aws.amazon.com/cloudwatch.
24. Wang, W., Niu, D., Li, B., & Liang, B. (2013) Dynamic cloud resource reservation via cloud brokerage. *Distributed Computing Systems*. https://doi.org/10.1109/ICDCS.2013.20.
25. Buyya, R., Garg, S. K., & Calheiros, R. N. (2011). SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. Proceedings of the 2011 international conference on cloud and service computing. https://doi.org/10.1109/CSC.2011.6138522.
26. Moore, L. R., Bean, K., & Ellahi, T. (2013). A coordinated reactive and predictive approach to cloud elasticity. Fourth international conference on cloud computing, GRIDs, and virtualization.
27. Bodik, P. et al., (2009). Statistical machine learning makes automatic control practical for internet data centers. In Proc. HotCloud (pp. 1–5).
28. Liu, H., & Wee, S. (2009) Web server farm in the cloud: Performance evaluation and dynamic architecture. In Proceeding of 1st International Conference on Cloud Computing (pp. 369–380).
29. Bu, X., Rao, J., & Xu, C.-Z. (2009). A reinforcement learning approach to online web systems auto-configuration. In Proceeding 29th IEEE ICDCS.
30. Tesauro, G., Jong, N. K., Das, R., & Bennani, M. N. (2006). A hybrid reinforcement learning approach to autonomic resource allocation. In Proceeding of IEEE international conference on autonomic computing
31. Bodik, P. et al., (2009). Automatic exploration of datacenter performance regimes. In Proceeding of 1st workshop ACDC (pp. 1–6). https://doi.org/10.1145/1555271.1555273
32. Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., & Tantawi, A. (2005). An analytical model for multi-tier internet services and its applications. In Proceeding of ACM SIGMETRICS international conference on measurement and modeling of computer systems SIGMETRICS.
33. Rao, J., & Xu, C.-Z. (2010). Online capacity identification of multitier websites using hardware performance counters. *IEEE Transactions on Parallel and Distributed Systems, 22*, 426–438.
34. Dejun, J., Pierre, G., & Chi, C.-H. (2011). Resource provisioning of web applications in heterogeneous clouds. In Proceeding of 2nd USENIX Conf. Web Appl. Develop.
35. Villela, D., Pradhan, P., & Rubenstein, D. (2007). Provisioning servers in the application tier for e-commerce systems. *ACM Transactions on Internet Technology*. https://doi.org/10.1145/1189740.1189747.
36. Sharma, A., et al. (2008). Automatic request categorization in internet services. *SIGMETRICS Performance Evaluation Review, 36*, 16–25.

37. Bodik, P., Fox, O., Franklin, M. J., Jordan, M. I., & Patterson, D. A. (2010) Characterizing, modeling, and generating workload spikes for stateful services. In Proceedings of the 1st ACM SoCC (pp. 241–252).
38. Gracia-Tinedo, R., García-López, P., Gómez, A., & Illana, A. (2016) Understanding data sharing in private personal clouds. 2016 IEEE 9th international conference on cloud computing. https://doi.org/10.1109/CLOUD.2016.0059.
39. Breiman, L. (1996). Bagging predictors. *Machine Learning, 26*, 123–140.
40. Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.
41. Breiman, L.. https://www.stat.berkeley.edu/~breiman/RandomForests/
42. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*, 119–139.
43. Freund, Y. (2009) A more robust boosting algorithm. arXiv:0905.2138v1.
44. Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*, 1189–1232.
45. Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics, 28*, 337–407.
46. Schapire, R. E., et al. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of Statistics, 26*, 1651–1686.
47. Seiffert, C., Khoshgoftaar, T., Hulse, J., & Napolitano, A. (2008) RUSBoost: Improving classification performance when training data is skewed. 19th international conference on pattern recognition (pp. 1–4).

Part II
Multi-cloud Setup and Resource Management

Chapter 5
Resource Management Framework Using Deep Neural Networks in Multi-Cloud Environment

S. Brilly Sangeetha, R. Sabitha, B. Dhiyanesh, G. Kiruthiga, N. Yuvaraj, and R. Arshath Raja

5.1 Introduction

Today's enormous growth in Information Technology paves the way for the development of various mechanisms for transmitting information. The multi-cloud model eliminates strategies to communicate directly with a server for all kinds of services as the cloud computing model. This model is preferable instead to the online mode of communication. The content of information stored on the websites is retrieved directly by means of relevant requests. The systems that can define the online data recovery procedures are integrated here, making it easier for the user to recover the required assets with the help of this model [1–5].

In this model, the main factor to take into account is asset management, which is considered the model's most important task. The next major factor is the load

S. B. Sangeetha (✉)
Department of Computer Science and Engineering, IES College of Engineering, Thrissur, India

R. Sabitha
Department of Electronics and Communication Engineering, Hindusthan College of Engineering and Technology, Coimbatore, India

B. Dhiyanesh
Department of Computer Science and Engineering, Hindusthan College of Engineering and Technology, Coimbatore, India

G. Kiruthiga
Department of Computer Science and Engineering, IES College of Engineering, Thrissur, India

N. Yuvaraj · R. A. Raja
Training and Research, ICT Academy, IIT Madras Research Park, Chennai, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_5

balancing task, which is ultimately intended to ensure that the load is distributed uniformly, allowing the entire group of users to use the resources efficiently [6, 7].

Ultimately, the maximum use of the resources would improve the throughput factor and thereby reduce the energy consumption rates. Another benefit of the load balance strategy is that it reduces the response time substantially. This work has proposed a load balancing strategy for improving resource delivery and the quality of service factors, based on the various strategies [8, 9].

Adequate task scheduling is important to ensure high cloud productivity. Scheduling for the disseminated frameworks is an entire NP problem, so that in such circumstances customary booking techniques do not yield significant productivity [10, 11]. Scheduling is a critical concern for suitable situations that have to address some errands in numerous assets while improving the use of assets and the make-up. Task booking system charts for the most part provide cloud-based assignments to accessible assets as shown by their compliance qualities [12–14].

In addition, problem strategies are sorted into heuristic and conjectured. Heuristic algorithms are suspected of the status of assets as workable, such as the load or the length of employment prior to the schedule of employment. Surveyed scheduling methods depend on similar information data and the formal computer model as ideal reservation strategies, but by reducing arranged space, they defeat the NP complement of ideal schedulers [15, 16].

The collection of all information clearly presents new practical problems, where the valuable heuristic systems are [17, 18]. Swarm approaches to knowledge in a dispersed schedule are extremely prevalent and impressive. The main reason is that progress problems can be illuminated without the need for excess data on the issue.

This article therefore uses the need for the properties of gray wolf optimization (GWO) on a genetic algorithm (GA) model, which can effectively compute the selection of the optimal path with iterative training instances. This model has an optimal routing principle, which takes on less compute complexity and more energy efficiency from the cloud-based dynamic sensor VMs.

The outline of the paper is given as follows: Sect. 5.2 provides the problem of the study. Section 5.3 discusses the proposed scheduler. Section 5.4 concludes the entire work.

5.2 Problem Formulation

The problem formulation [19] for the present study is considered under the following assumptions: (1) prior knowledge on computational time period of each task and (2) similar overheads prior scheduling of tasks.

Task constraint is modeled as:

$$\gamma_i = \frac{c_i}{p_i}, i = 1, 2, \ldots N \tag{5.1}$$

$$r_{ij}\left(\forall i\right)=\begin{cases}0 & j=1\\ d_{ij-1} & j=2,3,\ldots,n_i\end{cases} \tag{5.2}$$

$$d_{ij}=r_{ij}+p_{ij},\ i=1,2,\ldots,N,\ j=1,2,\ldots,n_i \tag{5.3}$$

$$s_{ij}<s_{i'j'}<f_{ij}, \tag{5.4}$$

$$\text{if priority}\left(\tau_{i'}\right)>\text{priority}\left(t_i\right)\text{and} r_{i'j'}\geq r_{ij}\forall i,j,i',j' \tag{5.5}$$

where

$i2$ – Task index
$j2$ – Index of j^{th} task executed
N – Total tasks
T – Scheduled time
n_i – Total task executed
r_{ij} – Release time (j) of task τ_i
s_{ij} – Start time (j) of task τ_i
f_{ij} – Finish time (j) of task τ_i
i – Time required for executing τ_i with τ_i

The following are the metrics utilized for estimating the resource allocation based on the available resources:

Makespan: The general requirement that shows the entire duration of the tasks. When the machining rate is at lowest, the method of task planning can be efficient for VM. The service quality and scheduling can be improved if the makespan is smaller.

$$W=\frac{\sum_{i=1}^{N}w_i}{N} \tag{5.6}$$

where

w_i – Total completion time of a task t_i

Average Waiting Time represents the performance of total throughput and overall processing capability of cloud.

$$M=\frac{\min\sum_{i=1}^{N}\tau_i}{N} \tag{5.7}$$

where

τ_i– Waiting time for a task t_i

CPU Utilization Rate of VMs represents the actual resource requirements required by VMs to consume and allocate the task in each VM at every instant of time.

$$U_{ij}\left(t_i\right) = \frac{\sum_{j=1}^{v_i} C_{ij}^{con}\left(t_i\right)}{\sum_{j=1}^{v_i} C_{ij}^{alc}\left(t_i\right)} \tag{5.8}$$

where

$C_{ij}^{con}\left(t_i\right)$ – Consumed tasks at time multi-val t_i
$C_{ij}^{alc}\left(t_i\right)$ – Allocated tasks at time multi-val t_i

Failure Tasks Scheduling Rate represents the cloud stability.

$$Q\left(\%\right) = \text{Time processing tasks / Total time} \tag{5.9}$$

5.3 Proposed Method

There are various VMs in the cloud computing system. Different requests from the users are received, and separate capacities for the processing of such received requests are identified and incorporated. The time of the task here depends on the processing power of the virtual machines (VMs).

Figure 5.1 shows the architecture of the IoT-multi-cloud proposed to sense, collect, and transmit the data from the source VM to the destination VM. Three different levels, including sensing, data, and control plane, are used in this architecture.

The detection aircraft is the collection of IoT inputs, which is a heterogeneous device that senses and gathers input data at a greater rate. In the sensor plane, the IoT VMs must be fixed and the proposed architecture should not be mobile. Greater input data collection creates explosive traffic so that the need for an adequate transmission path is critical across cloud.

The proposed architecture uses a control plane consisting of the GWO model integrated with GA algorithm to alleviate this challenge. The deep sense model is so responsible that the connection or path is not congested with path assignment, packet selection, and flow control. In some cases, bursty traffic is checked with optimum packet selection through routing paths to ensure better packet delivery by the cloud.

The data plane is then used to transfer massive IoT data into the destination VM or sink or the gateway in the form of packets. A GWO [11] model integrated with the GA model, which aligns the goal of the proposed mechanism to improve the scheduling in the cloud, controls the routing between cloud VMs.

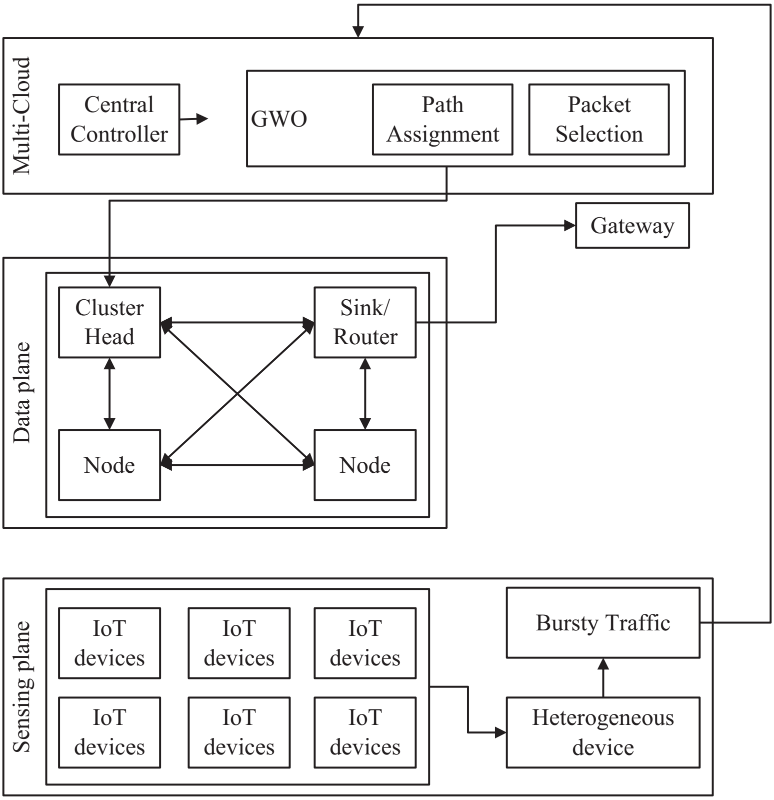


Fig. 5.1 Architecture for minimizing the time overhead using GWO

In the case of cloud data centers, the variations in computing power of the VM in question are seen, which means that there are significant differences in VMs during similar tasks. Tasks are generally different, and therefore, the performance efficiency of the various VMs makes the allotment of tasks to these VMs possible. Excess tasks would be assigned to VMs with better performance capacity. The above reason turns out to overload certain VMs, while the other machines remain idle. Ultimately, this issue would result in a waste of resources.

The efficiency of the cloud data centers is usually affected and reduced by the load imbalance of the VMs. This work thus focuses on the average load conditions applicable for the VMs, referring to the enhanced GWO to enhance the use of resources.

5.3.1 Scheduling Using GWO

The time multi-val between the submission of the IoT task in VM and its answer is viewed as defining the response time. The load balance factor is one of the factors influencing the process of reducing response time and increasing the reactivity of VMs. The easy procedure here is adopted to minimize the machining span, and the response time involves transferring the tasks from an overloaded VM to an overloaded VM, which is thus called the load balance procedure. In order to maintain the load conditions, it is important that the VM informs about its capacity to balance the load correctly.

The schedulers will be hosted based on the management nodes, from the computer nodes to the storage nodes. Based on the requests the scheduler selects relevant compute nodes for the VMs, it monitors requests that have been sent to a VM.

Task scheduling is a technique in cloud computing to ensure a dedicated resource is assigned work and that assigned work is completed. The resources include virtual computer elements or hardware. The programming activity in turn is performed by a programmer. The scheduler assigns multiple users to occupy a certain part of a resource for a QoS. The GWO scheduler allows the computer system to try multitasking for each CPU.

The scheduler prioritizes each task according to the user's needs and therefore the multitasks in parallel distributed applications set the schedule of work over idle VMs to complete the process soon. The main problem with task execution lies in the increase of parallelism, as it depends on another task to perform a task in cloud computing. Figure 5.2 shows the process of scheduling using GWO.

5.3.2 Load Balancing

This work classifies the load balance approach into two different phases. The first phase relates to the two-tier task planning strategy, which focuses on the dynamic needs of each user and the achievement of high resource utilization through the adopted load balance mechanism. In this case, load balances are used to map the IoT tasks to the entire VMs, followed initially by the VMs in the resource hosting tasks with the objective to improve the overall Cloud-IoT performance.

In the first phase itself, certain investigations, like identification of the CPU uses and the determination of memory requirements, must be completed with a determination of the number of available cycles, etc. The second stage involves determining the resources available and the amount of resources needed in the immediate future; on the basis of the resource requirements which instances would be either discarded or added, this would finally be seen as the user's prescribed status.

Load checking on VMs should be carried out regularly; this checking is wisely performed by the proposed algorithm; on the basis of observations, the following strategy is used to deduce load migration. It is the primary task of this algorithm to

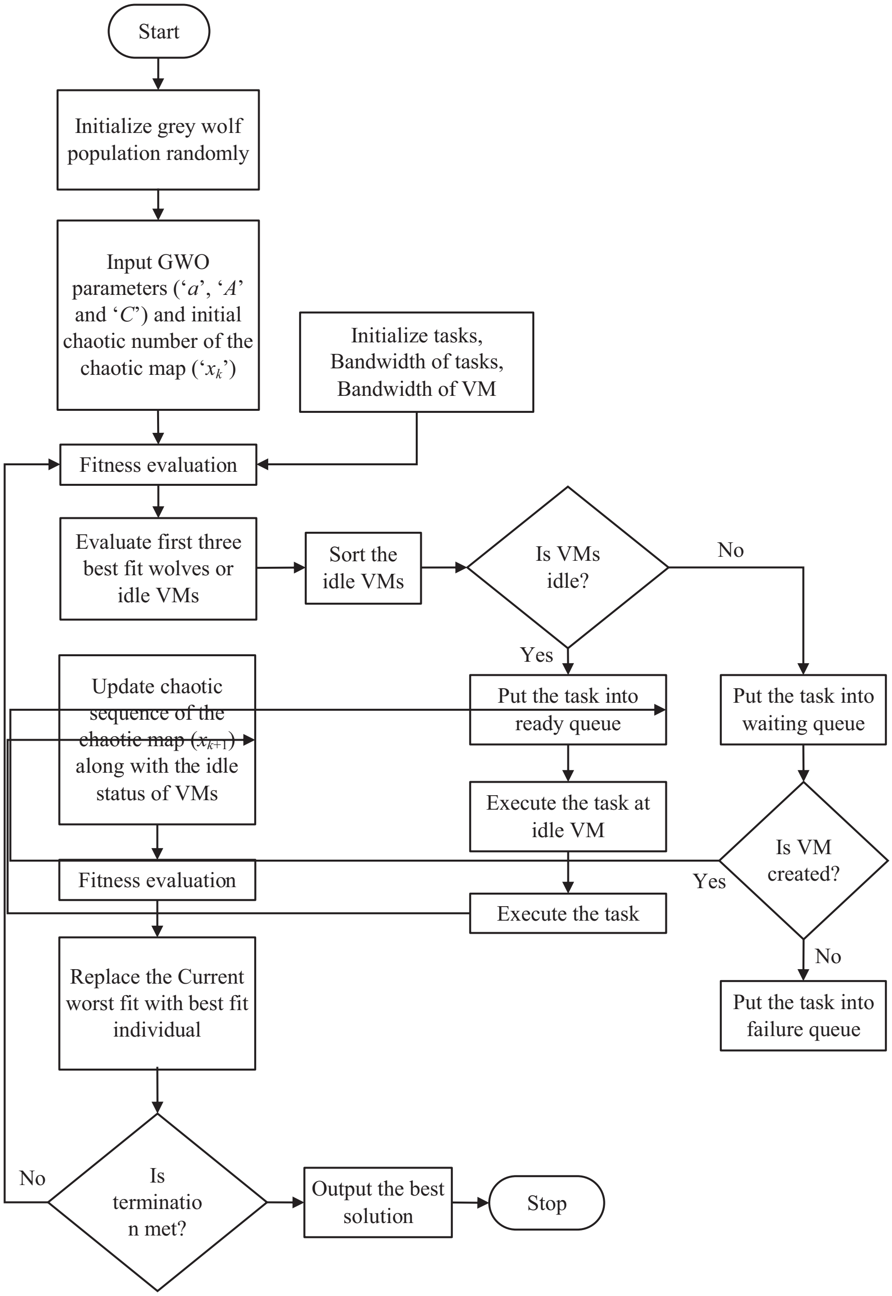


Fig. 5.2 Slot scheduling architecture

Pseudo Code of Cloud Scheduling

```
Initialize the wolfs
Initialize the VM in clouds
Initialize the IoT devices
Assign the value of threshold
While all VMs is found to be optimally balanced with loads
(apply GWO Algorithm – See pseudocode of GWO)
if (load < threshold)
if (load < threshold)
{
 do
 {
 Assign the scheduled task to the VM present in cloud
 Sort the scheduled task
 Estimate and update the distance of Wolfs
 } while (load < threshold)
}
else
Update the particles with updated solution
Search for similar neighbor VM
Finally check the load balanced VMs
Pseudocode of GWO
For i = 1 : n
 z_i ← i^th row vector elements of F;
 Generate initial search agents G_i (i=1, 2,….,n)
 Initialize the vector's a, A and C
 Estimate the fitness value of each hunt agent
Repeat
For i=1: G_s (pack size of grey wolf)
 Update the current hunt agent location
End for
Calculate the fitness value of the entire obtained hunt agents
Update the values of the vectors a, A and C
Update the value of first three best hunt agent
I=I+1
Check if I ε I_max (maximum iterations i.e. the Stopping criteria)
Output largest k eigenvectors G
End For
```

identify the loading conditions of VMs on the basis of the differentiation of machines from loaded VM and loaded VMs, followed by transferring loads from loaded VM to loaded VMs.

This process ensures that the loads are evenly distributed. This reduces the response time and improves the utilization of resources through a uniform distribution of load. This task for observation is like the bee movement, which periodically searches for loads. This search task continues to reach a threshold value. When the threshold value is reached, the optimization identifies the minimum loaded VM, and the task is set to minimum loaded VM.

5.4 Results and Discussions

This section discusses the experimental studies to concentrate on the efficacy of GWO. Optimization of GWO is verified using a range of performance measures including a package delivery ratio, average end-to-end delay, standardized payload routing and IoT, and cloud network life. To our best, it is the first IoT-multi-cloud GWO technique, and therefore a comparison is carried out between GWO routing with the Ant Colony Optimization (ACO) model and Bees Swarm Optimization (BSO). The following are the five performance metrics:

Packet delivery ratio (*PDR*): PDR is defined as the ratio from the total amount of packets generated and transmitted by IoT-source VMs at the Cloud-based base station.

End-to-end average delay (*EAD*): The delay is determined as the late time when packets from source IoT VMs are successfully transmitted to the cloud sink VMs and back to the source IoT VMs. This time delay includes queuing at cloud VMs, IoT cache latency, air propagation delay, MAC Cloud layer transmission delay, and GWO transformation time for tracking.

Normalized routing payload (NRP): For each data packet that is being delivered, NRP is the total number of control packets transmitted to the cloud sink VM. When the data is sent via one hop, the packets generated by the GWO occur.

Normalized MAC payload (NMP): NMP is defined as the total number of packets with address resolution, routing and control packets, and overhead packets that include overheads generated for each IoT data packet in the cloud MAC layer.

Network lifetime: IoT-multi-cloud network lifetime is defined as the total time taken to simulate the IoT cloud from the start to the last packet transmitted following the death of IoT VMs because VMs lie in a remote location and fail to receive continuous power.

The GWO routing efficiency with routing loads is considered important to the PDR and the EAD. MAC payload is an effective measure of wireless media for data streams in which the measurements are independent.

5.4.1 Simulation Results and Discussions

The data packet rate of the 10 IoT-source VM for session generation is between 45 and 54 Mbps. However, the optimal selection of control and data packet transmission via GWO makes the network congestion without affecting communication; this can lead to an increase in network congestion.

The simulation results of PDR in different sessions, which includes packet transmission at IoT source VMs and receipt of the cloud sink VMs, are shown in Fig. 5.3. The results show that the PDR is relatively less than the increase in pause times when pause times are reduced. However, the PDR is reduced by an increasing number of sessions from 10 to 40 than conventional ANN and enhancement.

The increase in performance in the GWO model is because of the efficient calculation of data transmission paths. The conventional systems fail to calculate the routes where the generated data rates on IoT devices do not coincide. GWO model performance is thus considered as stable and stabilizes the routing connection with increased route stability and minimal connection failure.

Figure 5.4 shows the EAD for various sessions when the EAD is reduced and with the increased break times it increases. The EAD is however much lower than ANN and reinforcement education with increasing sessions. The increased delay is due to the selection of longer routes while EAD is calculated, which causes severe cloud network congestion.

Thus, meta-heuristic algorithm learning does nothing to balance the loads. However, the GWO model, after certain iterations, increases the computing capability of the routes by minimizing the data transfer rate at that time. In contrast,

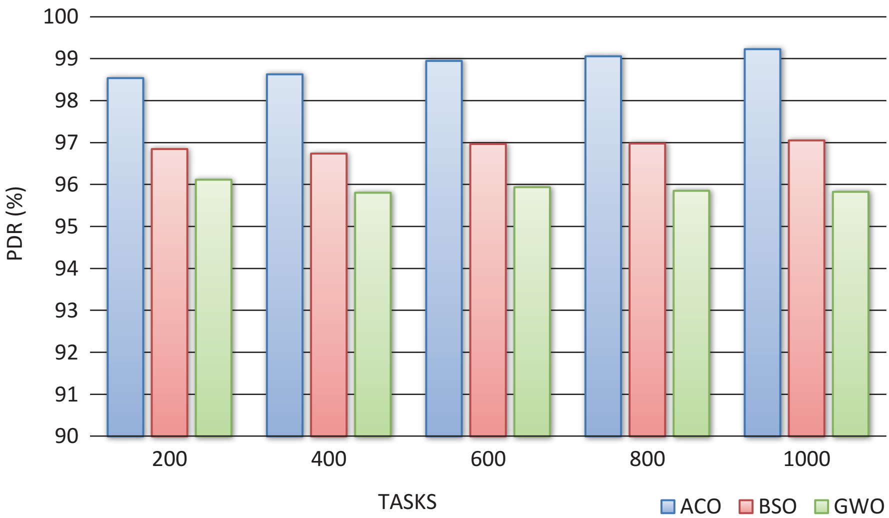


Fig. 5.3 Packet delivery rate

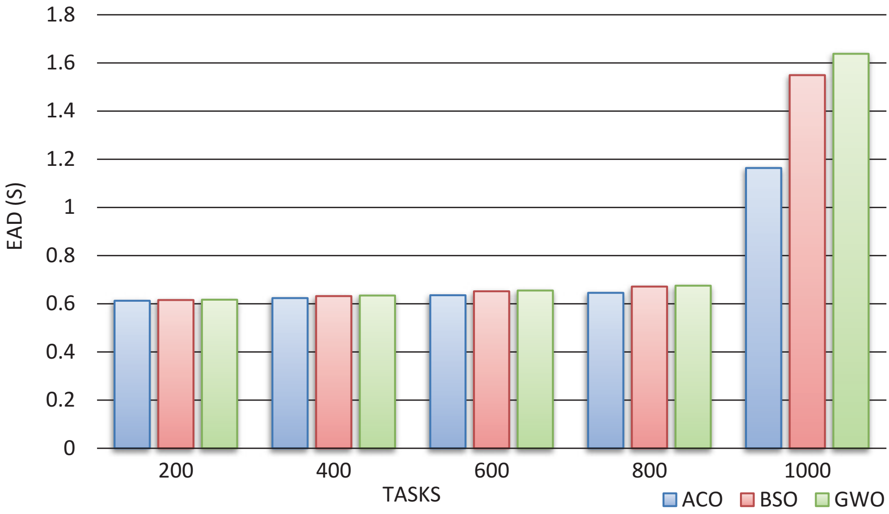


Fig. 5.4 Average end-to-end delay

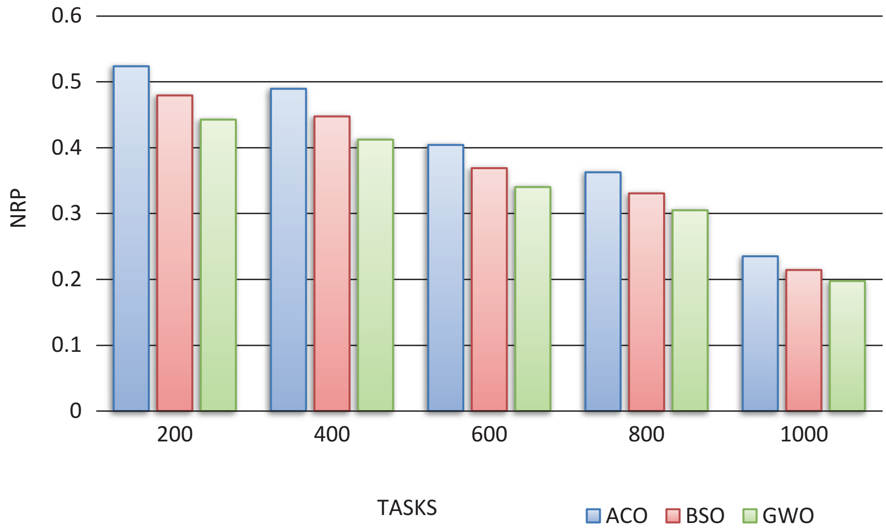


Fig. 5.5 Normalized routing payload

the fact that cloud VMs are delayed increases the EAD overall without affecting the kernels' stability.

The NRP and NMP for various sessions, which lower NRP and NMP for increased pauses, are presented in Figs. 5.5 and 5.6. The NRP and NMP

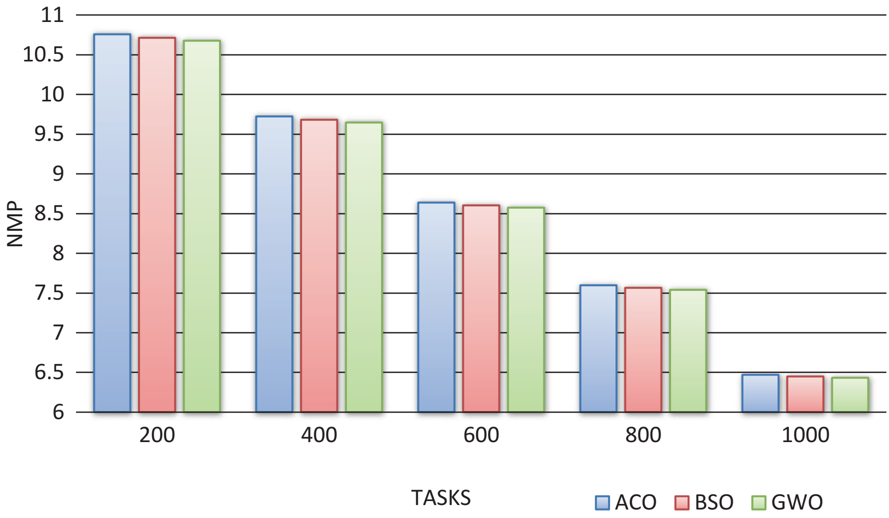


Fig. 5.6 Normalized MAC payload

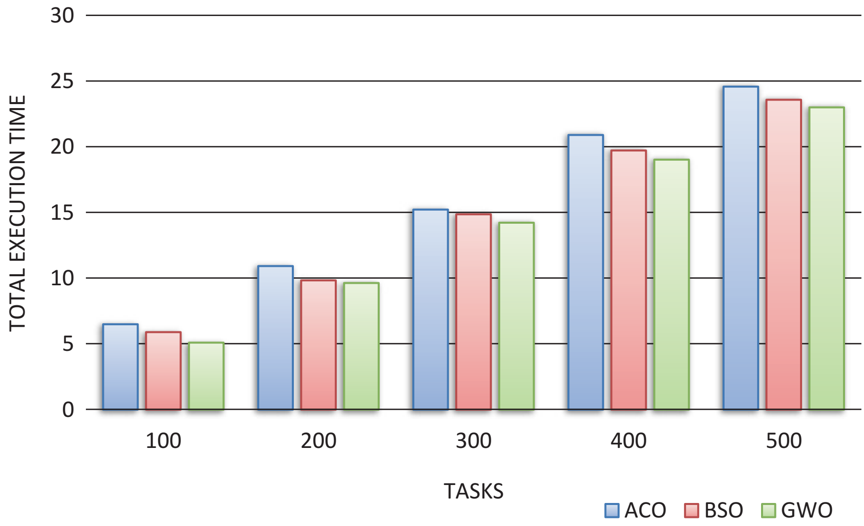


Fig. 5.7 Makespan for different number of tasks

significantly decrease with more and more sessions than ANN and enhancement learning. GWO handles address, routing, control packets, and overhead packets effectively with higher computation speeds.

Figure 5.7 shows the average overall makespan for different scheduling methods with different tasks. The GWO strategy consists of low make-up in comparison with the GWO strategy. The GWO does not waste energy with lower length tasks. The performance achieved by GWO has fallen by 13%. The main reason for this is that

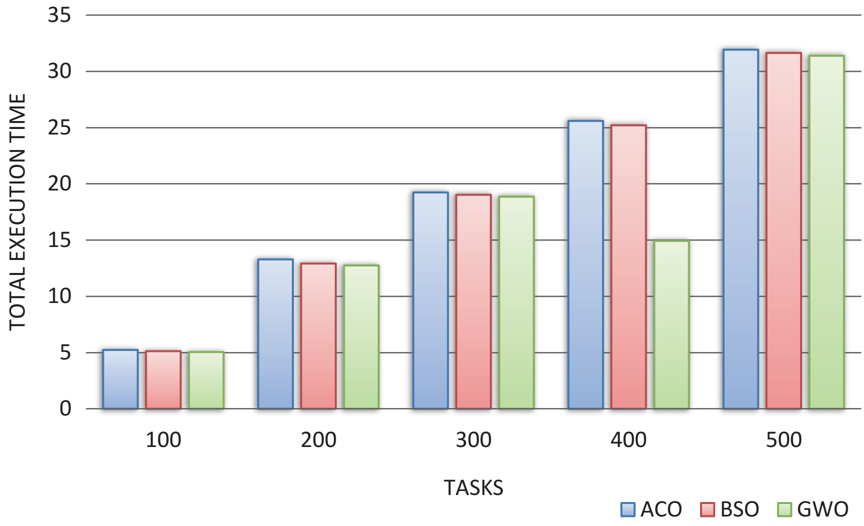


Fig. 5.8 Total execution time for different number of tasks

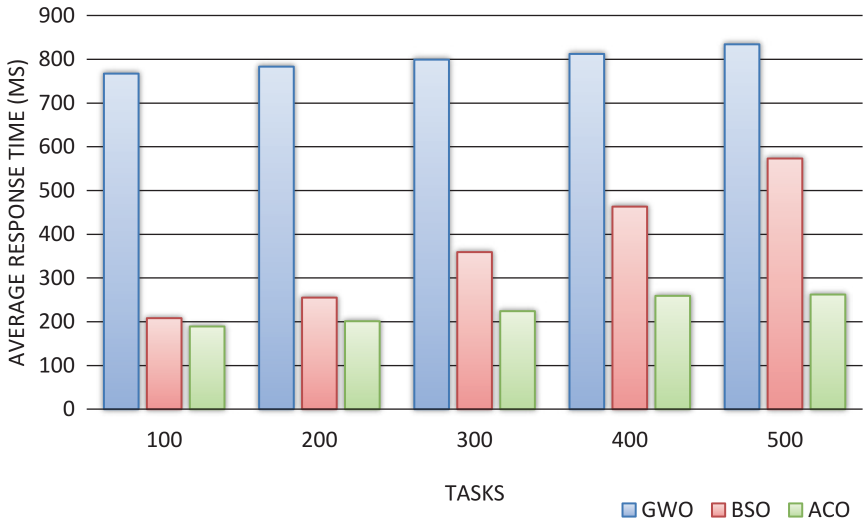


Fig. 5.9 Average response time in multi-cloud environment

the planned strategy takes on the characteristics of the tasks, total implementation time.

The resources are sufficient while there are small numbers of tasks. It takes a very long execution time (Fig. 5.8) to compare the GWO strategy. With discovering the whole area of search, wolf moves around the best position in the GWO algorithm. Compared to the entire method, GWO provides an average of 8–16% reduction in the total execution time. The metrics like increased average response time (Fig. 5.9),

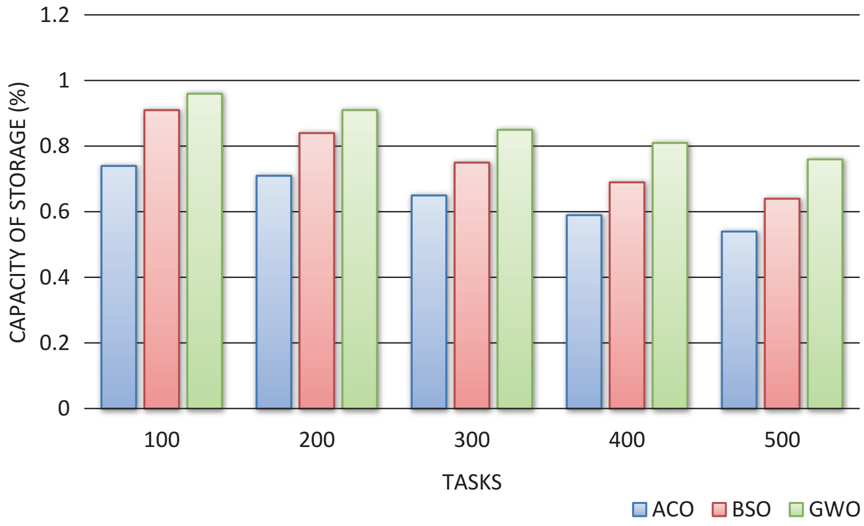


Fig. 5.10 Capacity of storage (%) in multi-cloud environment

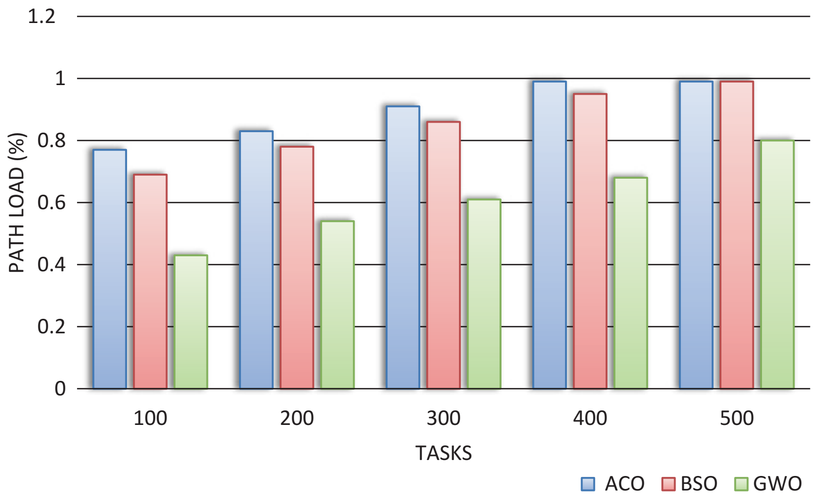


Fig. 5.11 Path load (%) in multi-cloud environment

increased storage capacity (Fig. 5.10), reduced path load (Fig. 5.11), and reduced cost (Fig. 5.12) shows improved performance by proposed GWO than existing methods.

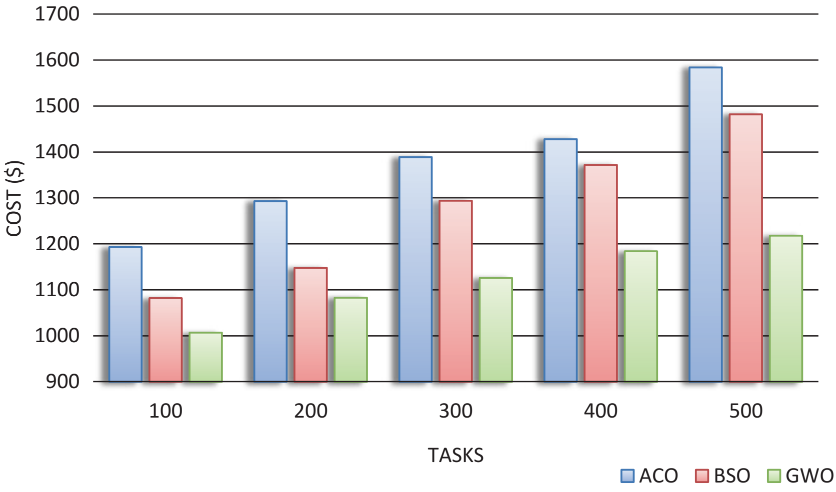


Fig. 5.12 Cost ($) in multi-cloud environment

5.5 Conclusions

In this paper, we framed a GWO model to enhance IoT-multi-cloud routing performance, thus increasing routing the performance of VMs. By adapting to the data acquisition speed by IoT VMs, GWO routes the packets effectively. IoT-multi-cloud multi-connection is effectively managed so that the data collection and transmission lack design defects. The results of the simulation show that the EAD is small and longer, with lower PDR, the scalability of the GWO is decreasing, and the risk of packet loss is high. But GWO reduces PDR on long routes which biases the samples and ensures a balanced scalability. The GWO routing model has a highly satisfactory scalability for shorter routes. The performance of the GWO model with high-speed packet routing provides improved packet transmission via Cloud, increasing IoT-multi-cloud longevity in IoT and Cloud sensor VMs as a result of increased residual energy.

References

1. Pham, X. Q., & Huh, E. N. (2016, October). Towards task scheduling in a cloud-fog computing system. In 2016 18th Asia-Pacific network operations and management symposium (APNOMS) (pp. 1–4). IEEE.
2. Basu, S., Karuppiah, M., Selvakumar, K., Li, K. C., Islam, S. H., Hassan, M. M., & Bhuiyan, M. Z. A. (2018). An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment. *Future Generation Computer Systems, 88*, 254–261.
3. Boveiri, H. R., Khayami, R., Elhoseny, M., & Gunasekaran, M. (2019). An efficient swarm-intelligence approach for task scheduling in cloud-based multi-net of things applications. *Journal of Ambient Intelligence and Humanized Computing, 10*(9), 3469–3479.

4. Ismail, L., & Materwala, H. (2018). Energy-aware VM placement and task scheduling in cloud-IoT computing: Classification and performance evaluation. *IEEE Multi-Net of Things Journal, 5*(6), 5166–5176.
5. Nguyen, B. M., Thi Thanh Binh, H., & Do Son, B. (2019). Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment. *Applied Sciences, 9*(9), 1730.
6. Al-Turjman, F., Hasan, M. Z., & Al-Rizzo, H. (2019). Task scheduling in cloud-based survivability applications using swarm optimization in IoT. *Transactions on Emerging Telecommunications Technologies, 30*(8), e3539.
7. Vashishth, V., Chhabra, A., & Sood, A. (2017, January). A predictive approach to task scheduling for big data in cloud environments using classification algorithms. In 2017 7th multi-national conference on cloud computing, data science & engineering-confluence (pp. 188–192). IEEE.
8. Hasan, M. Z., Al-Rizzo, H., Al-Turjman, F., Rodriguez, J., & Radwan, A. (2018, December). Multi-net of things task scheduling in cloud environment using hybrid wolf flame optimization. In 2018 IEEE global communications conference (GLOBECOM) (pp. 1–6). IEEE.
9. Wu, G., Bao, W., Zhu, X., & Zhang, X. (2018). A general cross-layer cloud scheduling framework for multiple iot computer tasks. *Sensors, 18*(6), 1671.
10. Yin, L., Luo, J., & Luo, H. (2018). Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing. *IEEE Transactions on Industrial Informatics, 14*(10), 4712–4721.
11. Zhao, X., & Huang, C. (2020). Microservice based computational offloading framework and cost efficient task scheduling algorithm in heterogeneous fog cloud network. *IEEE Access, 8*, 56680–56694.
12. Lin, W., Peng, G., Bian, X., Xu, S., Chang, V., & Li, Y. (2019). Scheduling algorithms for heterogeneous cloud environment: Main resource load balancing algorithm and time balancing algorithm. *Journal of Grid Computing, 17*(4), 699–726.
13. Xu, J., Hao, Z., Zhang, R., & Sun, X. (2019). A method based on the combination of laxity and ant colony system for cloud-fog task scheduling. *IEEE Access, 7*, 116218–116226.
14. Gawali, M. B., & Shinde, S. K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing, 7*(1), 4.
15. Cheng, N., Lyu, F., Quan, W., Zhou, C., He, H., Shi, W., & Shen, X. (2019). Space/aerial-assisted computing offloading for IoT applications: A learning-based approach. *IEEE Journal on Selected Areas in Communications, 37*(5), 1117–1129.
16. Li, W., Liao, K., He, Q., & Xia, Y. (2019). Performance-aware cost-effective resource provisioning for future grid IoT-multi-cloud system. *Journal of Energy Engineering, 145*(5), 04019016.
17. Kaur, K., Garg, S., Kaddoum, G., Ahmed, S. H., & Atiquzzaman, M. (2019). KEIDS: Kubernetes based energy and multi-ference driven scheduler for industrial IoT in edge-cloud ecosystem. *IEEE Multi-net of Things Journal*.
18. Deng, W., Yao, R., Zhao, H., Yang, X., & Li, G. (2019). A novel intelligent diagnosis method using optimal LS-SVM with improved GWO algorithm. *Soft Computing, 23*(7), 2445–2462.
19. Gen, M., Cheng, R., & Lin, L. (2008). Tasks Scheduling Models. Network Models and Optimization: Multiobjective Genetic Algorithm Approach (pp. 551–606).

Chapter 6
SLA-Based Group Tasks Max-Min (GTMax-Min) Algorithm for Task Scheduling in Multi-Cloud Environments

G. K. Kamalam and K. Sentamilselvan

6.1 Introduction

Nowadays, cloud is the fastest emerging field. People and industries were moving to keep their information to cloud [1]. The CSP provides various services to their customers, and the information can also be accessible everywhere in the world. Due to that, cloud users (CU) are drastically increased. As a result, the CSP is started toward to increase their resources (e.g., Software, infrastructure, Platform) to satisfy numerous customers [2, 3]. Multi-cloud is a scheme which supports a numerous cloud to execute several tasks. Cloud users select the service that provides the best among the available cloud services provided by numerous CSP. Multi-cloud avoids CU to depend on a single CSP. In multi-cloud environment, tasks are distributed among computing heterogeneous resources to minimize the execution time. Cloud service provider specifies the functionality of the service provided by them, and they are the sole responsible for delivering quality service to user. CSP clearly defines the cloud services QoS information in SLA document. CSP enhances quality service by considering parameters like QoS, uptime, downtime, pricing, policy for data protection, backup, service period, and security policy. Service providers need to provide good quality of service to their customers. If they fail to provide, CSP needs to pay the penalty to cloud users. While providing QoS, CSP is facing so many problems. For solving those issues, they use several algorithms and techniques. One of the major issue the CSP faces is mapping meta-tasks to a machine effectively to achieve minimum makespan or minimum execution time of the mapped meta-tasks [4].

Min-min, max-min, opportunistic load balancing (OLB), genetic algorithm (GA), minimum execution time (MET), minimum completion time (MCT), and

G. K. Kamalam (✉) · K. Sentamilselvan
Department of Information Technology, Kongu Engineering College, Perundurai, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_6

simulated annealing algorithm resulted in efficient scheduling results. The algorithms were used to solve complex scientific problems like NP hard and NP complete problem. Overall solution is to reduce makespan and increase resource usage by minimizing resource idle time [5–7].

Significant focus points of the proposed calculations over the current calculations are as per the following. As a result, a variety of algorithms provides the facility to their customers for selecting the parameters in terms of execution cost and time. Here the major customers are more concern about the execution cost only, not considering the execution time. Some customers are considering penalty cost also. CSP is in the state to balance the execution time, cost, and their services to admire many customers [8–10].

Service level agreement (SLA) term is enacted among CSP and CU. Both the CSP and CU settle and sign in the document. This SLA may vary from each service provider. In that SLA agreement, they will define the services such as QoS, services up and downtime, and service periods starts and ends; cost and security details are clearly defined in the rules and regulation about the service. Both of them should meet their expectations. Otherwise, those violating the policies defined in the SLA need to face the consequences [11].

This chapter organization proceeds as follows: Section 6.1 brings out the introduction. Section 6.2 lists related works on SLA-based task scheduling. Section 6.3 illustrates the problem definition and cloud model for our work. Section 6.4 proposes the algorithm SLA-GTMax-Min. Section 6.5 illustrates the performance comparison for SLA-GTMax-Min and existing algorithms. Section 6.6 concludes our work.

6.2 Related Works

Cloud service (CS) is the information technology service provided and invoked through Internet-enabled cloud computing by an interface. Cloud service provider (CSP) is the third party who provides the cloud services through the Internet as pay-per-use model. Cloud Service Consumer (CSC) is the primary stakeholder that uses the CS. Cloud SLA (service level agreement) is a prime unit of contractual association among CSC and CSP of a cloud service. Service level objective (SLO) represents the expectation for CS attribute. Expectation is specified either in quantitative or in qualitative terms. The SLA fixes the SLOs for the cloud service.

Baset et al. [12] have systematically divided the SLA into various components and analyzed the analogy and disparity among SLAs of CSP and provided direction on elucidating SLAs for future CS. The author also highlighted the challenges associated in describing performance-based SLAs. Maurer et al. [13] introduced an adaptive SLA matching approach, which provides an easy way of mapping among available private, public SLA templates of cloud market. The heuristic technique involved in the adaptation approach balances between cost and benefit in performing the SLA mapping process efficiently. Gao et al. [14] introduced method for managing the resources dynamically. The method provides effective energy planning by performing voltage scaling and server consolidation in speedily changing dynamic workloads at a greater rate. It achieves energy savings and better SLAs in cloud data centers. Ranaldo et al. [15] presented bilateral method for negotiation of SLA. Service provider's acceptable region and the value of utility are considered as a precondition for negotiation. The negotiation process optimizes the utility value and competitive prices by removing the agreements that will indulge in violations. Ivanovic et al. [16] generate constraints on SLA conformance and violation. The author identified the viable instance for SLA conformance, violation, and cause for its occurrence. The method leads to an optimized SLA matching. The parameters considered for optimal matching are makespan, availability of the resource, and cost of the cloud service. Son et al. [17] have presented a SLA framework WLARA. The framework considers the workload of the resource and data center's geographical location to perform efficient resource mapping. It also provides an automated SLA negotiation process which leads to higher profit for the service providers. It focused on establishing SLA which deals with business, cloud service, and utilization of resource. The main motive is to gain high profit in business, to provide guaranteed cloud service to users, and to perform effective management of resource to achieve greater resource utilization. Thus, the proposed approach is based on negotiation mechanism for multi-objective SLA establishment. Farokhi et al. [18] have proposed HS4MC approach. The approach calculates the score of the comparable services based on the preference of the cloud users. It performs automatic selection of the service efficiently persuading SLAs based on the service ranking and the provider's SLA claims. HS4MC performs service selection efficiently in multi-cloud environment. Garcia et al. [19] have implemented a platform CloudCompass. It is an SLA-aware platform that deals efficiently with resource allocation life cycle. It provides the method to correct the QoS violations with the help of cloud infrastructure elasticity features. The platform supports heterogeneous environment and leads to benefits of lesser cost and greater efficiency. Emeakaroha et al. [20] presented architecture DeSVi. It is used to detect violations in SLA using resource monitoring technique. Depending on the user requirement, DeSVi deploys the service in a virtualized cloud environment. To suit the heterogeneous multi-cloud environment, CSPs collaborate with each other and provide services to cloud users by bringing out a common SLA. The architecture is evaluated for image rendering and transactional web heterogeneous applications to detect the optimal monitoring level of SLA parameters.

Franke et al. [21] made an experimental investigation on SLA decision-making between IT professionals. Lu et al. [22] have presented an approach that completely automates the SLA life cycle management. The author proposed a systematic management framework considering SLA fault-tolerance concerns and approaches in several autonomous layers and processed them in a parallelized manner. Abawajy et al. [23] proposed a framework for managing SLA. It provides a pathway for CSP in identifying deliverable QoS before signing an agreement with cloud users to optimize the QoS guarantees. The framework is well organized to dynamically support distributed multimedia content adaptation. Son et al. [24] have proposed an effective mechanism which directs negotiation parameters adaptively and balances the cloud on high workload. The mechanism is well organized to support multi-objective SLA establishment and provides a direction for identifying the parameters that result in amenable cloud pricing and efficient management of resource. Panda et al. [25] have proposed two techniques SLA-Min-Min and SLA-MCT schedule tasks efficiently satisfying three different levels of SLA customer requirements. Proposed algorithm suits for an environment which is of heterogeneous and multi-cloud nature. The algorithm efficiently brings out a balance between the total completion time of set of tasks that are scheduled and gain cost for the services provided by the CSP.

Single-phase and two-phase heuristic techniques perform the mapping process efficiently in grid and cloud environment. Single-phase heuristic algorithms include cloud list scheduling [26], MCT [27], MET [28], OLB [29], and minimum completion cloud [30], SLA-MCT. Two-phase heuristic algorithms include SLA-Min-Min, cloud Min-Min scheduling [26], and cloud Min-Max normalization [30]. Freund et al. [27] presented a heuristic task technique, MCT which schedules tasks to machine with minimum completion time. Panda et al. [31] have included the parameter cost in the proposed two algorithms Profit-MCT and Profit-Min-Min, schedules task efficiently to available cloud with minimum execution cost. Rajganesh and Ramkumar [32] specified that in cloud environment, users have the flexibility to acquire the services given by different CSP as per requirement. Cloud users are not aware about the quality service provided by CSP and trust about the provided service. Authors proposed fuzzy logic trust evaluation system. The system acquires CU feedback about the services they utilized. A weight value is calculated for the user's feedback. Authors identified trust value by adding constraint in the system. Trust value is obtained from the weight value given for the user's feedback. Rajganesh and Ramkumar [33] stated that CU enjoys the benefits of the cloud services without having the required infrastructure. Selecting the services provided by numerous CSP is a difficult task. The authors presented a cloud broker which is smart enough to intelligently select the cloud service and highlights cloud broker's role in the selection procedure. Rajganesh and Ramkumar [34] presented a complete survey on the discovery of the services of the cloud. They also presented an architecture comprising cloud broker ranks the requirements of cloud users. Ranking assists in the cloud service decision-making. In this chapter, we presented heuristic heterogeneous SLA-based task scheduling technique SLA-GTMax-Min to bring out a proper balance between makespan and execution cost. Execution cost includes the gain cost, the amount paid to the CSP for the successful service rendered by them. Penalty cost is what CSP have to pay to cloud user for unsuccessful task completion. SLA-GTMax-Min algorithm schedules the task efficiently to available cloud as per SLA defined by the cloud users. SLA includes three levels: levels 1, 2, and 3. Level 1 specifies scheduling process to be done with minimum makespan, level 3 represents the task to be scheduled by the cloud with minimum execution cost, and level 2 considers both minimum makespan and minimum execution cost for task scheduling. The proposed algorithm gives the cloud user the flexibility in selecting the benefit they require by specifying the weight value for the parameters makespan and execution cost. SLA-GTMax-Min and existing algorithms are evaluated using Braun et al.'s benchmark datasets [35], and the proposed algorithm SLA-GTMax-Min brings out a proper balance among makespan and execution cost.

6.3 Problem Definition

6.3.1 Environment Setup

In muti-cloud environment, clouds cooperate in performing the task execution. Set of clouds belong to the cloud service providers. Through the Internet, CSP provides service to CU based on SLA. The model assumed is IAAS cloud system, and the primary resource service provided is the CPU. A manager server for each cloud maintains the status of its computational resources. When CSP receives the task requests from the cloud users, the manager server performs SLA with the cloud users based on the task execution time, gain, and penalty cost and performs mapping tasks to appropriate cloud for execution. Mapping is referred as NP-hard problem. Heuristic algorithms perform the mapping of tasks to clouds satisfying SLA. Proposed heuristic algorithm and existing algorithm are implemented and compared by simulation study based on a set of common assumptions.

To make ease of comparisons among existing and proposed heuristic algorithms, certain simplifying scheduling assumptions are as follows:

1. A meta-task T defines group of tasks without dependencies in the data required for executing the tasks.
2. Perform static scheduling/mapping of meta-tasks to the cloud.
3. Each cloud in the cloud set C is to execute one task in scheduled/mapped/assigned arrangement.
4. Size of the meta-task T is mentioned as 'l'.
5. In multi-cloud environment, the amount of cloud available in cloud set C is 'm'.
6. Sizes 'l' and 'm' are static and known prior.
7. In static heuristic, each task's expected execution time on cloud is known in advance to execution and presented in ETC (expected time to compute) matrix.

The proposed heuristic algorithm meets the following stated objectives:

1. Balancing gain cost, makespan, and penalty cost
2. Satisfying SLA

6.3.2 Model for Application and Problem Definition

In multi-cloud environment, cloud set $C = \{C_1, C_2, C_3, \ldots, C_m\}$ and cloud C_i, $1 \leq i \leq m$, where m clouds are provided by different CSP. Cloud C_i, $1 \leq i \leq m$ varies in computational resources from other clouds C_j, $1 \leq j \leq m$, and $i \neq j$. Task set, $T = \{T_1, T_2, T_3, \ldots, T_l\}$, each task T_i and T_j are independent, $1 \leq i \leq 1$, $1 \leq j \leq 1$, and $i \neq j$.

6.3.2.1 Expected Time to Compute Matrix (ETC)

Table 6.1 lists the expected execution time ETC_{ij} of task T_i, on cloud C_j, where $1 \leq i \leq 1$, $1 \leq j \leq m$.

Row represents task T_i execution time on clouds belonging to set C. Similarly, column represents the execution time on a cloud for each task belonging to set T [36–38].

6.3.2.2 Expected Gain (EGM) Matrix

Based upon the ETC matrix, the gain cost is derived. Matrix EGM represents the gain cost. Cloud cost is maximum if it executes the task with minimum execution time, and for other clouds, the cost reduces in multiples of two, $1 \leq i \leq 1$, $1 \leq j \leq m$. Gain cost is credited to CSP if the cloud user's task executes successfully satisfying SLA. In general, cloud with minimum execution time will be provided for service by CSP with high cost and cloud with maximum execution time for service at a low cost.

6.3.2.3 Expected Penalty (EPM) Matrix

Based upon the matrix EGM, the penalty is derived and is presented in matrix EPM. Penalty cost is credited to the cloud user if the CSP cannot complete the task requested by the cloud user. The penalty cost is half (i.e., 50%) of the gain cost for simplifying assumptions.

Table 6.1 An ETC matrix

	C_1	C_2	……	C_m
T_1	ETC_{11}	ETC_{12}		ETC_{1m}
T_2	ETC_{21}	ETC_{22}		ETC_{2m}
‘‘‘‘‘			……	
T_l	ETC_{l1}	ETC_{l2}	……	ETC_{lm}

6.3.2.4 Agreement Level of Service

The kind of service the CSP should provide to the cloud user is represented in SLA. The service requested by the cloud user is faster completion of the task submitted to the CSP or the task to be completed with minimum gain cost, or the cloud user may expect some percentage of task completion time and task gain cost. Agreement levels 1, 2, and 3 represent three types of agreement level of service provided by the CSP to the cloud user. The $x1$ and $x2$ represent the agreement level of service expected by the cloud user. These values are expressed as the linear combination of $x1$*execution time and $x2$*gain cost, such that $x1 + x2 = 1$. The agreement level of service, level 1, mainly considers only the task execution time to be minimum and does not consider the gain cost. The agreement level of service, level 3, primarily considers the task gain cost is to be minimum without considering task execution time. Agreement level service, level 2, indicates the combination of levels 1 and 3, based on cloud user preference in percentage of task's execution time and cost. For example, if $x1 = 0.2$, and $x2 = 0.8$, it specifies that the cloud user expects more importance for minimum execution time than gain cost. For level 1, the value of $x1$and $x2$ is 1 and 0; similarly, for level 3, the value of $x1$and $x2$ is 0 and 1.

6.3.2.5 SLA-GTMax-Min Scheduling Algorithm

SLA-GTMax-Min maps task and cloud based on customer agreement level. The kind of service the customer requires is specified in the SLA. The kind of service required is minimum completion time (α) or minimum gain cost (β). α and β values are represented in percentage, and it is normalized to 1, i.e., $0 \leq (\alpha) \leq 1$ and $0 \leq (\beta) \leq 1$. According to the α and β values, the scheduling of the tasks to the cloud is categorized in three levels, termed as levels 1, 2, and 3.When $\alpha = 1$ and $\beta = 0$, task scheduling is considered to be in level 1. When α = 'x' and β = 'y' $(0 \leq (x + y) \leq 1)$, the level of task scheduling is considered to be in level 2. When $\alpha == \beta = 1$, the task scheduling is considered to be in level 3. Level 1 tasks are scheduled to cloud with minimum completion time. Level 2 tasks are scheduled to cloud with minimum gain cost. Level 3 tasks are scheduled to cloud as per α and β percentage values. Table 6.2 lists the SLA-GTMax-Min algorithm terms.

6.3.2.6 An Illustration

Eight tasks are represented in task set, T = {T1, T2, …, T8}, and three cloud resources are represented in the cloud set C = {C1, C2, C3} are taken for illustration of the proposed SLA-GTMax-Min algorithm. The arrival time of tasks is considered as 0. The ETC matrix in Table 6.3 shows EET, service level of agreement of each task, and α, β values. EG and EP matrices are shown in Tables 6.4 and 6.5. For simplicity, the assumption is made that 50% of the gain cost is considered as the penalty cost. From the values illustrated in Table 6.3, it is clear that each task executes with

Table 6.2 Terms and purpose

Notations	Definitions
RT[j]	jth cloud ready time
EET	Expected execution time
ECT	Expected completion time
EGM	Expected gain matrix
EPM	Expected penalty matrix
SLA	Service level agreement matrix
L	Tasks present in task set T
M	Clouds present in multi-cloud environment
TECT[i]	Total ECT of ith task on all clouds
MS[j]	Makespan of jth cloud
GC[j]	Gain Ccst of jth cloud
PC[j]	Penalty cost of jth cloud
NCT	Normalized completion time
NGC	Normalized gain cost
WCT	Weighted completion time
WGC	Weighted gain cost
WS	Weighted sum

Table 6.3 EET, α, β values, and service level for eight tasks and three clouds

Tasks/clouds	C1	C2	C3	α	B	Service level
T1	745.2	839.8	1192.9	0.3	0.7	2
T2	5000.3	5084.6	7350.5	1	0	1
T3	2119.7	2975.5	3046.0	0.2	0.8	2
T4	2571.3	2788.2	3100.9	0.6	0.4	2
T5	1344.3	1559.0	1758.3	1	0	1
T6	4479.1	6283.3	8735.4	0.1	0.9	2
T7	3775.2	4506.4	4902.4	0	1	3
T8	2227.6	5199.6	5896.1	0.8	0.2	2

Table 6.4 Expected gain matrix

Tasks/clouds	C1	C2	C3
T1	8	6	4
T2	10	8	6
T3	6	4	2
T4	8	6	4
T5	8	6	4
T6	10	8	6
T7	6	4	2
T8	8	6	4

Table 6.5 Expected penalty matrix

Tasks/clouds	C1	C2	C3
T1	4	3	2
T2	5	4	3
T3	3	2	1
T4	4	3	2
T5	4	3	2
T6	5	4	3
T7	3	2	1
T8	4	3	2

Table 6.6 Average execution time of tasks

Task	AET
T1	926.0
T2	5811.8
T3	2713.7
T4	2820.1
T5	1553.9
T6	6499.3
T7	4394.7
T8	4441.1

different execution time on each cloud because the efficiency of the proposed algorithm SLA-GTMax-Min suits heterogeneous multi-cloud environment.

The proposed algorithm SLA-GTMax-Min is illustrated as follows.

Step 1: Calculating Average Task Execution Time

AET calculated by the formula:

$$\mathrm{AET}_i = \left(\sum_{m}^{j=1} \mathrm{ETC}_{ij} \right) / m$$

Task T_1 AET calculated by the formula,

$$\mathrm{AET}_1 = \frac{25137.5 + 52468.0 + 150206.8}{3} = 75937.4$$

Similarly, for remaining tasks, AET value is shown in Table 6.6.

Step 2: Grouping Tasks

The tasks are grouped into two sets based on their average execution time. The task sets are named as MAX_AET and MIN_AET. $\lceil l/2 \rceil$ tasks with the highest average execution time are grouped under the task set MAX_AET and are arranged in descending order. The remaining tasks are grouped under the task set MIN_AET and are arranged in ascending order. Now, for the given scenario, the tasks grouped under the task set MAX_AET and MIN_AET are shown as follows:

$$MAX\_\mathrm{AET} = \{\mathrm{T6}, \mathrm{T2}, \mathrm{T8}, \mathrm{T7}\}$$
$$MIN\_\mathrm{AET} = \{\mathrm{T1}, \mathrm{T5}, \mathrm{T3}, \mathrm{T4}\}$$

Step 3

Schedule the tasks listed in the task set MAX_AET = {T6, T2, T8, T7} one by one in the order specified from left to right to cloud as per α and β values. First, we illustrate the scheduling of task T3 to the appropriate cloud using the values $\alpha = 0.1$ and $\beta = 0.9$. Task T6 requires SLA level 2; SLA-GTMax-Min schedules task T6 to the cloud which possesses minimum completion time and gain cost.

ECT of task T_i on all clouds is calculated as

$$\mathrm{ECT}_{ij} = \mathrm{RT}_j + \mathrm{EET}_{ij} \; 1 \le i \le 8, 1 \le j \le 3$$

TCET of task T_i on all clouds calculated as

$$\mathrm{TECT}_i = \sum_{m}^{j=1} \mathrm{ECT}_{ij}$$

TotalGain cost of task T_i on all clouds is calculated as

$$\mathrm{TGC}_i = \sum_{m}^{j=1} \mathrm{GC}_{ij}$$

NCT of task T_i on all clouds is calculated as

$$\mathrm{NCT}_{ij} = \frac{\mathrm{ECT}_{ij}}{\mathrm{TECT}_i}, 1 \le j \le 3$$

Normalized gain cost of task T_i on all clouds is calculated as

$$\mathrm{NGC}_{ij} = \frac{\mathrm{GC}_{ij}}{\mathrm{TGC}_i}, 1 \le j \le 3$$

WCT of task T_i on all clouds is calculated as

$$\mathrm{WCT}_{ij} = \mathrm{NCT}_{ij} * \alpha_i, 1 \le j \le 3$$

WeightedGain cost of task T_i on all clouds is calculated as

$$\mathrm{WGC}_{ij} = \mathrm{NGC}_{ij} * \beta_i, 1 \le j \le 3$$

WeightedSum of task T_i on all clouds is calculated as

$$\mathrm{WS}_{ij} = \mathrm{WCT}_{ij} + \mathrm{WGC}_{ij}, 1 \le j \le 3$$

Allocated cloud for a task Ti is calculated as

$$AC_i = \min\left(WS_{ij}\right), 1 \leq j \leq 3$$

Table 6.7 provides the values of EET, ECT, TECT, TGC, NCT, NGC, WCT, WGC, and WS of task T6 while planning to schedule to three different clouds C1, C2, and C3. Among the three clouds, WS value is minimum for cloud C3. Thus, task T6 is allocated to C3 earning gain cost of six units. Otherwise, C3 pays a three-unit penalty cost. RT of C3 becomes 8735.4.

In a similar fashion, the proposed algorithm assigns the tasks in the task set MAX_AET to the respective clouds based on the WS value and is listed in Tables 6.8, 6.9, and 6.10.

Step 4

The proposed algorithm SLA-GTMax-Min schedules the tasks listed in the task set MIN_AET = {T1, T5, T3, T4} one by one in the order specified from left to right to cloud as per α and β values. The proposed algorithm assigns the tasks in the task set MIN_AET to the respective clouds based on the WS value and is listed in Tables 6.11, 6.12, 6.13, and 6.14.

Step 5

The average cloud usage, makespan, gain, and penalty cost for scheduling tasks to clouds are shown in Table 6.15. As shown in Table 6.15, it is clear that SLA-GTMax-Min performs a balanced schedule among minimum gain cost and minimum makespan than existing algorithms.

Table 6.7 Task cloud allocation for task T6

T6	C1	C2	C3	TECT/TGC	Allocated cloud
RT	0	0	0		
EET	4479.1	6283.3	8735.4		
ECT	4479.1	6283.3	8735.4	19497.8	
GC	10	8	6	24	
NCT	0.23	0.32	0.45		
NGC	0.42	0.33	0.25		
WCT	0.02	0.03	0.04		
WGC	0.38	0.30	0.23		
WS	0.40	0.33	0.27		C3

Table 6.8 Cloud allocation for T2

T2	C1	C2	C3	Allocated cloud
RT	0	0	8735.4	
EET	5000.3	5084.6	7350.5	
ECT	5000.3	5084.6	16085.9	C1

Table 6.9 Cloud allocation for task T8

T8	C1	C2	C3	TECT/TGC	Allocated cloud
RT	5000.3	0	8735.4		
EET	2227.6	5199.6	5896.1		
ECT	7227.9	5199.6	14631.5	27,059	
GC	8	6	4	18	
NCT	0.27	0.19	0.54		
NGC	0.44	0.33	0.22		
WCT	0.21	0.15	0.43		
WGC	0.09	0.07	0.04		
WS	0.30	0.22	0.48		C2

Table 6.10 Cloud allocation for task T7

T7	C1	C2	C3	Allocated cloud
RT	5000.3	5199.6	8735.4	
EET			4902.4	
ECT			13637.8	C3

Table 6.11 Cloud allocation for task T1

T1	C1	C2	C3	TECT/TGC	Allocated cloud
RT	5000.3	5199.6	13637.8		
EET	745.2	839.8	1192.9		
ECT	5745.5	6039.4	14830.7	26615.6	
GC	8	6	4	18	
NCT	0.22	0.23	0.56		
NGC	0.44	0.33	0.22		
WCT	0.06	0.07	0.17		
WGC	0.31	0.23	0.16		
WS	0.38	0.30	0.32		C2

Table 6.12 Cloud allocation for task T5

T5	C1	C2	C3	Allocated cloud
RT	5000.3	6039.4	13637.8	
EET	1344.3	1559	1758.3	
ECT	6344.6	7598.4	15396.1	C1

Table 6.13 Cloud allocation for task T3

T3	C1	C2	C3	TECT/TGC	Allocated cloud
RT	6344.6	6039.4	13637.8		
EET	2119.7	2975.5	3046		
ECT	8464.3	9014.9	16683.8	34163	
GC	6	4	2	12	
NCT	0.25	0.26	0.49		
NGC	0.50	0.33	0.17		
WCT	0.05	0.05	0.10		
WGC	0.40	0.27	0.13		
WS	0.45	0.32	0.23		C3

Table 6.14 Cloud allocation for Task T4

T4	C1	C2	C3	TECT/TGC	Allocated cloud
RT	6344.6	6039.4	16683.8		
EET	2571.3	2788.2	3100.9		
ECT	8915.9	8827.6	19784.7	37528.2	
GC	8	6	4	18	
NCT	0.24	0.24	0.53		
NGC	0.44	0.33	0.22		
WCT	0.14	0.14	0.32		
WGC	0.18	0.13	0.09		
WS	0.32	0.27	0.41		C2

Table 6.15 Comparison of existing and proposed algorithm

Algorithm	Allocated tasks C1	C2	C3	Average cloud utilization	Makespan	Gain cost	Penalty cost
SLA-GTMax-Min	T2, T4, T5	T1, T8	T3, T6, T7	63.64657	16683.8	46	23
SLA-MCT	T2, T5	T1, T3, T4. T6, T8	T7	54.06162	18086.4	50	25
Execution-MCT	T1, T2, T3, T4, T5, T6, T7, T8	–	–	33.33333	22262.7	64	32
Profit-MCT	–	–	T1, T2, T3, T4, T5, T6, T7, T8	33.33333	35982.5	32	16
SLA-min-min	T2	T4, T8	T1, T3, T5, T6, T7	57.68038	17781.9	40	20
Execution min-min	T1, T3, T5, T6	T2, T8	T4, T7	87.43445	10284.2	52	26
Profit min-min	–	–	T1, T2, T3, T4, T5, T6, T7, T8	33.33333	35982.5	32	16

6.4 Evaluation Parameters

Evaluation parameters to measure the performance of SLA-GTMax-Min are listed further.

6.4.1 Makespan

Proposed heuristic algorithm SLA-GTMax-Min maps '*l*' tasks to '*m*' clouds. Makespan (MS) is the task's overall completion time. Makespan MS[*k*] of a particular cloud 'c_k' defined the sum of the expected execution time of '*t*' tasks mapped to cloud 'ck'. MS[*k*] is defined as follows:

$$\mathrm{MS}\left[k\right] = \sum_{t}^{i=1} \mathrm{ETC}_{ik}$$

Makespan (MS) is mathematically set as

$$\mathrm{MS} = \max(\mathrm{MS}\left[k\right], 1 \le k \le m$$

6.4.2 Average Cloud Utilization

Algorithm efficiency mainly considers better usage of the cloud resources. If set of tasks are scheduled on a particular cloud c_k and none of the tasks is scheduled on the cloud c_p, then the cloud c_k is set to be overloaded and the cloud c_p is set to be underutilized. To overcome this issue, algorithm has to be designed such that the cloud resources are better utilized. Cloud utilization of a cloud c_k (i.e., CU[*k*]) is defined as the ratio between the MS[*k*] and MS. Mathematically, cloud utilization CU[*k*] is calculated as

$$\mathrm{CU}\left[k\right] = \frac{\mathrm{MS}\left[k\right]}{\mathrm{MS}}$$

ACU is defined as the ratio among the total of MS of all clouds c_j, $1 \le j \le m$, and the product of MS and *m*. The average cloud utilization (ACU) is calculated as

$$\mathrm{ACU} = \frac{1}{m * \mathrm{MS}} \sum_{m}^{j=1} \mathrm{MS}\left[j\right]$$

6.4.3 Gain Cost

The set of clouds belong to the cloud service providers. Through the Internet, CSP provides a unified service to cloud users based on SLA. After the completion of the service successfully, the customer pays the cost to CSP based on SLA. Gain cost of a cloud c_k (i.e., GC[k]), $1 \leq k \leq m$ is the total GC of 't' tasks scheduled on cloud c_k and is defined mathematically as

$$\mathrm{GC}\left[k\right] = \sum_{t}^{i=1} \mathrm{EG}_{ik}$$

The overall gain cost (GC) is presented as

$$\mathrm{GC} = \sum_{m}^{i=1} \mathrm{GC}\left[i\right]$$

6.4.4 Penalty Cost

Cloud users' requests the service required for the execution of the meta-tasks by signing an agreement with CSP. If the service provided by the CSP is unsuccessful, the CSP pays the cost to customer. Penalty cost of a cloud c_k (i.e., PC[k]), $1 \leq k \leq m$ is the total PC of 't' tasks scheduled on cloud c_k and is defined mathematically as

$$\mathrm{PC}\left[k\right] = \sum_{t}^{i=1} \mathrm{EP}_{ik}$$

The overall penalty cost (PC) is presented as

$$\mathrm{PC} = \sum_{m}^{i=1} \mathrm{PC}\left[i\right]$$

6.5 Simulation Results

SLA-GTMax-Min experimental results are computed using Braun et al.'s benchmark datasets.

6.5.1 Benchmark Descriptions

Mathematical outcome of the proposed application is evaluated by Braun et al.'s dataset model. Braun et al.'s benchmark dataset model instances are classified in 12 ETC matrices. The twelve ETC matrix instances are formed from three metrics:

resource heterogeneity, task, and consistency. The twelve ETC instance matrices are obtained from the average of hundred ETC matrices of each instance. 12 ETC matrix instances are varied based on metrics heterogeneity, consistency as u-x-yyzz.k. 12 ETC instance matrixes are produced by following uniform distribution (u). Factors that distinguish 12 ETC matrices are: (1) consistency (x), x may be of three different types of consistency are considered for simulation, c-consistent, i-inconsistent, and s-semi-consistent or partially-consistent. (2) task heterogeneity (yy), and (3) resource heterogeneity (zz). yy and zz is either high (hi) or low (lo) based on the task and cloud heterogeneity. The twelve ETC instance matrix is comprised of three classifications of four instances and is shown in Table 6.16. The first, second, and third classification represents consistent, inconsistent, and semi-consistent ETC matrices with high and low combinations of task and resource heterogeneity.

The size of ETC matrix considered for experimental result evaluation is 512*16. 512*16 indicating 512 tasks mapped to 16 clouds. Braun et al.'s benchmark dataset details are listed in Table 6.17 [35].

The description of experimental evaluation steps of proposed SLA-GTMax-Min algorithm is as follows: (1) 12 ETC matrix instances are given as input data for validating proposed SLA-GTMax-Min algorithm. (2) EG and EP matrices are produced. (3) Gain cost (α) and penalty cost (β) are assumed as $\alpha = 2 \times \beta$. (4) SLA level is represented as weight values to represent the level of service required by the customer. (5) Simulation setup is executed for the proposed GTMax-Min algorithm. (6) Makespan, gain cost, and penalty cost are obtained for ETC, EG, and EP matrices, respectively. (7) The average cloud utilization is obtained.

The makespan of the proposed SLA-GTMax-Min algorithm calculated for 512*16, benchmark dataset [35] instances, and compared with the existing algorithm is shown in Fig. 6.1. Figure 6.2 shows the makespan values for high task high cloud heterogeneity. Figure 6.3 shows the makespan values for high task low cloud heterogeneity. Figure 6.4 shows the makespan values for low task high cloud heterogeneity. Figure 6.5 shows the makespan values for low task low cloud heterogeneity.

The average cloud utilization rate of SLA-GTMax-Min calculated for 512*16, benchmark data set instances, and compared with the existing algorithm is shown in Fig. 6.6. Figure 6.7 shows the average cloud utilization rate for high task high cloud heterogeneity. Figure 6.8 shows the average cloud utilization rate for high task low cloud heterogeneity. Figure 6.9 shows the average cloud utilization rate for low task high cloud heterogeneity. Figure 6.10 shows the average cloud utilization rate for low task low cloud heterogeneity.

Table 6.16 ETC model instances

	Heterogeneity			
	Task (high)		Task (low)	
Consistency	Resource (high)	Resource (low)	Resource (high)	Resource (low)
Consistent	u-c-hihi-0	u-c-hilo-0	u-c-lohi-0	u-c-lolo-0
Inconsistent	u-i-hihi-0	u-i-hilo-0	u-i-lohi-0	u-i-lolo-0
Semi-consistent	u-s-hihi-0	u-s-hilo-0	u-s-lohi-0	u-s-lolo-0

Table 6.17 Benchmark dataset descriptions

Benchmark model	Descriptions
Size of the ETC matrix	512 * 16
Number of tasks	512
Number of clouds	16
Number of ETC matrix instances	12
Number of matrices in each instance	100
Three metrics	Task heterogeneity, resource heterogeneity, and consistency
u	Uniform distribution (followed in generating ETC matrix)
x	Consistency (c-consistent, i-inconsistent, and s-semi-consistent or partially consistent)
Consistent ETC matrix	Whenever a resource (cloud) c_i executes any task t_j faster than cloud c_k, then cloud c_i executes all tasks faster than c_k
Inconsistent ETC matrix	Resource (cloud) c_i may be faster than cloud c_k, for executing some tasks and slower for others
Semi-consistent ETC matrix	Includes a consistent sub-matrix
Task heterogeneity	Variation in the execution time for a task t_i on all clouds
yy	Task heterogeneity (hi-high, lo-low)
Resource heterogeneity	Variation in the execution time of all the tasks on a cloud c_k
zz	Resource heterogeneity (hi-high, lo-low)

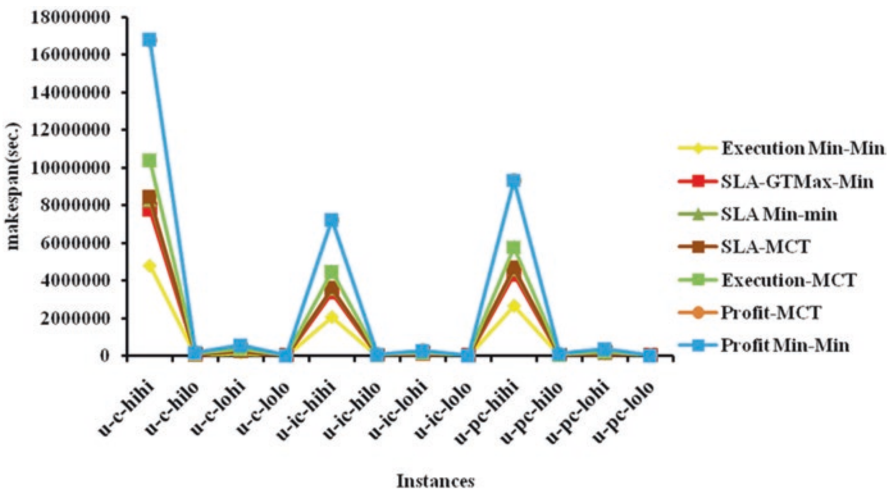


Fig. 6.1 Makespan values

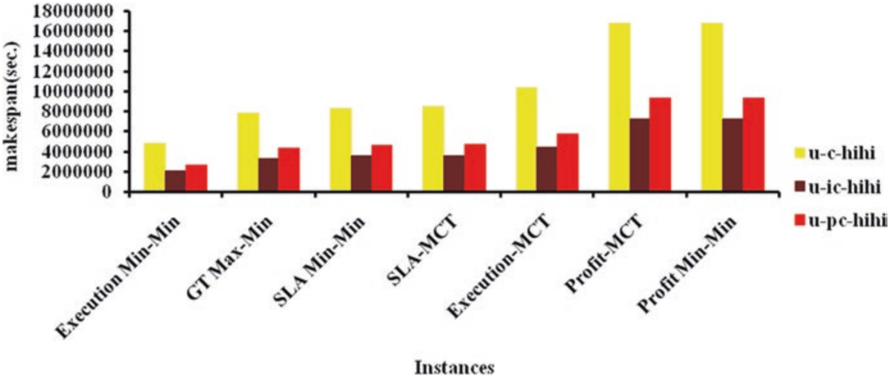


Fig. 6.2 Makespan for high task high cloud heterogeneity

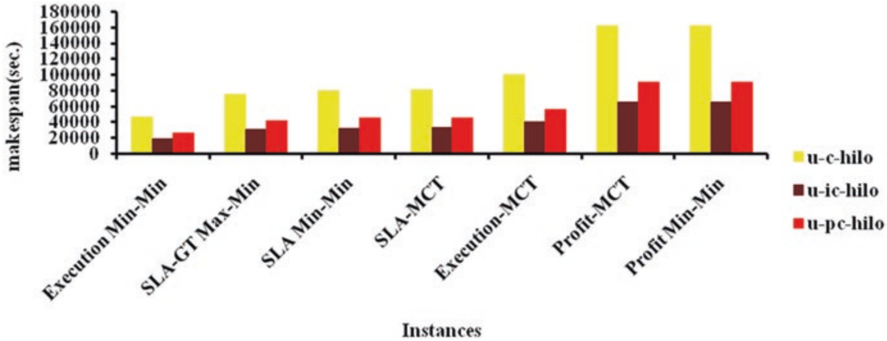


Fig. 6.3 Makespan for high task low cloud heterogeneity

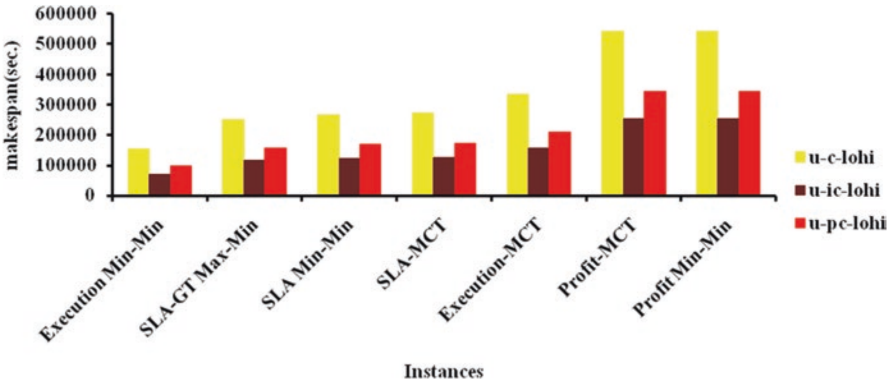


Fig. 6.4 Makespan for low task high cloud heterogeneity

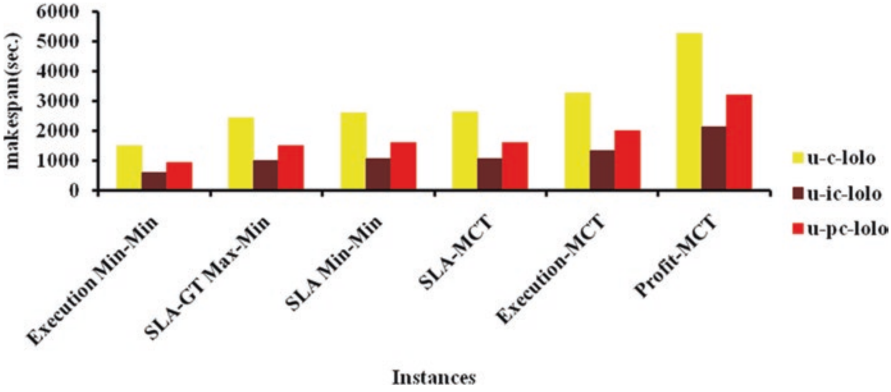


Fig. 6.5 Makespan for low task low cloud heterogeneity

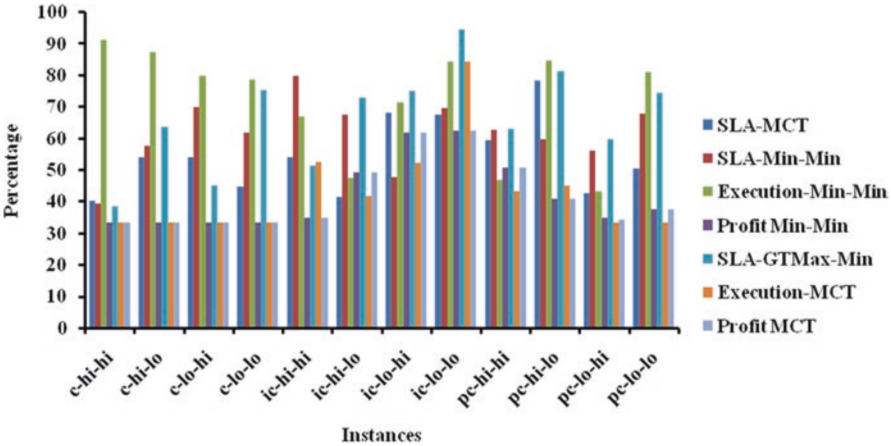


Fig. 6.6 Average cloud utilization rate

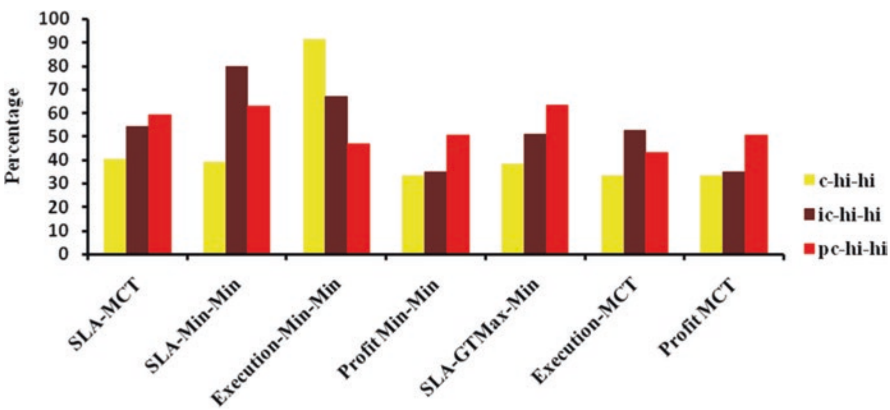


Fig. 6.7 Average cloud utilization rate for high task high cloud heterogeneity

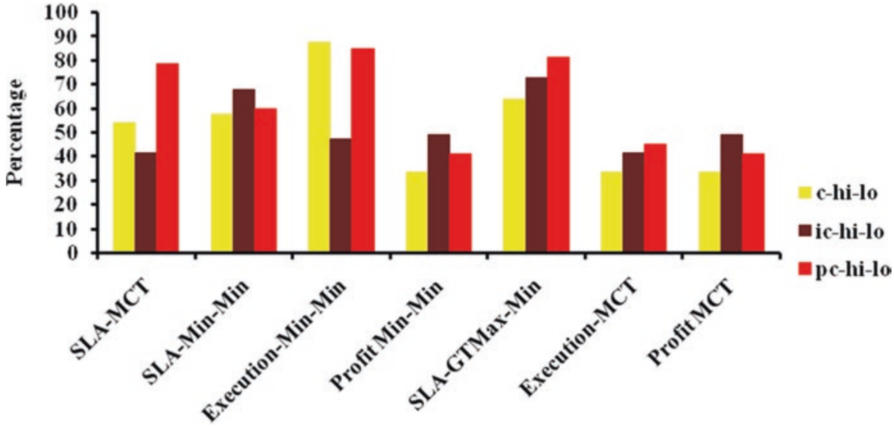


Fig. 6.8 Average cloud utilization rate for high task low cloud heterogeneity

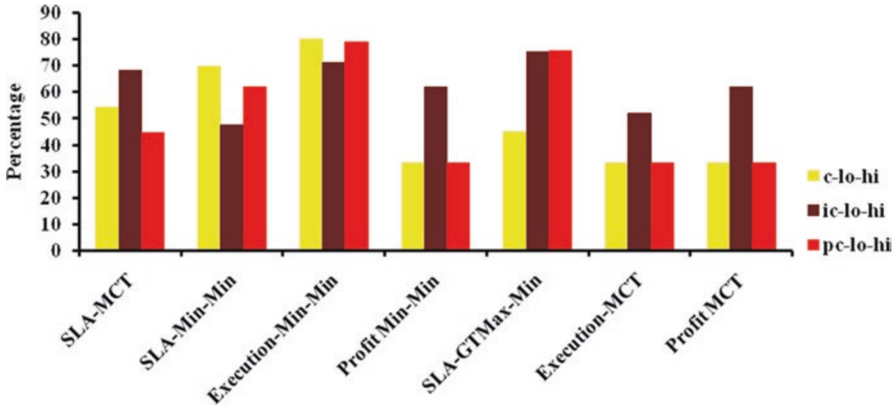


Fig. 6.9 Average cloud utilization rate for low task high cloud heterogeneity

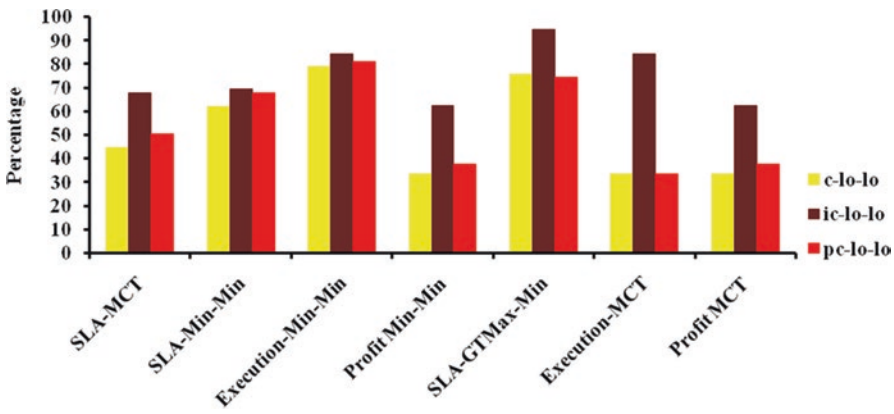


Fig. 6.10 Average cloud utilization rate for low task low cloud heterogeneity

Comparison results depict that proposed SLA-GTMax-Min makes a balance between makespan, penalty, and gain cost compared to existing algorithm.

6.6 Conclusion

SLA-based task scheduling algorithm is proposed. The algorithm is designed to bring out the performance enhancement in task scheduling with minimum makespan and reduced gain cost. To achieve the expected service level of customers, the proposed SLA-GTMax-Min algorithm schedules tasks to cloud based on the customer's SLAs. SLA-GTMax-Min schedules tasks efficiently considering QoS parameters like minimum execution time, minimum gain cost, or the proportion of execution time and gain cost without violating SLA. The proposed SLA-GTMax-Min algorithm gives importance to the customer's SLAs, and the task which requires faster completion scheduled to cloud with minimum completion time, task requiring minimum amount to pay for the service used is scheduled to the cloud with minimum gain cost, task which requires a combination of percentage value of expected completion time and the percentage of amount the customer is ready to pay for the service provided by the CSP is scheduled to the cloud with minimum completion time and gain cost. The proposed algorithm SLA-GTMax-Min provides an efficient balance among makespan, gain, and penalty/violation cost.

References

1. Panda, S. K., Gupta, I., & Jana, P. K. (2015). Allocation-aware task scheduling for heterogeneous multi-cloud systems. In 2nd international symposium on big data and cloud computing challenges (Vol. 50, pp. 176–184). Procedia Computer Science, Elsevier.
2. Abdullahi, M., Ngadi, M. A., & Abdulhamid, S. M. (2016). Symbiotic organism search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems, 56*, 640–650.
3. Durao, F., Carvalho, J. F. S., Fonseka, A., & Garcia, V. C. (2014). A systematic review on cloud computing. *The Journal of Supercomputing, 68*(3), 1321–1346.
4. Liu, L., Mei, H., & XieB. (2016). Towards a multi-QoS human-centric cloud computing load balance resource allocation method. *The Journal of Supercomputing, 72*(7), 2488–2501.
5. Kamalam, G. K., & MuraliBhaskaran, V. (2010). A new heuristic approach: Min-mean algorithm for scheduling Meta mask-task on Heterogeneous computing systems. *IJCSNS International Journal of Computer Science and Network Security, 10*(1).
6. Kamalam, G. K., & MuraliBhaskaran, V. (2010). An improved min-mean heuristic scheduling algorithm for mapping independent tasks on Heterogenous computing environment. *International journal of Computational Cognition, 8*(4).
7. Kamalam, G. K., & MuraliBhaskaran, V. (2012). New enhanced heuristic min-mean scheduling algorithm for scheduling Meta-tasks on heterogeneous grid environment. *European Journal of Scientific Research, 70*(3), 423–430.
8. Wu, F., Wu, Q., & Tan, Y. (2015). Workflow scheduling in cloud: A survey. *The Journal of Supercomputing, 71*(9), 3373–3418.

9. Ibarra, O. H., & Kim, C. E. (1977). Heuristic algorithms for scheduling independent tasks on nonidentical processors. *Journal of the ACM, 24*(2), 280–289.
10. Aazam, M., Huh, E., St-Hilaire, M., Lung, C., & Lambadaris, I. (2016). Cloud customer's historical record based resource pricing. *IEEE Transactions on Parallel and Distributed Systems, 27*(7), 1929–1940.
11. Loo, S. M., & Wells, B. E. (2006). Task scheduling in a finite-resource, reconfigurable hardware/software code sign environment. *INFORMS Journal on Computing, 18*(2), 151–172.
12. Baset, S. A. (2012). Cloud SLAs: Present and future. *ACM SIGOPS Operating Systems Review, 46*, 57–66.
13. Maurer, M., Emeakaroha, V. C., Brandic, I., & Altmann, J. (2012). Cost-benefit analysis of an SLA mapping approach for defining standardized cloud computing goods. *Future Generation Computer System, 28*, 39–47.
14. Gao, Y., Guan, H., Qi, Z., Song, T., Huan, F., & Liu, L. (2014). Service level agreement based energy-efficient resource management in cloud data centers. *Computer and Electrical Engineering, 40*, 1621–1633.
15. Ranaldo, N., & Zimeo, E. (2016). Capacity-driven utility model for service level agreement negotiation of cloud services. *Future Generation Computer System, 55*, 186–199.
16. Ivanovic, D., Carro, M., & Hermenegildo, M. (2011). Constraint-based runtime prediction of SLA violation in service orchestrations. In *9th international conference on service-oriented computing* (pp. 62–76). Springer.
17. Son, S., Jung, G., & Jun, S. C. (2013). An SLA-based cloud computing that facilitates resource allocation in the distributed data centers of a cloud provider. *The Journal of Supercomputing, 64*(2), 606–637.
18. Farokhi, S., Jrad, F., Brandic, I., & Streit, A. (2014). Hierarchical SLA-based service selection for multi-cloud environments. In 4th international conference on cloud computing and services science (pp. 722–734).
19. Garcia, A. G., Espert, I. B., & Garcia, V. H. (2014). SLA-driven dynamic cloud resource management. *Future Generation Computer System, 31*, 1–11.
20. Emeakaroha, V. C., Netto, M. A. S., Calheiros, R. N., Brandic, I., Buyya, R., & Rose, C. A. F. D. (2012). Towards autonomic detection of SLA violations in cloud infrastructures. *Future Generation Computer System, 28*, 1017–1029.
21. Franke, U., & Buschle, M. (2016). Experimental evidence on decision-making in availability service level agreements. *IEEE Transactions on Network and Service Management, 13*(1), 58–70.
22. Lu, K., Yahyapour, R., Wieder, P., Yaqub, E., Abdullah, M., Schloer, B., & Kotsokalis, C. (2016). Fault-tolerant service level agreement lifecycle management in clouds using actor system. *Future Generation Computer System, 54*, 247–259.
23. Abawajy, J., Fudzee, M. F., Hassan, M. M., & Alrubaian, M. (2015). Service level agreement management framework for utility-oriented computing platforms. *The Journal of Supercomputing, 71*(11), 4287–4303.
24. Son, S., Kang, D., Huh, S. P., Kim, W., & Choi, W. (2016). Adaptive trade-off strategy for bargaining-based multi-objective SLA establishment under varying cloud workload. *The Journal of Supercomputing, 72*(4), 1597–1622.
25. Panda, S. K., & Jana, P. K. (2016). *Normalization-based task scheduling algorithms for heterogeneous multicloud environment, information systems frontiers*. Springer.
26. Li, J., Qiu, M., Ming, Z., Quan, G., Qin, X., & Gu, Z. (2012). Online optimization for scheduling preemp table tasks on IaaS cloud system. *Journal of Parallel and Distributed Computing, 72*, 666–677.
27. Freund, R. F., Gherrity, M., Ambrosius, S., Campbell, M., Halderman, M., Hensgen, D., Keith, E., Kidd, T., Kussow, M., Lima, J. D., Mirabile, F., Moore, L., Rust, B., & Siegel, H. J. (1998) Scheduling resources in multiuser, heterogeneous, computing environments with SmartNet. In 7th IEEE heterogeneous computing workshop (pp. 184–199).

28. Cloud Service Level Agreement Standardisation Guidelines. http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?action=display&doc_id=6138. Accessed 4 June 2015.
29. Wang, S., Yan, K., Liao, W., & Wang, S. (2010). Towards a load balancing in a three-level cloud computing network. In 3rd IEEE international conference on computer science and information technology (Vol. 1, pp. 108–113).
30. Panda, S. K., & Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. *The Journal of Supercomputing, 71*(4), 1505–1533.
31. Panda, S. K., & Jana, P. K. (2017). SLA-based task scheduling algorithms for heterogeneous multi-cloud environment. *The Journal of Supercomputing, 73*(6), 2730–2762.
32. Nagarajan, R., Selvamuthukumaran, S., & Thirunavukarasu, R. (2017). A fuzzy logic based trust evaluation model for the selection of cloud services. In IEEE international conference on computer communication and informatics (ICCCI-2017), Coimbatore (pp. 1–5).
33. Nagarajan, R., & Thirunavukarasu, R. (2018). A review on intelligent cloud broker for effective service provisioning in cloud. In Second international conference on intelligent computing and control systems (ICICCS), Madurai, India (pp. 519–524).
34. Rajganesh, N., & Ramkumar, T. (2016). A review on broker based cloud service model. *Journal of Computing and Information Technology, 24*(3), 283–292. The University of Zagreb Computing Centre (SRCE), Croatia, Print-ISSN - 13301136 908.
35. Braun, T. D., Siegel, H. J., Beck, N., Boloni, L. L., Maheswaran, M., Reuther, A. I., Robertson, J. P., Theys, M. D., Yao, B., Hensgen, D., & Freund, R. F. (2001). A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel Distributed Computing, 61*(6), 810–837.
36. Demiroz, B., & Topcuoglu, H. R. (2006). Static task scheduling with a unified objective on time and resource domains. *The Computer Journal, 49*(6), 731–743.
37. Xhafa, F., Carretero, J., Barolli, L., & Durresi, A. (2007). Immediate mode scheduling in grid systems. *International Journal of Web and Grid Services, 3*(2), 219–236.
38. Xhafa, F., Barolli, L., & Durresi, A. (2007). Batch mode scheduling in grid systems. *International Journal of Web and Grid Services, 3*(1), 19–37.

Chapter 7
Workload Balancing in a Multi-Cloud Environment: Challenges and Research Directions

B. S. Rajeshwari, M. Dakshayini, and H. S. Guruprasad

7.1 Introduction

This section comprises of the need for multi-cloud environment, working functionality of multi-cloud, as well as architecture of multi-cloud. Presently, the users are progressing toward dynamic cloud services alternate to static, on-premise infrastructure due to numerous advantages in dynamic cloud services. The need for various cloud offerings aroused due to the trending business requirements with each focused on a specific set of services, viz., enterprise applications in Azure, machine learning in Google Cloud, enterprise resource planning in Oracle Cloud, and expanding variety of services in Amazon Web Services (AWS) [1]. This issue can be addressed using the concept of multi-cloud.

Multi-cloud is a next generation cloud, where business firms are moving toward the use of services from multiple clouds. The deployment of multi-cloud entails various cloud providers, wherein each cloud provider operates his own standalone and relatively independent cloud. Multi-cloud is the future for business firms and companies. The emerging technology has made many organizations to rely on the cloud to deploy and manage application stack. As a matter of fact, most of the organizations support several cloud-based applications [2]. The hosting in cloud minimizes initial infrastructure expenses, operational costs, as well as maintenance expenses. On one side, adoption of emerging multi-cloud computing technology

B. S. Rajeshwari (✉)
Department of Computer Science and Engineering, B.M.S College of Engineering, Bangalore, India

M. Dakshayini · H. S. Guruprasad
Department of Information Science and Engineering, B.M.S College of Engineering, Bangalore, India
e-mail: dakshayini.ise@bmsce.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_7

creates a lot of opportunities with numerous benefits for the organizations; on the other side, there are various challenges and difficulties while adopting multi-cloud architecture.

The multiple clouds are represented in various forms like hybrid cloud, federated cloud, and multi-cloud.

- **Hybrid Cloud:** This is a collaboration of private cloud and public cloud. Generally, the workloads will be divided over the cloud and on-premise, and these workloads belong to a single application, meaning the piece in the cloud and the piece on-premise interact to do something useful [3]. Multiple clouds incorporate federated cloud delivery model and multi-cloud delivery model [4]. These two deliver models can be distinguished based on the level of collaborations among the multiple clouds.
- **Federated Cloud:** In this federated cloud model, a group of cloud service providers associate together to provide seemingly infinite computing resources and services. This cloud architecture is administered by a federated cloud broker (FCB), whose responsibility is to manage, monitor, coordinate, and dispense services among multiple clouds. Initially, each cloud service provider makes an agreement with the federated cloud broker in sharing their resources to the federated cloud environment.
- **Multi-Cloud:** A multi-cloud environment is one wherein user uses two or more cloud platforms and each platform delivers a specific service or an application. It is a combination of several clouds that contains public clouds as well as in many cases private clouds. It pays more attention on cloud services rather than on specific deployment models. In this model, the end user will be aware of services from the different clouds.

A multi-cloud architecture empowers resource allocation, management, and monitoring among multiple cloud service providers. It makes cloud users to take benefit of economical parameter or locality by allowing them to create and access resources on multiple clouds without compromising its quality of service (QoS).

The multi-cloud system achieves higher levels of usability as well as locality benefits by tackling cross-cloud interactions and integration of multiple clouds. With the multi-cloud environment, an organization could choose to store user data on cloud, leveraging one cloud provider for infrastructure service and another cloud provider for software service. The locality can also be augmented by deploying the services required by user in the cloud data center which is near to user.

7.1.1 Need for Multi-Cloud

The services and resources from multiple clouds are very much essential for various reasons [4] as follows:

- To enhance quality of services along with optimizing its cost by choosing different services from different cloud service providers
- To follow the constraints, like new locations or country laws
- To replicate the applications or services among different cloud service providers in order to ensure high availability
- To avoid dependence on single cloud service provider
- To utilize different services from different cloud service providers as per the requirements

7.1.2 Architecture of Multi-Cloud and Working Functionality of Multi-Cloud

The architecture of multi-cloud environment is depicted as shown in Fig. 7.1. Multi-cloud architecture enables organizations to distribute their workloads across multiple cloud data centers with the profitable deal for each provider while lessening risks associated with an individual cloud. It provides enormous agility as well as cost efficiency by taking different services from different clouds based on their requirement specifications. A multi-cloud strategy empowers organizations to make best use of breed cloud services from the abundant cloud service providers, for example, compute service from Amazon Web Service, container services from Google cloud, and software services from SAP.

While adopting several cloud services from single cloud provider, the decision-making policy has to be implemented effectively. It is difficult to find the suitable services from the multiple cloud providers. Thus, a cloud broker is introduced who act as a mediator between the provider and the customer for service discovery. A Cloud Service Broker (CSB) [5, 6] is the one who interacts with multiple cloud service providers to satisfy the requirements of the user with the available cloud resources and services. CSB is mainly used to support inter-cloud in multi-cloud environments. The main role of this broker is to improve cloud services and to ensure security in cloud services. The users depend on these brokers to make use of publicly available cloud services in full potential, with respect to costs, quality, and flexibility. Intelligent service brokering in selecting the suitable service from multiple cloud providers can be done by soft computing techniques such as fuzzy logic.

Cloud service broker in a multi-cloud environment typically involves in service life cycle management, comparing, customizing, aggregating, negotiating, managing, and administering of cloud services from multiple clouds. Cloud service brokers in multi-cloud environment can be categorized into three types:

- **Cloud Aggregators:** Cloud aggregators provide a single user interface to users in a multi-cloud ecosystem by combining service catalogs from multiple cloud service providers in a unified way. The CSBs function primarily as re-sellers in

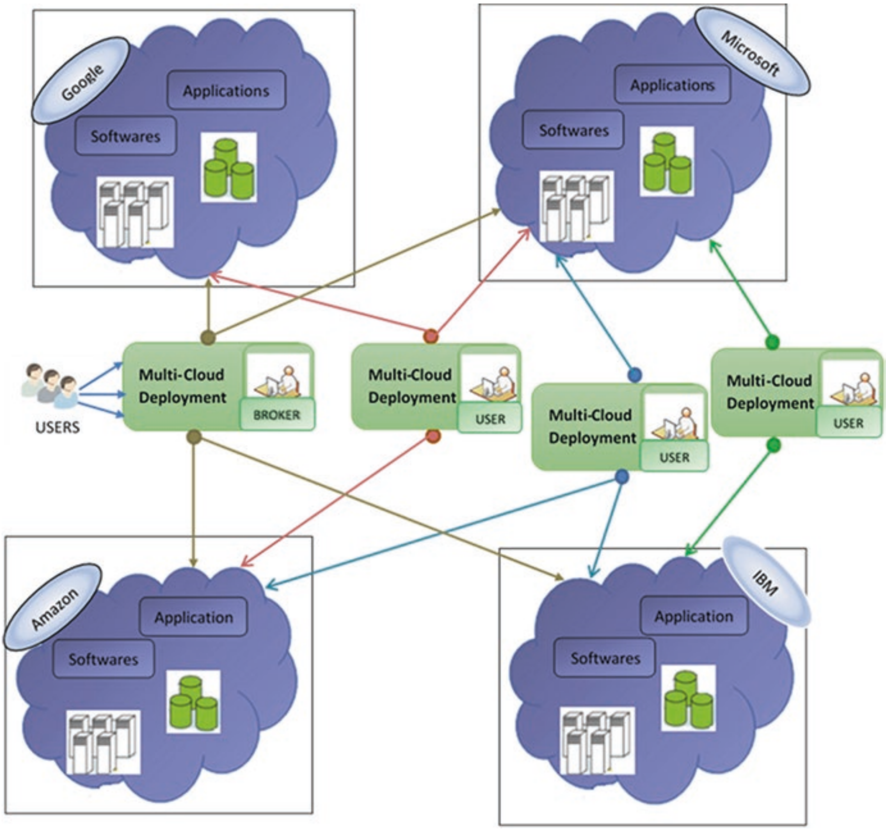


Fig. 7.1 Architecture of multi-cloud environment

the multi-cloud environment, but further most brokers offer additional services like security and governance compliance. As CSBs are mediators between cloud service providers and users, they will take care of negotiation between users and cloud service providers.

- **Cloud Integrators:** Cloud integrators try to improvise and optimize multi-cloud functionality by integrating various multi-cloud services.
- **Cloud Customizers:** Cloud customizers are responsible for the fulfillment of specific functional requirements, modification of the capabilities of the existing cloud services, and offering these modified services.

The common patterns that can be identified in multi-cloud architecture are categorized as follows:

- Firstly, a pattern is formulated on distributed deployments of applications: This pattern aims at providing the computing environment that is ideal for application requirement specifications. It combines several public cloud environments of various cloud providers and enables the deployment of applications in the best possible environment.

- For example, Application X executes in Google cloud platform and application Y executes in Azure platform.
- Secondly, a pattern is formulated on redundant deployments of applications: This pattern aims to increase scalability or resilience by deploying the identical application in several computing environments. So, this pattern provides more availability beyond what individual cloud provider provides.

7.1.2.1 Benefits of Multi-Cloud

The various benefits of using multi-cloud technology [7, 8] are as follows:

- **Cost-effective service with optimum satisfaction:** Organizations can meet their business, security, and performance needs in a cost-effective way according to their needs and budget using cloud solutions at varying prices with multiple providers.
- **One out of many alternatives to choose the best in Class Services:** The service capability of cloud service providers will differ based on their abilities. The capabilities like services, performance, and cost vary significantly. Usually, no individual cloud service provider is superior in each and every category. Multi-cloud allows organizations to choose best among several cloud services that satisfies their business needs.
- **Availability:** The applications can be distributed across multiple clouds, wherein everyone offers high availability in their service level agreement. Companies can especially reduce the risk of simultaneous downtime across all the clouds.
- **Avoiding Vendor Lock-in:** One of the most important benefit of multi-cloud architecture is avoiding vendor lock-in. This feature makes businesses to surpass the cloud they are using when they find other providers offering better deal. So, it allows users to frame their own infrastructure suitable to their company goals by opting better services from individual cloud platforms. Using this benefit, organization can check out with several cloud service providers that satisfy their business specifications instead of compromising with an individual provider's setup and execution.
- **Data Privacy and Compliance:** The data privacy is one of the important parameter to be considered in the cloud environment. Multi-cloud environment offers flexible data storage options for organizations finding hard with issues like data sovereignty and privacy compliance.
- **Proximity/Low Latency:** The multi-cloud has low latency time to its users in contrast to individual cloud, as an application deployment spans over multiple clouds. When application users are scattered in various parts of globe and

transmission of data is being done in a sole data center, the response time for users will be more. This drawback can be overcome in multi-cloud environment, by deploying services in more than one region based on user locations. Here, the cloud architect designs in such a way that the nearby datacenter provides requested data to users with least server hops. So, it benefits more for worldwide organizations which needs to maintain a unified end-user experience and serves data across geographically spread locations.

- **Disaster Recovery:** The company will be in risk if it depends on one cloud platform to get all its resources and services. There may be unplanned outages, service disruptions, and downtime even in cloud which causes to halt all the operations for prolong time and makes end user not able to access until it sorts out the problem. Here, in multi-cloud environment, these types of downtime in one cloud can be replaced by other clouds which are available to take these workloads.
- **Data Management:** Cloud data management is a method of managing data across cloud platforms. The companies generate different types of data. The data can be differentiated like cold storage (data accessed rarely) and hot data (accessed regularly like 10 times a day) and need to be stored in frequently accessed storage. The multi cloud gives an opportunity to take benefit of the right service for the right function instead of storing all data in a single cloud.

7.2 Workload Balancing Among Multiple Clouds and Effect on Performance Parameters in a Multi-Cloud Environment

In this section, the workload distribution among multiple clouds is discussed. Also, balancing of workload among multiple clouds and influence on performance parameters like waiting time, response time, service level agreement violations, resource utilization, and power utilization are elaborated.

In the multi-cloud scenario, the user requests will be distributed among cloud data centers through a common entry point. Inter-cloud deployment architecture is taken into consideration, in case an application comprises multiple management domains. This multi-cloud architecture ensures high data availability and enables balanced distribution of load, efficient management of resources [9].

As cloud advances to this new technology, challenges are associated in dealing with issues like scale, reallocation, and performance and also geo-distribution of multiple resources in several data center [10]. However, in the multi-cloud scenario [11], the load balancing and failover mechanisms are the main challenges.

Nikolay Grozev et al. [12] emphasizes on distribution of workload across clouds. Each cloud data centers in a multi-cloud environment may have diverse workload.

For instance, a cloud data center in the UK will be heavily loaded during the peak hours and substantially less loaded, otherwise. The load balancer is a separate component and is not installed within any of the virtual machine. This load balancer distributes the arriving requests to the application servers that has minimum utilization of CPU.

Tordsson et al. [13] presented an integer programming formulation to resolve resource provisioning problems in multi-cloud environment. The presented model, by limiting the number of virtual machine type instances, tried to balance the load among all the virtual machines irrespective of cloud locations, with existing typical hardware configuration limitations.

Lucas Simarro et al. [14] presented an architecture that takes care of optimal deployment of services in dynamic pricing multi-cloud environment. The presented architecture includes broker component. Among the various components of the broker, the scheduling module is responsible for deploying services based on criteria like cost optimization, performance optimization, budget, and load balancing constraints.

Vamis Xhagjika [10] focuses mainly on the allocation of resources in real-time application optimizing for latency with new multi-cloud architecture. The users in multi-cloud are scattered worldwide and need to communicate with one another either with audio or video, especially in conference or live streaming applications. A user with farther network proximity to the data center will suffer from quality, whereas user with nearby distance enjoys good quality using centralized solution. These drawbacks can be overcome by *Inter-Cloud deployment*, which provides availability of resources to all end users uniformly irrespective of the locality and the user's location, as well as by optimizing the cost. The authors explored an architecture based on the performance parameters of locality aware services for optimizing selection of resources.

Seungmin Kang et al. [15] introduced a novel distributed scheduling approach in multi-cloud scenario which accepts the request and schedules the required resources in the gateway. But the integrity among various schedulers in each gateway besides maintaining the quality of being consistent with respect to information should be maintained in this distributed scheduling approach. The designed scheduling algorithm schedules dynamically arriving loads across multi-cloud platforms.

N. Grozev et al. [16] presented a technique for resource allocation and management and also optimal distribution of workload across multiple clouds in a multi-cloud environment.

Rajganesh Nagarajan et al. [17] presented intelligent cloud broker model that uses fuzzy ontology to select optimal cloud among multiple clouds by defining the rank. The same authors [18] discuss about the service computing from the perspective of two architectural models: service-oriented architecture and cloud computing.

7.3 Current Challenges and Research Direction Toward the Multi-Cloud

This section covers the current challenges in multi-cloud like application deployment in cloud, services from multiple cloud providers, balanced distribution of workload, processing of large-scale data, and security issues in multi-cloud environment. Also, research directions toward these challenges are highlighted and discussed.

7.3.1 Challenges in Multi-Cloud Architecture

Multi-cloud architecture allows organizations to run their applications and store data across multiple clouds. The multi-cloud environment has its own advantages and also specific challenges that organizations need to consider.

The challenges in the multi-cloud environment are depicted in Fig. 7.2. Addressing these challenges needs the development of new technologies for unified multi-cloud management. In addition to efficient resource management, effective performance management, optimized power consumption, and costs of private cloud as well as public cloud services, multi-cloud management platforms must take care of both applications running on premises as well as across various multiple clouds and interoperating with the multi-cloud architecture. Cloud services can lag and cloud services cost can escalate out of control, if multi-cloud environment is not managed in a right way due to its complex nature.

Fig. 7.2 Challenges in multi-cloud environment

Workload Portability: When an organization need to move their workloads across multiple clouds, workload portability becomes a key requirement. Interoperability in multi-cloud environment is the capability of supporting applications and services running across multiple public clouds and private clouds associated with diverse aspects of cloud services, data formats, authentication as well as authorization processes, etc. Thus, workload portability in multi-cloud environment is a challenging issue.

Security in Multi-Cloud Architecture: Security is one of the important concern for an organization, if they want to move their workload and services out of the organization across multiple clouds. Using a multi-cloud platform has various security issues. Users need to ensure securing of data and applications which are distributed across multiple clouds. User can integrate automated security policies into the multi-cloud management platform. But when user uses multiple security tools from different cloud service providers, then security environment becomes fragmented and security teams need to take care of manually correlating data in order to implement security protections. This human intervention may lead to increased error, leaving organizations exposed to threats as well as data breaches. Thus, an automated security policy needs to be integrated into multi-cloud management platform.

Another security challenge includes managing trust among the clouds and establishing a common identity between multiple clouds so that multi-cloud systems can authenticate securely across environment boundaries. This forces to develop consistent identity management and authentication system. Also, cross cloud data transfers in a multi-cloud architecture may lead to serious security attacks which need to be taken care.

Complexity in Managing and Monitoring Multiple Clouds: There are several challenges in adopting multi-cloud architecture, managing, and monitoring services at multiple clouds. The main aim of the new cloud management technologies to deal with these challenges is providing a unified management platform with nominal management overhead in monitoring and managing workloads and policies across multiple clouds [19]. Unfortunately, these tools can only monitor and manage workloads running on that cloud provider's infrastructure. But, if organizations distributed their workload across multiple clouds and need to monitor and manage their workloads across multiple cloud providers, then they have to use multiple tools to get the job done, which is very challenging. Also, lack of application management across multiple clouds in a multi-cloud environment makes it easy to lose track of which applications are running, where, and how much it costs every day. Thus, a unified management platform that takes of monitoring and managing workloads and policies across multiple clouds is very much needed for multi-cloud environment.

Platform Integration for Multi-Cloud: In multi-cloud environment, the information moves from one cloud platform to another. This requires an organization to have an expertise in multiple cloud providers and complex cloud management, which is challenging. Thus, a platform integration tool for multi-cloud environment that takes care of workload movement from one cloud platform to another cloud platform is very much needed. The development toolchain needs to be unified for agile development across different platforms in multiple clouds of multi-cloud environment.

Financial Management in Multi-Cloud Environment: When users adopt services from multiple clouds, complexity of multi-cloud architecture reflected in understanding multi-cloud costs and billing which is a new challenge for the users. Since the multi-cloud technology is relatively new, approaches to multi-cloud cost management and billing management are still in development.

Ensuring Governance and Compliance Issues: In multi-cloud environment, users are widely distributed across the globe. For example, an organization can use Google Cloud for users in North America and Microsoft Azure to serve the users in Europe. Each country has their own regulatory requirements for data storage within the country. Thus, country-specific compliance requirements need to be followed. Ensuring governance policies and compliance could be tricky and challenging.

Spiraling Costs: Because of the flexibility and scalability with multi-cloud approach, organizations decide to adopt multi-cloud strategies. However, the advantage of the cloud also turns out to be one of its biggest disadvantages. Cloud environment keeps changing, and the costs associated with each service among multiple cloud providers can quickly spin out of control. Users need to ensure and take care of costs associated with each service across multiple clouds in multi-cloud environment, which is very challenging.

Availability of Skilled Staff: The rapidly changing cloud technology makes finding the staff with multi cloud skills a challenge [19, 20]. When an organization required to deploy a new multi-platform monitoring tool, they often have to train the staff, which is tricky for an organization.

Network Bandwidth: Network bandwidth is another major challenging issue with the multi-cloud environment [21]. Thus, network bandwidth and latency rates need to be taken care when operating with multi-cloud environment.

Scaling Workloads across Multiple Clouds: Automating the scaling up and scaling down of resources and load balancing across the multiple clouds is another challenge.

In addition to the main challenges on multi-cloud environment discussed above, still some additional challenges exist related to optimal service from the multi-cloud environment as follows:

- Monitoring and guaranteeing SLA performance in multi-cloud environment
- Automatic migration of load across cloud providers
- Automatic discovery of service and service composition
- Proprietary APIs and lack of interoperability
- Optimal power management in servicing the load

7.4 Simulators to Solve Research Challenges in Multi-Cloud Management

This part of chapter lists the pertained management platforms and simulators in multi-cloud to resolve research challenges and briefs to resolve research challenges with the use of simulators.

7.4.1 Multi-Cloud Management Platforms Tools

The various multi-cloud management tools available are listed and briefed as follows [1]:

- **Dell Multi Cloud Manager**: It is a cloud-agnostic automated cloud management solution including a wide range of public and private cloud platforms and incorporates many features like a self-service portal, automated provisioning, application and infrastructure templates, scaling, governance, security, monitoring, and integrations. This management platform is available either as software as a service subscription basis or as on-prem deployment.
- **BMC Multi Cloud Management:** This tool provisions an IT services over multiple cloud platforms in an automated way. It deploys several service management processes including change management, configuration management, compliance, and patching by allowing governance and compliance controls for cloud workloads. This tool in multi-cloud platform provides an association of on-prem and distributed cloud resources besides providing a view of those resources.
- **Embotics vCommander:** The several cloud environments are supported by this tool by providing automated provisioning as well as self-service abilities.
- **Flexera Multi-Cloud Platform:** This tool comprises a self-service portal that allows developers to access private as well as public cloud infrastructures and assists in automatic cloud application deployment. Besides, it also supports monitoring, forecasting, and optimizing costs in a multi-cloud platform.
- **Nutanix Enterprise Cloud Platform:** The technology-agnostic architecture of this tool allows IT teams to organize their applications in multi-cloud platforms. This tool includes multi-cloud cost optimization service. It analyzes cloud consumption pattern details which outline cost optimization in multi-cloud environments.
- **Red Hat CloudForms:** This tool allows to setup policy-controlled, self-service multi-cloud environments. It provides a compatible cross-platform experience by offering uniform management for hybrid environments.
- **Scalr Cloud Management Platform:** This is an open source cloud-agnostic management solution that offers a wide variety of functionalities in cloud management principal fields like governance, security, and compliance, business

agility, cost optimization, and visibility. It empowers companies to organize, automate, and constraint multi-cloud platforms by using an administrative console with an individual user interface and API standardizing several clouds to arrange and workload deployments automatically across multi-cloud environment.

- **VMware vRealize Suite:** This suite allows company to widen the cloud environment available by putting several public clouds together. An application lifecycle management, which has been recently added, aids DevOps in developing applications in multi-cloud. It can organize applications and infrastructure over various clouds and provisions an IT resources in an optimal way.

7.4.2 Simulation Tool

In the real world, it is difficult for researchers to set up a physical cloud with hundreds and thousands of virtual machines in place to test real-time experiments and implement their novel methodologies and algorithms. It becomes very costly to conduct repeated experiments for evaluation under different scenarios. Thus, simulation tool is very much necessary to evaluate their newly designed methodologies and algorithms in the domain of cloud computing. Researchers have to use a cloud simulator to evaluate new algorithm, to do in-depth analysis of the proposed research work, and to measure the overall performance and quality of the proposed work. It is important for a researcher to get a right simulation tool based on the experimentation scenario and the features supported by the simulation tool [13]. Few simulators are open source and free, while others are paid.

There are several good simulation tools to simulate workloads running on a single-provider cloud system, but only limited tools available to simulate workloads running on multiple clouds. Hence, it is necessary to design and develop a simulation framework for multi-cloud environment. The various simulators available to solve research challenges in multi-cloud environment are discussed further.

7.4.2.1 CloudSim

The implementation on the CloudSim simulator can be done using Java. CloudSim is widely using by research scholars, academicians, as well as cloud specialists in the industries for the simulation of their newly designed cloud-based algorithms and methodologies. It provides a platform to create a data center model, brokers, scheduling, and allocation policies inside the cloud with many data centers [22, 23]. It is a complex toolkit using which desired cloud scenarios can be designed either by extending the classes or replacing the classes. The limitation of CloudSim simulator is that it does not support a graphical user interface.

7.4.2.2 CloudReports

CloudReport is designed upon CloudSim simulator with several enhancements [24]. The advanced features of Cloud Report are running parallel simulation runs, an effective graphical interface, and enhanced simulation results. A cloud environment with many data centers can be modeled using Cloud Report with customization as per the requirements. The user can easily make resource configuration settings like processing capacity, RAM capacity, storage capacity, bandwidth, number of cloudlets, etc. After the completion of simulation completes, it generates the results in the form of HTML reports representing execution time and resource utilization power consumption of each data center.

7.4.2.3 Cloud Analyst

Cloud Analyst is a graphical simulation toolkit which is created upon the CloudSim toolkit. Cloud Analyst supports for routing the Internet traffic among the data centers and user bases situated in different geographic locations [25]. CloudAnalyst is a well-known simulation framework for modeling real-time cloud data centers and experimenting the algorithms and strategies on it. The results of experimentations with lot of data can be graphically represented using Cloud Analyst.

7.4.2.4 CloudSME

CloudSME simulation platform [26] can be used to simulate multi-cloud scenarios using technologies like WS-PGRADE/gUSE gateway framework, the Cloud Broker Interface, and the corresponding AppCenter. This platform supports the simulation of customized cloud environment and scenarios in the form of SaaS-based delivery.

7.4.2.5 MDCSim

MDCSim is a commercial scalable and flexible simulation platform [27, 28]. It supports the simulation of hardware features of the servers, communication, and network infrastructure of the data centers. The power utilization by these components can be estimated using this platform. MDCSim also supports the detailed analysis of multiple layers of the data centers [19]. It supports the implementation of design specifics of the networking and communication infrastructure, scheduling at the kernel level, and the various interactions among these layers.

7.4.2.6 DCSim

DCSim simulation tool [29] develops data center management technique and concentrates toward creating virtualized data centers. DCSim simulation tool emphases on both transactional and continuous workloads. This simulator allows researchers to model dependencies among the virtual machines which belongs to a multitiered application [30]. Using this DCSim, researcher can also simulate replicated virtual machines sharing incoming workload. With this DCSim, researcher can easily measure the SLA achievement and evaluate power utilization in the servers of a multi-tier cloud.

7.4.2.7 SimIC

SimIC [31] is a discrete event simulation tool. It is built upon SimJava, which is a process-oriented simulation package. SimIC supports the simulation of multiple clouds by collaborating with each other. SimIC [32] allows effective interaction between the cloud data centers for increasing the quality of service. SimIC supports static and dynamic service level matching policies among the metabrokers, static and dynamic virtual machine creation based on history recirds, preemptive as well as non-preemptive schedulings, and virtual machine migrations as per the provider's requirement.

Nikolay Grozev et al. [12] designed a simulation framework which is an extension of the CloudSim simulator that support interactive three tier applications.

Leonard Heilig et al. [33] designed a simulation framework which is an extension of CloudSim, implementing brokering schemes and an effective selection of virtual machines in multi-cloud environment. The designed framework uses greedy technique to support cloud management functionalities.

7.5 Conclusion

Multi-cloud is the next-generation cloud technology where user uses more than one cloud platform and each one of them delivers a specific application or service. Multi-cloud architecture allows organizations to run their applications and store data across multiple clouds. The multi-cloud environment has its own advantages and also specific challenges that organizations need to consider. The multi-cloud architecture, working functionality of multi-cloud, and managing multi-cloud environment such as distribution of load among multiple clouds are discussed in detail. Also, the current challenges in multi-cloud as well as simulation tools available for researcher in multi-cloud environment and applicability of simulators are elaborated. This helps the researchers in finding out the suitable simulation tool for solving different research challenges.

References

1. https://education.dellemc.com/content/dam/dell-emc/documents/en-us/2019KS_Gloukhovtsev-Why_You_Need_Multicloud_and__How_to_Manage_It.pdf.
2. https://resources.idg.com/download/executive-summary/cloud-computing-2018.
3. https://architectelevator.com/cloud/hybrid-Multi Cloud/.
4. Petcu, D. (2013). Multi cloud: Expectations and current approaches. International workshop on multi cloud applications and federated clouds, Prague, Czech Republic, April 2013 (pp. 1–6). https://doi.org/10.1145/2462326.2462328.
5. Jyoti A, Shrimali M, Tiwari S, & Singh, H. P. (2020). Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey. *Journal of Ambient Intelligence and Humanized Computing*, 1–30.
6. Hentschel R, Strahringer, S. (2020, April). A broker-based framework for the recommendation of cloud services: A research proposal. In Conference on e-business, e-services and e-society (pp. 409-415). Springer, .
7. https://www.opensourceforu.com/2016/11/best-open-source-cloud-computing-simulators/.
8. https://www.simform.com/Multi Cloud-architecture/.
9. Alshammari, M. M., Alwan, A. A., Nordin, A., & Al-Shaikhli, I. F. (2018) Disaster recovery in single-cloud and multi cloud environments: Issues and challenges. 4th IEEE international conference on engineering technologies and applied sciences, https://doi.org/10.1109/ICETAS.2017.8277868.
10. https://kth.diva-portal.org/smash/get/diva2:1090952/FULLTEXT01.pdf.
11. Moreno Vozmediano, R., Montero, R. S., Huedo, E., & Llorente, I. M. (2018). Orchestrating the deployment of high availability services on multi-zone and multi cloud scenarios. *Journal of Grid Computing, 16*, 39–53. Springer.
12. Grozev, N., & Buyya, R. (2015). Performance Modelling and simulation of three-tier applications in cloud and multi cloud environments. *The Computer Journal, 58*(1), 1–22. https://doi.org/10.1093/comjnl/bxt107
13. Tordssona, J., Montero, R. S., Moreno-Vozmediano, R., & Llorente, I. M. (2012). Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Computer Systems, 28*(2), 358–367.
14. LuisLucas-Simarro, J., Moreno-Vozmediano, R., Montero, R. S., & Llorente, I. M. (2013). Scheduling strategies for optimal service deployment across multiple clouds, future generation computer systems. *ScienceDirect, 29*(6), 1431–1441.
15. Kang, S., Veeravalli, B., & Aung, K. M. (2018). Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi cloud systems. *Journal of Parallel and Distributed Computing, 113*, 1–16. Elsevier.
16. Grozev, N., & Buyya, R. (2014) Multi cloud provisioning and load distribution for three tier applications, *ACM Transactions on Autonomous and Adaptive Systems*, (13).
17. Nagarajan, R., & Thirunavukarasu, R. (2018). A review on intelligent cloud broker for effective service provisioning in cloud. Proceedings of the second international conference on intelligent computing and control systems IEEE xplore compliant part number: CFP18K74-ART, ISBN:978-1-5386-2842-3 (pp. 519–524).
18. Nagarajan, R., & Thirunavukarasu, R. (2020) Service-oriented broker for effective provisioning of cloud services - A survey. *International Journal of Computing and Digital Systems*, (5), 863–879. ISSN: 2210-142X.
19. https://www.actualtech.io/5-big-challenges-of-Multi Cloud-management/.
20. https://www.networkcomputing.com/cloud-infrastructure/top-5-challenges-monitoring-multi-cloud-environments.
21. https://innovationatwork.ieee.org/the-Multi Cloud-challenges-and-solutions/.
22. Bahwaireth, K., Tawalbeh, L., Benkhelifa, E, Jararweh, Y, & Tawalbeh, M. A. (2016) Experimental comparison of simulation tools for efficient cloud and mobile cloud computing

applications. *EURASIP Journal on Information Security*. https://doi.org/10.1186/s13635-016-0039-y.
23. Buyya, R., Ranjan, R., & Calheiros, R. N. (2009). Modeling and simulation of scalable cloud computing environments and the CloudSim toolkit: Challenges and opportunities. IEEE international conference on high performance computing and simulation, Germany. https://doi.org/10.1109/HPCSIM.2009.5192685.
24. http://cloudsim-setup.blogspot.com/2013/01/cloud-report-cloudsim-seulator.html.
25. Sharma, A., & Sharma, S. (2016) A comprehensive study of cloud analyst tool and its various algorithms. *International Journal of Engineering Sciences and Management Research*, (6), 1–7, DOI: https://doi.org/10.5281/zenodo.55186.
26. Taylor, S. J. E., TamasKiss, A. A., Terstyanszky, G., Kacsuk, P., & JorisCostes, N. F. (2018). The CloudSME simulation platform and its applications: A generic multi cloud platform for developing and executing commercial cloud-based simulations. *Future Generation Computer Systems, 88*, 524–539.
27. Pandey, R., & Gonnade, S. (2014). Comparative study of simulation tools in cloud. *International Journal of Scientific and Engineering Research, 5*(5).
28. Lim S H, Sharma B, Nam G, Kim E K, & Das C R (2009) MDCSim: A multi-tier data center simulation platform. IEEE international conference on cluster computing and workshops (pp. 1–9). https://doi.org/10.1109/CLUSTR.2009.5289159.
29. Kalpana, E., & Rama Dev, Y. (2015). A study on cloud simulation tools. *International Journal of Computer Applications, 115*(14), 18–21.
30. Jyoti Reddy, S., & Sofia, S. K. (2017). A novel method for data hosting and load balancing in multi cloud environment. *International Journal of Engineering and Computer Science, 6*(10), 22817–22822.
31. Sotiriadis, S., Bessis, N, & Antonopoulos, N. (2013). Towards inter-cloud simulation performance analysis: Exploring service-oriented benchmarks of clouds in SimIC. 27th IEEE international conference on advanced information networking and applications workshops, Spain. https://doi.org/10.1109/WAINA.2013.196.
32. https://www.startertutorials.com/blog/cloud-simulators.html.
33. Heilig, L., Buyya, R, & Voß, S. (2017). Location-aware brokering for consumers in multi cloud computing environments. *Journal of Network and Computer Applications*, 79–93.

Chapter 8
Mediator-Based Effective Resource Allotment on Multi-Clouds

G. Sumathi and S. Rajesh

8.1 Introduction

The fast advancement of distributed computing innovations carries another degree of productivity to services presenting in Internet. With virtualization innovation, CSPs give computational resources (e.g., memory, CPU, and capacity) to meet prerequisites of end clients powerfully with pay-per-utilize model. However, the development and vacillation of clients' solicitations makes it difficult to give sufficient resource and satisfy the Service Agreement Level (SLA). Federation of Cloud has been presented as another worldview of distributed computing that permits various CSPs to divide resources among federation individuals [1, 2]. When a cloud encounters an explosion of incoming jobs, it might allocate some of jobs onto other cloud server farms with inactive resources, which empower load balance between various clouds. To execute the federated cloud worldview, critical interest has emerged on creating interfaces and norms to empower cloud interoperability and task portability across various cloud stages [3, 4].

In multi-clouds, customers demand many kinds of provisions from various suppliers. Therefore, they want the data essentially in every service supplier, which is also important for every supplier. Selecting the best supplier is extremely inconvenient since they do not have the thought regarding the active expense about every resource in various clouds. Negotiation-related provision allotment methodology, purchaser, as well as supplier clearly talk with everyone. If referenced provisions

G. Sumathi (✉)
Department of Information Technology, Kalasalingam Institute of Technology, Krishnankoil, India

S. Rajesh
Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, India
e-mail: srajesh@mepcoeng.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_8

are accessible in solitary supplier, it is clear. In multi-cloud, provisions ought to be assembled from various suppliers, which is a troublesome technique. Similarly, cost of each resource continuously changes dependent on demand and arrangement. So, arriving at present expense about provisions from various suppliers is problematic. Providers depend on reaction from purchasers to which it has completed allotments. Every time broker mediator will verify situation about every cloud supplier. The broker mediator examines with resource supplier mediators also that if the need about a client mediator is not satisfied through a particular supplier, it begins a plan with a different resource supplier mediator.

Various computer scientists and specialists have tried to characterize cloud in different manners. A presented definition is as pursues: "Cloud is kinds about parallel as well as distributed framework comprising about gathering of interlinked also virtualized PCs that are powerfully allocated and introduced as many of the combined computing provisions dependent on the service level arrangements recognized during discussion among service supplier and client [5]."

Distributed computing frameworks can be preoccupied as a bunch of mediators. The effective, heterogeneous, independent, and scattered qualities of distributed computing resources make the efficient administration of an issue to be resolved in the field of distributed computing. The two troubles in accomplishing efficient administration of distributed computing resources are the flexibility and extensibility of the framework. Giving cloud clients with real distributed computing services is finished by each distributed computing service mediator. These distributed computing service mediators can be resources, applications, or a specific sort of distributed computing services [6].

Multi-mediator framework is a sort of framework which can flexibly and keenly react to the requirements of the encompassing cycle and the difference in its working situations. It comprises of various mediators, which have great expansibility and flexibility. It can well resolve the distributed computing resource effectual management of the two troublesome problems, so the presentation of multi-mediator framework in distributed computing resource allocation is a best arrangement [7].

Aiming to attain the cloud resource allocation with dynamic, autonomous, and hetrogeneous resources successfully executed. With the benefits of cloud computing frameworks and multi-mediator framework, this article denotes a dependence on multi-mediator framework of distributed computing resource allocation model and utilizes the upsides of multi-mediator framework and effective distributed computing resource allocation.

8.2 Related Research Works

Yann et al. [5] have characterized the problems about multi-mediator resource allotment. The fundamental problems are conventionally utilized for announcement, methodologies of producer purchaser mediators, and the calculations which are utilized for ideal resource allotment. The convention might be a federal method similar to auctions, otherwise dispersed system similar to discussion. Marian et al. [2] have thought of an effective valuing methodology. In effective evaluating, costs about provisions are placed by the powers about request as well as response. They recommend an auction by an outsider described market producers, who gather bids also choose champ and register instalment.

In multi-mediator framework, a few market-related techniques are utilized for provision designation. Shneidman et al. [8] determine how marketplace tackles provision allotment issue as well as the difficulties on previous business sector-related allotment. Wolski et al. [9] indicate the techniques to computational grid. They define effectiveness about resource distribution below two diverse economic situations similar to service marketplace and sales. They thought about market procedures as far as cost stability, market harmony, and customer productivity and maker effectiveness. Xindong [10] presented cloud provision allotment dependent on marketplace method. In the aforementioned technique, if allotment is in a balance situation below the cost about every resource, next it is ideal. It will receive the advantage about purchaser as well as supplier to the summit. For cost changing, they utilize genetic algorithm [10].

Wei-Yu [11] presented an effective auction method in support of provision allotment on cloud frameworks. They utilized the next evaluated auction method [11] by dynamic cost. It guarantees that supplier could obtain sensible benefit as well as dynamic allotment of its computational provisions. One more sort about auction is progressive double auction [12]. In the aforementioned strategy, supplier decides estimation about demand through its workload as well as customer decides offer worth dependent on excess time to auctioning also leftover provisions to auctioning.

Bo An [13] presented provision allotment via arrangement through recommitment punishment. Buyer discusses among two suppliers also contract with one; another one should be agreed a de-responsibility punishment. The aforementioned technique utilizes systems similar to following: due date, buyer should fulfil its provision prerequisites via the due date, and seller's expense, supplier/dealer will not acknowledge a value lower than its expense. In this method, purchasers openly speak with the suppliers.

8.2.1 Overview of Multi-mediator Framework

Multi-mediator framework is an adaptable intelligent response around the demand and changes in the workplace of the framework [14]. It can move to the relating resource nodes and instances by the task requirements multi-mediator framework is made up by various agreeable mediators. The cooperative problem-solving problem of multi-mediator framework is undeniably more than that of single mediator. Multi-mediator framework is described by non-global control, appropriated information, job asynchronous, knowledge, independence, and sociality. Distributed computing frameworks can be disconnected as a bunch of mediators. Providing cloud clients with real distributed computing services is done by each cloud computing service mediator. These cloud computing service mediators can be resources, applications, or a specific sort of distributed computing services.

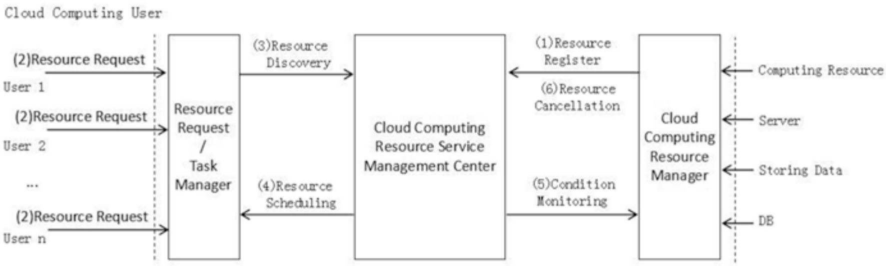


Fig. 8.1 Cloud computing resource management process

8.2.2 Cloud Computing Resource Allocation Process

Distributed computing provision allotment is the principle job – provisions of the match, for the cloud client jobs to indicate the fitting cloud resources. Resources become the reason of dynamic resources of the distributed computing framework is to be enrolled in the framework, to turn into a resource service. For an alternate point of view, distributed computing resource service board community is extraordinary. For the whole distributed computing framework, it is a short reach organizer; for a nearby framework, it is a central regulator. In distributed computing resource allotment scenario cycle as appeared in Fig. 8.1, the principle steps are as per the following:

1. Access to resource data. Resource data acquired incorporates two classifications: the sending of distributed computing related data and distributed computing related data produced by the inward elements.
2. Update resource data. Regularly update the resource data to guarantee the idealness of resource data.
3. Resource disclosure. Predominantly finish the job resources sensible coordinating.
4. Scheduling resource. As indicated by the initial (3) to decide the job resources of the match, the predetermined allotment technique to accomplish a best allocation method.
5. Location resources. The real actual address is acquired from the significant address properties of the resource.
6. Monitoring status. It is fundamentally liable for the status of resources, service execution, just as the completion of the job of cloud client much as checking.

8.3 Mediator-Based Effective Resource Allotment

In negotiation-based resource allotment technique, purchaser and supplier frankly speak with one another. If the mentioned provisions are accessible in a solitary supplier, it is basic. If a cloud is in federated environment, provisions should be gathered from various suppliers, which is a difficult strategy. Additionally, cost of every resource progressively differs depending on request and flexibility. So, obtaining the current cost of resources from various suppliers is exceptionally troublesome.

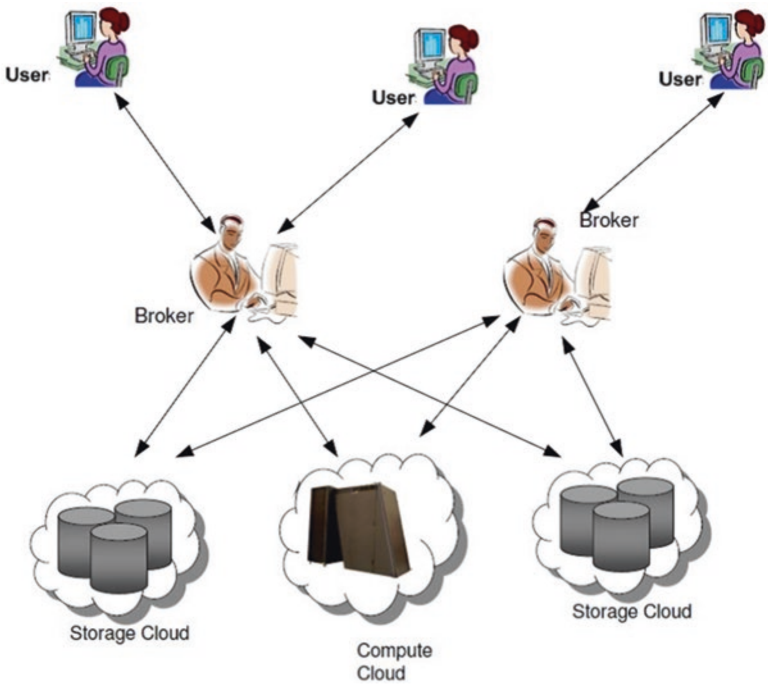


Fig. 8.2 System design

Figure 8.2 displays the proposed framework; it comprises about certain dealers as mediators. Our framework has three kinds of mediators, in particular, consumer mediator, resource broker mediator [15], and resource provider mediator. Consumer mediator forwards its demand toward broker mediator. Broker mediator includes the entire data of cloud suppliers similar to where is the area, what is the present cost of every provision, and least expense about provisions with supplier which gives huge quality of service (QoS). Broker mediator allots an evaluation to the suppliers, depending on input from buyers to which it has completed the jobs. Every time, broker mediator will check the status of every cloud supplier. Broker mediator discusses among resource provider mediators that if the prerequisite about consumer mediator is not satisfied through a solitary supplier, it starts a discussion among other resource provider mediator.

8.3.1 Problem Modeling Purchaser Mediator Has Accompanying Features

- *R*p – pair about resources mentioned via the purchaser. For every $r \in R\text{p}$, $a(R\text{p},r)$ contains the amount of resource *a*.
- ibt – initial begin time about a job to be finished. The job ought not begin prior to ibt.
- pl – period length, measure about time for which provision is mentioned.

- ftt – Finish time about job. Task should be begun prior to ftt, else ftt would be measured as unsuccessful.
- *u* – Discretionary trait to the purchaser to indicate particular service supplier.
- bt – task begin time. The resource broker mediator has a supplier list which holds data of every provision, its expense, and (QoS) quality of service. The resource broker mediator occasionally refreshes supplier list.

Utility about a purchaser mediator is based on task finish time along with its expense. The task finish time is superior if task begin time is nearer to the initial begin time(ibt) and also lesser when nearer to the finish time (ftt). The expense is considered by dividing the number of provisions to the price of provisions. Opl corresponds to the concrete quantity of time occupied through the job purchaser mediator's utility at time *t*:

$$U_p\,(t) = \frac{\text{ftt} - \text{bt}}{\text{ftt} - \text{ibt}} + \frac{\text{Pl}}{\text{Opl}} + \frac{\text{Q}(R\text{p})}{\text{C}(R\text{p})}.$$

8.3.2 Negotiation Protocol

In this design, three models are utilized for discussion: CM, RBM, and RPM. Methodology is the set of rules utilized for exchange about CM by RBM and RPM.

Consumer Mediator Communication Algorithm
Step1: Begin
Step2: load Rp,ibt,ftt,pl,s,bt
Step3: forward CFP (Rp,ibt,ftt,pl,s,bt)to RBM
Step4: accept PROPOSE (Price of Rp p(Rp)) from RBM // Price of Rp
Step5: if price acceptable then
a) forward "ACCEPT SCHEME" to RBM else
b) forward "REJECT SCHEME (price-limit)" to RBM
c) goto step3
Step6: end
Step7: accept PROPOSE from RBM
Step8: if completed then
a) forward"GRANT" to RBM else
b) forward"PROVOKE"to RBM
c) goto step5
Step9: end
Step10: accept "COMPLETED" from RPM
Step11: receive agreement and forward "INTIMATE" to RPM
Step12: running task ...
Step13: compute utility
Step14: send feedback "INTIMATE" to RPM
Step15: End

8.4 Resource Allotment Design Based on Multi-mediator Framework in Cloud Computing

8.4.1 Architecture Design

Cloud computing resource allotment design based on multi-mediator framework is partitioned into three levels [16, 17]: cloud resource level, cloud resource management level, and cloud client level, as appeared in Fig. 8.2.

8.4.1.1 Cloud Resource Level

Cloud resource level is the service of the whole distributed computing framework, which incorporates the computing resources, storage systems, application programming, and different resources. Cloud resource level is the essential capacity of cloud resources for neighborhood control, to give admittance to the upper interface to get to the cloud resources.

8.4.1.2 Cloud Resource Management Level

Cloud resource management level is the center of the multi-mediator framework. It is answerable for the finishing point of the whole distributed computing framework of resource management. Cloud resource management is made out of multi-mediator framework; as indicated by the various executive work, it is principally separated into semantic mediator, query mediator, and resource mediator. They are the solid mediator of distributed computing resource management. Simultaneously, each kind of mediator is separated into two sorts, the primary mediator and the standard mediator, to improve the effectiveness of the entire distributed computing framework.

8.4.1.3 Cloud User Level

Cloud client level is the client of cloud resources. In this level, it can accomplish a wide scope of cloud client applications. It is the most significant level on the framework structure, which is acknowledged through the application support condition. Cloud clients can build up their own cloud applications regardless of whether they do not recognize the information.

8.4.2 Organization Design of Multi-mediator Framework

To additionally improve the effectiveness of resource management in distributed computing frameworks, each sort of mediator is partitioned into master-slave form. Figure 8.3 provides its authoritative structure. The middle control is MM (master mediator). It is answerable for the creation, the executives, and the checking of SM (slave mediator). Simultaneously, the reaction associations outside of the cloud client demands, and acknowledge message with different mediators, to understand the data trade between the master mediator and client. SM gives an approach to manage each sub job. It incorporates two principle fields of p key phrases and strategies. Among them, the function of the key is to give "SM-sub-task" and "MM-SM" coordinating work. The particular usage strategy is answerable for the finishing time of the task area.

8.4.3 Working Process of Multi-mediator Framework

Distributed computing framework can be viewed as a combination of numerous cloud space datas and mediator instances; every area has a space regulator, in this form, by the master mediator charged this function, its fundamental capacity is: inside the space of cloud resource data administration control, message, and enrolment. Each distributed computing resource hub is answerable for the administration of cloud resources inside their own area.

Mediators can be found on the need to finish the job, which can convey the mobile code in each distributed computing space portable. The particular cycle of the work process dependent on the mediators is as follows:

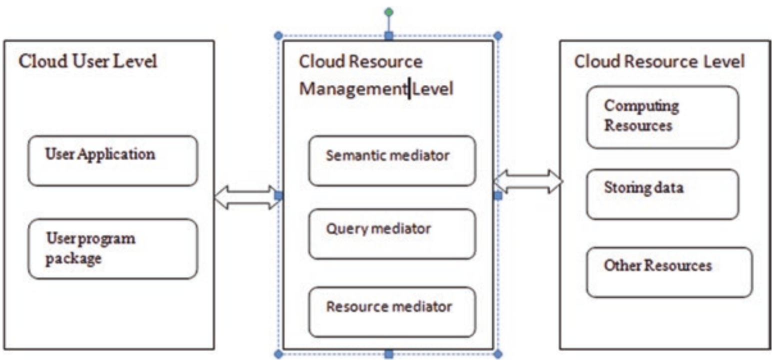


Fig. 8.3 Architecture of resource allotment design based on multi-mediator framework in cloud computing

1. Hub A in the area x issues access to hub B cloud resource demand, because hub B is situated in the space y, and hub A of an alternate area should be executed by the MAx as the middle person of the relating activity.
2. MMx, MMy started to associate in order to decide if the MMy, MMx reaction to the mediator demand;
3. MMy, in the event that you acknowledge the MMx mediator demand, then produces SMy and travels to MMx to execute the comparing activity. SMy can speak with the area y hub for secure message, to complete character verification, and so forth.
4. MMx and hub B finish verification and access control at that point straightforwardly to distributed computing.
5. MMx to hub A is sending distributed computing interchange outcomes.
6. After the completion of node A task, Node B will access the node C from domain Z and node C will sent request to MMx;
7. MMx, MMz started to collaborate in order to decide if the MMz MMx reaction to the mediator demand;
8. MMz, if you agree to acknowledge MMc's mediator demand, at that point produces SMz and moves to MMx. Z SM can speak with the area Z hub for security, to do personality validation, etc.
9. MMx and hub C recognize the identity through the entrance control after the beginning of message.
10. MMx to hub B forwards the outcomes of the distributed computing messages, to finish all the work.

8.4.4 Collaborative Process of Multi-mediator Framework

Various mediators in a multi-mediator framework cooperate to one another to achieve the jobs of cloud clients in the distributed computing framework:

Stage 1. The initialization, the accessible cloud resource hub, and cloud client's demand hub to the distributed computing framework delivered the fundamental data, the data will be discovered the recipient mediator, to the enrolment of cloud resources to the association and resource mediator concurrently; Next, observing mediator answerable for forwarding credit card to every one of the clients of cloud.

Stage 2. Cloud client forwards a demand accessible cloud resource, the scheduling mediator is answerable for the cloud client data of IP address, and the cloud client credit appeal on, if the demand from illicit clients, is dismissed, while observing mediator denoting the client as unlawful clients.

Framework necessities: a cloud client if the resource is set apart as unlawful clients, the scheduling mediator will dismiss any demand of the cloud clients, simultaneously the scheduling mediator to check agency gave a notification to the cloud client screen to the cloud framework, and go to (e); then again, in the event that it looking for legitimate, arranging mediator start to demand the resources required, if the sort is unidentified, Schedules to intimate the client mediator demand is null, and go to (e); then again, if the sort is right, the Scheduling mediator detailing of the allocation Schedule, a few factors that play allocation Schedule generally based on the accompanying: the cloud client demands, and regulations all application approval list, simultaneously to forward control mediator dispersion Schedule;

Stage 3. Control the depending on the organization to the allocation schedule, started to satisfy the need for cloud clients accessible cloud resources to discover, data on the area of the resource hub simultaneously the record; in the event that we can locate the accessible cloud resources, at that point the outcomes will be conveyed to the dissemination mediator, to finish the task; on the other hand, if it cannot be discovered, the appropriation mediator forwards a query disappointment message, informs the agency schedules to drop the demand, by the scheduling mediator will demand drop message to start the demand to a cloud client, and go to (e);

Stage 4. The appropriation of the mediator to the control mediator to restore the final outcome of the allocation of cloud resources;

Stage 5. The finished job.

8.5 Simulation Experiment

Distributed computing resource allocation dependent on multi-mediator framework can increase the reliability and effectiveness of the framework by utilizing the cloud simulation tool test and investigation. Simulation results analyzed 100 distributed computing client demand jobs and need the execution of 20 virtual machines in the distributed computing framework for cloud client task allocation.

Figure 8.4 displays the response time of small instances in cloud environment.

Figure 8.5 displays the response time of medium instances in cloud computing environment.

Figure 8.6 displays the response time of large instances in cloud computing environment.

In the recreation trial of sub-task execution time evaluation, the quantity of sub-tasks is set from 1 to 100, and Fig. 8.7 demonstrates the evaluation outcomes of the sub-task execution time of the two cloud resource board methods. From Fig. 8.7, it tends to be drawn that with the expansion in the quantity of sub-tasks, the execution time of MMSCRM is slower than the development rate of the execution time of TCCRM. Particularly, when the quantity of jobs to arrive at least 80, MMSCRM sub-task execution time than the TCCRM then 16% will be decremented.

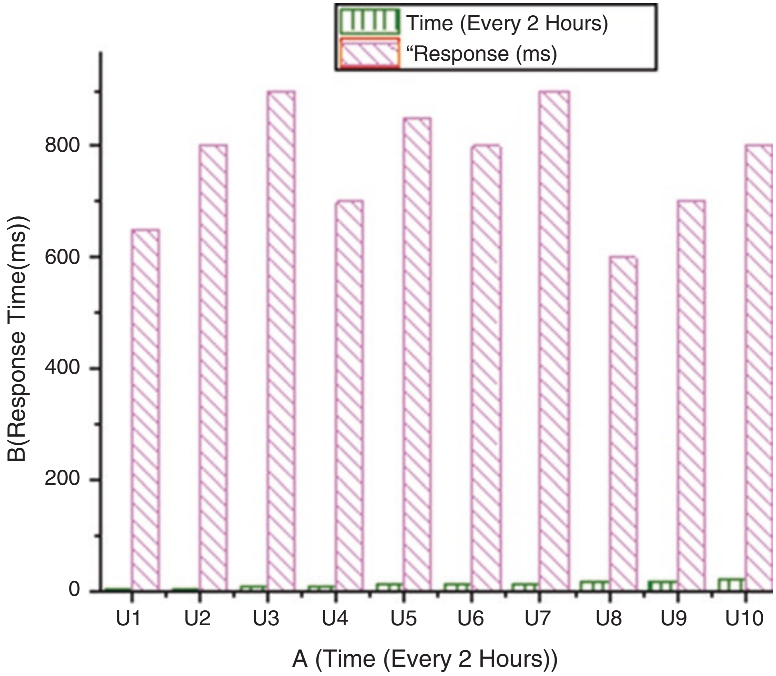


Fig. 8.4 Response time of small instances

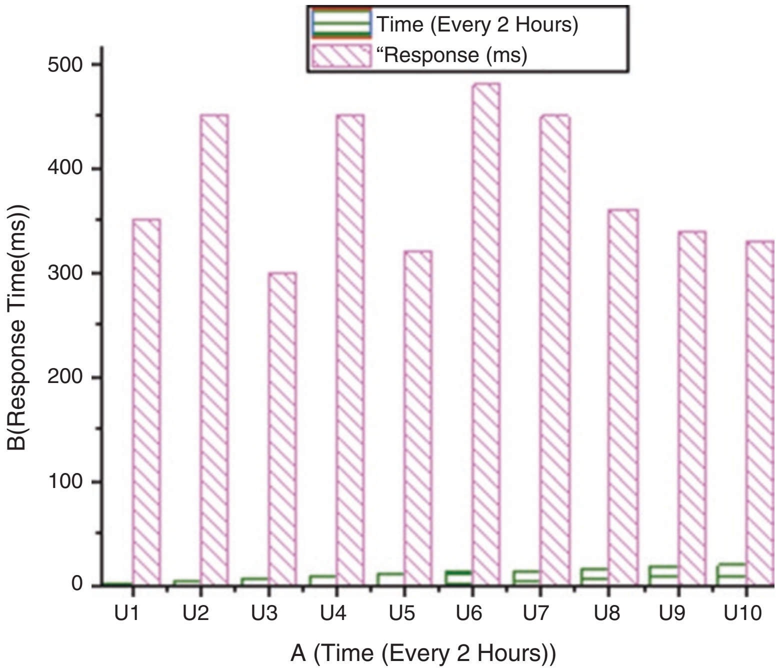


Fig. 8.5 Response time of medium instances

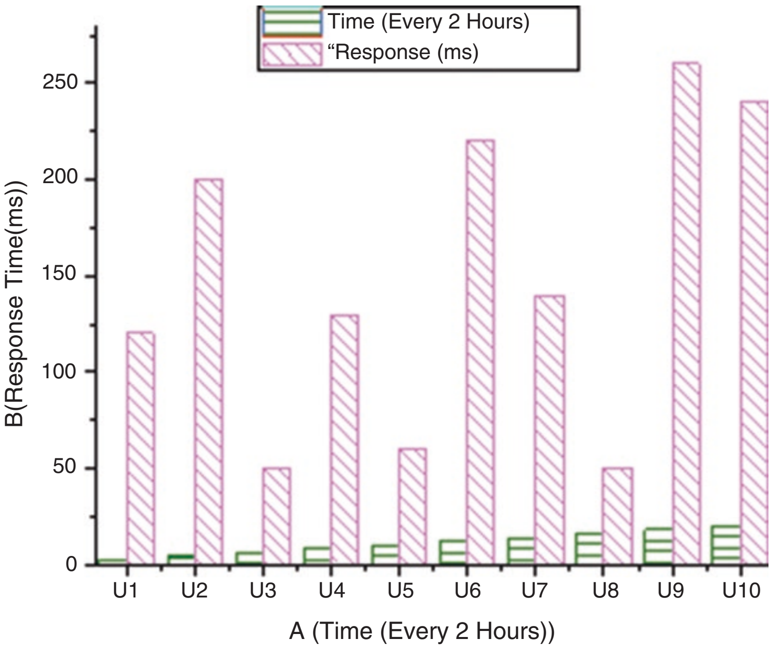


Fig. 8.6 Response time of large instances

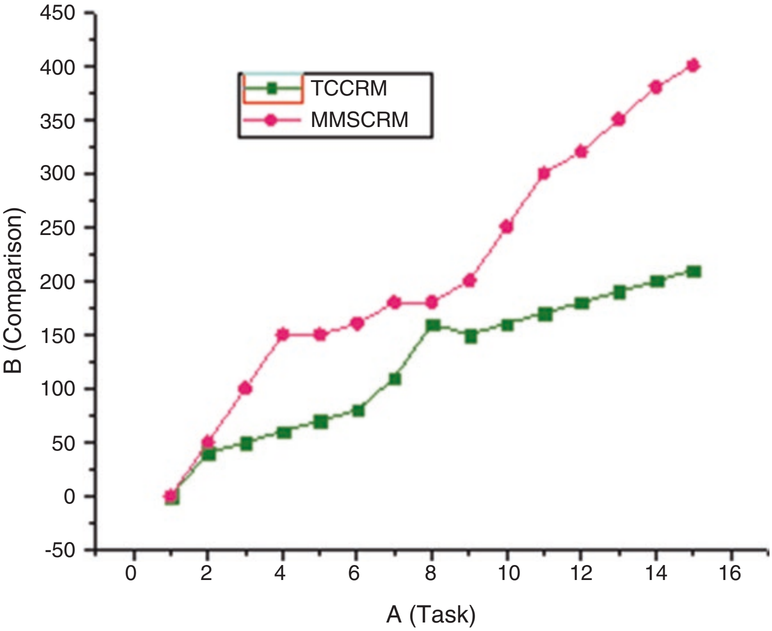


Fig. 8.7 Comparison of subtask execution time

8.6 Conclusions

Generally, cloud computing resource scheduling form dependent on multi-mediator framework gives an achievable and powerful resource allocation form for multi-client cloud computing framework and extraordinarily improves its administration proficiency.

References

1. Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In High performance computing and communications, HPCC'08. 10th IEEE international conference on (pp. 5–13). IEEE.
2. Mihailescu, M., & Teo, Y. (2010). Strategy-proof dynamic resource pricing of multiple resource types on federated clouds. Algorithms and architectures for parallel processing (pp. 337–350).
3. Mihailescu, M., & Teo, Y. M. (2010). Dynamic resource pricing on federated clouds. In 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing (pp. 513–517). IEEE.
4. Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.
5. Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaıtre, M., Maudet, N., Padget, J., Phelps, S., Rodrıguez-Aguilar, J. A., & Sousa, P. (2006). Issues in multiagent resource allocation. *Special Issue: Hot Topics in European Agent Research II Guest Editors: Andrea Omicini, 30*, 3–31.
6. Chen, Q. (2016). Research on cloud computing resource management model based on multi agent system. International conference on computational intelligence and security.
7. Haresh, M. V., Kalady, S., & Govindan V. K. (2011). Agent based dynamic resource allocation on federated clouds. Recent advances in intelligent computational systems (RAICS).
8. Shneidman, J., Ng, C., Parkes, D. C., AuYoung, A., Snoeren, A. C., Vah-dat, A., & Chun, B. (2005). Why markets could (but don't currently) solve resource allocation problems in systems. In Proceedings of the 10th conference on hot topics in operating systems (Vol. 10, p. 7). USENIX Association.
9. Wolski, R., Plank, J. S., Brevik, J., & Bryan, T. (2001). Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications, 15*(3), 258–281.
10. You, X., Xu, X., Wan, J., & Yu, D. (2009). RAS-M: Resource allocation Strategy based on market mechanism in cloud computing. In ChinaGrid annual conference, 2009. ChinaGrid'09. Fourth (pp. 256–263). IEEE.
11. Lin, W. Y., Lin, G. Y., & Wei, H. Y. (2010). Dynamic auction mechanism for cloud resource allocation. In 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing (pp. 591–592). IEEE.
12. Izakian, H., Ladani, B. T., Zamanifar, K., Abraham, A., & Snasel, V. (2009). A continuous double auction method for resource allocation in computational grids. In Computational intelligence in scheduling, CI-sched'09. IEEE symposium on (pp. 29–35).
13. An, B., Lesser, V., Irwin, D., & Zink, M. (2010). Automated negotiation with decommitment for dynamic resource allocation in cloud computing. In Proceedings of the nineth international joint conference on autonomous agents and multi-agent systems (pp. 981–988). Citeseer.
14. Suchithra, R., Selvarani, R., Dagamalai, D. (2011). Elements of cloud computing: A perspective on service oriented enterprises[C]. First international conference on digital image processing and pattern recognition, DPPR 2011, India (pp. 366–377).

15. Haring, G., Kotsis, G., Puliafito, A., & Tomarchio, O. (1999). A transparent architecture for agent based resource management. In Proceedings of the international conference on telecommunications (ICT'98) (pp. 338–342).
16. Gray, R. S., Cybenko, G., Kotz, D., & Rus, D. (2000). Mobile agents: Motivations and state of the art. Dartmouth computer science report TR2000-365.
17. Li, M., Wu, Y., & Chen, J. (2015). Multi-agent system-based semantic search engine for cloud resource management. *Journal of Computational and Theoretical Nanoscience, 12*, 1–6.

Chapter 9
A Robust Communication Strategy for Inter-Cloud Networking Environment through Augmented Network-Aware and Multiparameter Assistance

K. P. Suhaas and S. Senthil

9.1 Introduction

The sudden increase in the challenges in communication between the clouds due to increasingly competitive businesses and offering solution through wireless systems has gained widespread attention in upholding Quality of Service (QoS) provision, improvised energy-efficiency, and higher resource utilization. The ever-changing communication scenario due to various allied factors like weak inter-cloud communication links and frequent network breakages has further accelerated the need for increasing the QoS provisions. Amidst the foremost routing schemes in communication, the inter-cloud communication setup configuration is an extremely distributed infrastructure that predominantly relies on resource-sharing mechanism while parallelly maintaining consistency and coherence. Also, the motivation to increase such properties has improved in a vast deliberation. On the other side, the communication in this inter-cloud infrastructure setup experiences complications in communication as it may suffer network changes and many associated difficulties like increased link vulnerability, loss of data, data retransmission, energy consumption, condensed network lifetime, congestion probability, and unsolicited resource consumption. All these factors reduce the efficiency of the inter-cloud communication in consideration. The communication in a multi-cloud architecture may be used for several operations such as requesting for data, permission to use the resources where the former has no setup, and various other operational goals to felicitate the

K. P. Suhaas (✉)
School of Computing and Information Technology, REVA University, Bengaluru, Karnataka, India

S. Senthil
School of Computer Science and Applications, REVA University, Bengaluru, Karnataka, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_9

end-users. Irrespective of the existing service models and considering the entire cloud environment setup, a finite number of clouds based on their geographical location can be considered as one cloud zone. These zones in turn consist of several intercommunicating clouds within them. This entire setup works in a similar fashion and upholds the functionalities of a cloud infrastructure. The need of creation of such zones that consist of geographically nearer clouds is to ultimately help in conserving the bandwidth and allied QoS parameters such as congestion reduction and preserving link quality. In such a case, the communication between the cloud zones (inter-zonal clouds) and within the clouds of a zone (intra-zonal clouds) is considered using various routing methods. In both cases, the QoS parameters remain the same, and the goal to achieve better and higher QoS provision also remains the same.

The common zone-based routing strategy consists of three chief routing methods. They are reactive routing protocol, proactive protocol, and hybrid routing protocol that can be applied and implemented to achieve data transmission through wireless topological cloud network. The reactive routing strategy and protocol primarily depends on the distance criteria to achieve routing choice [1]. On the opposing side, the classical and proactive routing methods function well with the connectionless topology by carrying out self-developed table management scheme that aids in proactive routing decisions. However, the zone-based routing protocols take advantage of effectiveness and advantages of the both, to accomplish intra-zone and inter-zone routing by transferring of data using the reactive and proactive network management (PNM) schemes, respectively. Overall, the zone-based routing protocol requires enhancing PNM and its associated Best Route Formation (BRF) in case of multi-hop transmission of data through intermediate cloud zones to uphold ideal presentation resulting in minimum data loss, energy draining, and associated delay. Due to frequent disruptions in network, the cloud zones experience very high topological discrepancies and therefore increase the vulnerability of the link, Packet-Loss probability, and extended End-to-End time. Furthermore, it further requires optimum routing pronouncement to improve the aforementioned problems.

With this as a motivation, in this chapter, an effective and augmented robust strategy for Congestion Resilient Communication through Augmented Network-Aware and Multiparameter Assistance for Data Transfers in Inter-Cloud Networking Environment has been established for assurance of Quality of Service. The proposed strategy performs Differentiation of Service (DoS), Dynamic Resource Scheduling (DRS), Dynamic Link Quality Approximation, Congestion Detection and Avoidance models in the Network layer, and data transfer rate estimation at various other levels of the IEEE 802.11a communication standard. Estimating the above-mentioned parameters and considering the multi-hop data transfer in the cloud environment, the proposed system yields the selection of Best Forwarding Node (BFN) and is trailed by the BFR formation for communication between the clouds while simultaneously maintaining the QoS parameters.

This chapter is categorized into six successive parts where Sect. 9.2 represents related work trailed by proposed system in Sect. 9.3. In Sect. 9.4, the results obtained and respective inferences are discussed, which is followed by conclusion. References used are mentioned at the end of the chapter.

9.2 Related Works

Authors in [2] proposed a basic architecture for inter-cloud and for media-storage design considerations. However, the authors could not address the congestion arising due to the extensive media and data transfers. In [3], Vincent et al. used an efficient mechanism for management of data traffic between merged cloud infrastructure environments to conserve bandwidth. The authors in [3] failed to address the size of the data traffic that was considered. In distributed cloud environment, the authors in [4] proposed master-slave communication to support faster data access. Their designed system improves the performance and availability along with higher consistency of the data. However, they could not address the bandwidth requirements. Authors in [5] concentrated on the quality of cloud services. The authors also emphasized on scalability of high-performance computer (HPC) applications. Hence, the authors proposed Network Interface Cards (NIC) as ready-to-use integration component, and the effectiveness was about 38% more efficient compared to the systems without NIC. However, the authors could not stress on the classification and size of the data. Following the low resource utilizations of resources during migration, the authors in [6] considered traffic arising between services, and the proposed system was found to be achieving low latency and lesser load. In [7], a large-scale distributed system of clouds was envisioned. The system considers cloud node qualities and performance in communication between various nodes. The results showed better performance in the cluster-based strategy. Considering the aspect of dynamic resource allocation, the authors in [8] used combination of virtual ad hoc cloud concept. However, the authors could not address the bandwidth issue and associated network congestion degree. In [9], authors used QoS aware cost-efficient choice of data centers with an optimization approach. Although the authors suggested a heuristic approach, they failed to address the delay parameters associated in selection of such data centers. Authors in [10] used scalability issue to instantiate virtual resources. However, authors did not address the virtual resource size management technique. In [11], Dzmitry Kliazovich et al. proposed a green cloud, keeping in mind about the energy-aware and conservation techniques. However, keeping in mind the energy aspect, the authors did not address QoS provisions—bandwidth and congestion probability. Mouna Garai et al. in [7] proposed communication-as-a-service (CaaS) while ensuring best QoS. The authors use Vehicular Cloud Architecture with roadside assistance. As a drawback, the authors did not address the bandwidth allocation in changing topology. Authors in [12] used synchronization approach for scalable communication between the clouds using socket-based communication. The proposed system envisioned cloud environments in which the systems are asynchronous. However, the time-delay and congestion factors were not taken into consideration. In [13], Shiping Chen et al. proposed

Connectivity-as-a-Service for both intra-cloud and inter-cloud environments. However, the authors failed to utilize the resource constraints for such environments. Authors in [14] used dynamic traffic management in a distributed manner. The evaluations show 29% decrease in the transit traffic. However, the bandwidth issues are not addressed in the proposed system.

To exploit effectiveness of zone-based routing for communication, several optimization techniques have been made. Chellathurai et al. [15] established an evolutionary zone-based routing protocol (EZRP) that considered innermost zone as a whole, while the outer zone oppressed the proposed technique for approximating the forwarding path. However, the authors could not report the grid dynamism and its influence on vulnerability of the link, data loss probability, and end-to-end time delay. Sree Ranga Raju et al. [16] focused on using zone-based routing protocol by dropping overload control packets while reconnoitering the Best Forwarding Node technique. In continuation, they also applied a query restriction mechanism for control of traffic. Basically, it attaches the existing routing with the zone assembly to achieve overlying detection of query and its avoidance. Their proposed model empowered zone-based routing protocol to transfer data to all linked nodes with lesser control traffic as associated to proactive route detection methods. Also, researchers like Benni et al. [17] concentrated on the optimization of performance by attaching the Best Forwarding Route choice while they found the Euclidean-based Zonal Routing Protocol improved than Intra-Zone Routing Protocol and Inter-Zone Routing Protocol. A comparable proposal was made in [18, 19], where authors used geographical location info of node to achieve Best Forwarding Route selection. Authors made use of the distance data to evaluate a better route which might lodge all linked nodes to attain better Data Delivery Ratio (DDR). In [20], authors used the idea of affecting the object exhibiting and indexing mechanisms to achieve better routing conclusion. Multi-zonal notion was useful in [20] to attain dependable communication over assorted wireless networks. Authors used triangle zone-based grouping notion to accomplish bigger time. However, this method could not support with mobile topological conditions. Mafakheri et al. [21] projected a fuzzy clustering and adaptive model, called as MACP-FL. In this proposed model, fuzzy clustering mean (FCM) technique was used to fragment the network into several multiple clusters, and associated parameters like remaining energy and membership function were considered to perform the choice to enable transmission. However, this considered method did not report the topic of link vulnerability and data transfer rate to attain target-sensitive communication. To accomplish the QoS presentation, the authors projected a Virtual Base Station (VBS) choosing mode. Kusumamba et al. [22] established a multi-cross-layer architecture-based routing structure (CLRS); however, this investigation chiefly concentrated on lifetime optimization of the node. In [23], a routing model based on cross-layer was established, where authors appraised the cost of individual path to achieve inter-zone BFN and BFR selection. Though the authors used a cross-layer approach, they failed to use buffer utilization. A hierarchical cross-layer optimization protocol (HCLOP) was made in [24] that predominantly focused on achieving high utilization of resource, lesser delay, and low jitter. However, these authors did not consider the probability of congestion. Fahmy et al. [25] have established a Virtual Base Station (VBS) scheme called VBS-ZRP that transfers signal-to-noise ratio data to achieve best next node selection. Prominently, exploring their work, only SNR was considered to become best next node. Equally, Shankar et al. [26] established zone-based geographical multicast routing (ZGMR) scheme that used the distance with duration of link to select the forwarding nodes. A cross-layer-based routing method was created in [27], in which the authors established a model at the network layer of IEEE 802.11 standard. For improvement of the signal inundation, a cross-layered location-aided and energy competent routing strategy was developed. Authors [28] established cross-layer routing strategy model while also supplementing real-time arrangement at the network layer with amplified rate monotonic algorithm (RMA) and earliest deadline first (EDF) strategy.

9.3 Proposed System

This part focusses on the proposed routing protocol for inter-cloud communications. The considered routing scheme utilizes diverse varying parameters from every cloud in the network. The data is collected from various layers of the communication protocol stack to make the optimal BFR selection. To achieve such best route-finding mechanism for transmission of data or resource query mechanism to the destination cloud, the protocol chooses cross-layer routing approach consisting of physical layer, data link layer, network layer, and application layer of the communication stack of IEEE 802.11a (Fig. 9.1). The chosen Differentiation of Service (DoS) supports in recognizing real-time data (RTD) and non-real-time data (NRT) that, in later point in time, aids in better resource allocation. Considering this, the protocol finds the probability of congestion and detection in the network layer to

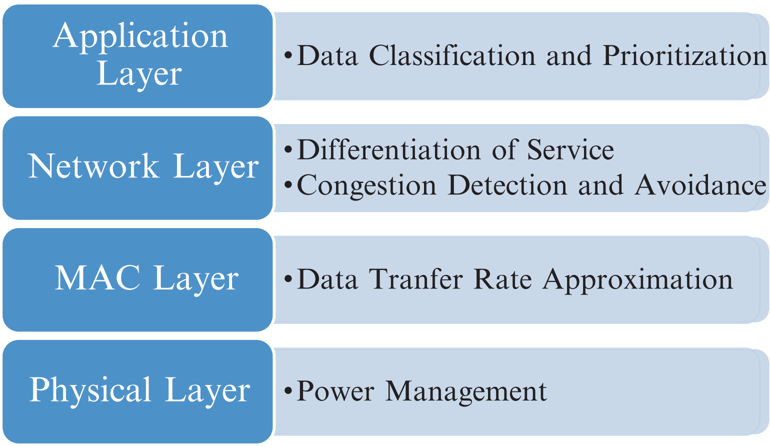


Fig. 9.1 Proposed protocol

undergo best resource management. The routing scheme achieves Quality of Link and Estimation of Data Transfer Rate in the MAC layer. As the protocol is a cross-layer, the protocol encompasses sharing of dynamic information to achieve best optimal routing decision. Overall, the proposed routing protocol performs Differentiation of Service (DoS), dynamic resource scheduling (DRS) along with congestion detection and avoidance, rate of data transfer, and quality estimation of the link. It is to be considered that in this mechanism, every cloud node in the network maintains information about the neighbors that are one-hop away from them. Also, other information like Cloud ID, available and maximum capacity of the buffer storage, and link quality of the neighboring cloud is considered. This information shall be obtained by sending a *beacon* message and the acknowledgement message at regular intervals.

The functions carried out at different subjective layers in the communication stack at every cloud node are listed layer-wise.

Upon receiving a data request from a cloud node, the incoming cloud node resets its timer and thereby avoids unsolicited message or resource requirements from other cloud nodes. This technique also benefits in lessening the congestion probability. In Proactive Node Table Management technique, each cloud node preserves a table. Assume that L_j be a one-hop neighboring cloud node, and let BFN_i be the ideal best and next progressing cloud node in zone-based inter-cloud routing approach, and therefore the table at every cloud node table is updated as

$$C_{\mathrm{Table}} = \left\{ \mathrm{BFN}_i \in L_j \mid D_D - D_F \geq 0 \right\} \tag{9.1}$$

In the above Eq. (9.1), D_D is the distance in Euclidean between the source cloud and destination cloud, and D_F refers the distance in Euclidean between source and the best nearest cloud node. Also, it is noted that an increase in the demand for resources can make a cloud node buffer deficient and therefore inappropriate to be a BFN. The Proactive Network Table Management (PNM) strategy improves the opportunity of the repetitive Node Discovery (ND) procedure which decreases monitoring and signaling expenses in a significant manner. In this proposed system, every contributing cloud node shall preserve information on one-hop neighboring cloud node through the *beacon* message. This beacon comprises several important cloud node information such as Cloud Node ID, determined buffer capacity, existing buffer space, geographical location of the cloud node, data transfer speed, and quality of the link. Remarkably, in this proposed strategy, the discussed parameters are attained with the help of acknowledgement for the beacon that decreases unsolicited and repetitive signaling messages that ultimately causes energy exhaustion.

The complete description of the proposed routing strategy is presented in the subsequent sections.

9.3.1 Differentiation of Service

Fulfilling communication challenges requires finest allocation of resource; yet, the unlike types of data and its significance create resource ranking more challenging. In real-world scenario, a communication will encompass both real-time data and non-real-time data. In such viable and normal scenarios, differentiating the type of data and distributing resource is an immense mission to retain best communication set-up. The proposed protocol assigns the Differentiation of Service (DoS) by considerable resource allocation for real-time data and utmost available resource for the non-real-time data, in terms of buffer capacity. In the proposed routing policy for inter-cloud communication, every node shall contain two dissimilar types of buffers for two distinctive types of data mentioned earlier. In case of a real-time data in consideration, every node shall utilize maximum buffer, i.e., use complete buffer capacity. Now, the remaining and supplementary buffer from the real-time data shall be provided to the non-real-time data. It is also taken cared that the non-real-time data is not discarded to preserve quality and QoS provision. This approach is carried out by considering the first-in-first-out (FIFO)-based allocation approach. In the existing approach, the buffer allocated to the non-real-time buffer is discarded, while the proposed model discards only the data after the FIFO queue has been fully occupied. In this scenario, the advantage is that the data remains in the buffer queue to preserve QoS provision. This way of Differentiation of Service guarantees best and optimal resource allocation along with priority of data.

9.3.2 Congestion Detection

Frequent network link breakages rise the possibility of bottleneck that forces the communication between the clouds to suffer data drop, data retransmission, and reduction in resource power. To improve such issues, the proposed model utilizes a congestion, and such bottleneck detection scheme utilizes buffer capacity to the maximum and current availability of buffer to approximate the probability of congestion in a cloud node. In accumulation, the proposed model integrates a congestion detection and avoidance scheme that estimate the buffer availability in real time or dynamically. Therefore, having the maximum buffer capacity allotted and currently remaining available buffer, the model approximates the likelihood of congestion in a cloud node. This further makes the cloud node to be a contender of a Best Route Selection Path. In continuation, the proposed routing strategy works in parallel with Differentiation of Service (DoS) to avoid congestion probability. As mentioned earlier, buffer dedicated for real-time data stores data in prioritized queue manner, while buffer dedicated for non-real-time data considers FIFO-based approach to store the data. Since there is a lifetime defined for every data, it must be taken cared that it reaches the desired destination cloud well within the stipulated time period. The major contemplations of the proposed approach are to estimate the

Euclidean distance from the source cloud node and the destination cloud node. This supports in finding the data having the topmost precedence as given in (9.2).

$$\text{Priority}(P) = \frac{(\text{Remaining Deadline Time})_i}{\text{Sink Cloud Node} - \text{Source Cloud Node}} \tag{9.2}$$

In Eq. (9.2), Remaining Deadline Time$_i$ states the remaining time, and the denominator is the distance between the source cloud node and progressing/forwarding cloud node and the next sink in Euclidean distance. The remaining time is calculated by the time of arrival of each data packet and is reset before transmission of every new data packet.

Next, a *Consolidated Congestion Rank* is calculated by taking into the buffer capacity of real-time data and non-real-time data. This is denoted as follows:

$$\text{CCR}_r = \frac{\text{Avail}_{\text{NRTMem}} + \text{Avail}_{\text{RTDMem}}}{\max_{\text{NRTMem}} + \max_{\text{RTD}}} + \sum_{i=1}^{\mathbb{N}} CCR_{ri} \tag{9.3}$$

In Eq. (9.3), $\text{Avail}_{\text{NRTMem}}$ is the buffer availability at NRT-FIFO queue, while $\text{Avail}_{\text{RTDMem}}$ is the buffer availability at real-time data buffer. Similarly, the maximum capacity is considered for denominator. Hence, the lesser the CCR value indicates the candidature for route selection.

9.3.3 Inter-Node Data Transfer Rate or Injection Rate

It is more practical to know the data transfer rate or the speed at which the data is sent into the link. It is used to determine the quality of the link. Having more buffer capacity may be a feasible solution, but, without enough capacity for the data to be transmitted in the network, it does not become viable. Hence, the injection rate or the rate of a data transfer is of utmost importance. Otherwise, it may impose high end-to-end delay, which ultimately degrades the QoS provision. In this proposed system of inter-cloud routing strategy for inter-cloud communication, the time between the sending of data packet and receiving of an acknowledgement for the same data packet is used to calculate the inter-cloud distance. In order to decide on the rate at which the data is injected into the network, the round-trip time (RTT) needs to be calculated. The RTT is calculated as the difference in time from the acknowledgement for the data received and the time at which the data transfer was initiated, considering ‘n’ packets being sent.

In this way, engaging the Euclidean distance values and round-trip time, the speed factor S_n is obtained as in (9.4).

$$S_n = \left(\frac{\text{Distance}_{\text{SrcCloud to DestCloud}} - \text{Distance}_{\text{SrcCloud to Next Cloud}}}{\text{Mean RTT}} \right) \tag{9.4}$$

In Eq. (9.4), $\text{Distance}_{\text{SrcCloud to DestCloud}}$ is the distance in Euclidean between the source cloud and the destination cloud, while $\text{Distance}_{\text{SrcCloud to Next Cloud}}$ is the Euclidean distance between the source cloud and the next sink cloud.

9.3.4 Link Quality Estimation

To help in better reliable data transfer and several other QoS provisions, the link quality estimation is of vital importance. The quality of the link varies over distance, and choosing an intermediate cloud node with better link quality aids in faster transfer of data. On the contemporary side, if most of the data received is correct and simultaneously if such a fraction is high, the quality of the link is considered to be high. Considering this as the motivation, the link quality can be derived dynamically by considering the number of data packets sent and the data received. The dynamic link quality index η is estimated as in Eq. (9.5)

$$\eta = \left(\frac{\text{No. of Data Packets}_{\text{recvd}}}{\text{No. of Data Packets}_{\text{sent}}} \right) \tag{9.5}$$

9.3.5 Consolidated Grade Vector Estimation and Node Rank Index

After approximating the network parameters such as rate of data injection rate, quality of the link, and probability of congestion, a parameter called consolidated grade vector (CGV) has been projected that categorizes a cloud node to be established as best forwarding node for best route formation. The CGV is the vector having node rank index (NRI) which is computed as shown in Eq. (9.6).

$$\text{NRI}_i = \omega_1 * \eta_i + \omega_2 * \text{CCR}_i + \omega_3 * S_{ti} \tag{9.6}$$

In expression (9.6), ω is a weighted parameter to be considered on the preference of the network. The range of weight parameter trails the condition as in Eq. (9.7).

$$\sum_{i=1}^{3} \omega_i = 1 \tag{9.7}$$

In Eq. (9.7), the node rank index of *i*-th cloud node is estimated for all neighboring cloud nodes and is updated in the PNM table. The algorithm is given as follows (Fig. 9.2):

Here, the cloud node with maximum quality is considered as the best forwarding node (BFN). In other words, once estimating the NRI value, the cloud node having a maximum NRI value is considered the best forwarding node. However, with

Algorithm for Best Next Cloud Node Selection
Input: *Number of cloud nodes, max buffer capacity in each cloud, geographical inter − cloud distance.* ***Output***: *Cumulative Grade Vector, Next Forwarding Cloud Node* *Initialize* $CGV = 0;$ $Next\ Best\ Ideal\ Cloud\ Node = -1;$ ***foreach*** *Cloud_node i, estimate −* $CGV\ Score = \omega_1 * \eta_i + \omega_2 . CCR_i * Neighbour\ Cloud\ Node. CCR_i + \omega_3 . S_{t_i};$ $If\ CGV\ (Threshold) \leq CGV_{Score\_i} then$ $Choose\ ID\ of\ Best\ Next\ Cloud\ Node\ (BNN) = i;$ ***end*** ***end***

Fig. 9.2 Pseudo-algorithm

ever-changing network dynamics, the current value of NRI may be obsolete. In such a case, a threshold value is computed by considering the previous '*n*' values and comparing the computed threshold value with the current NRI value. Now, a cloud node with NRI value more than the threshold is selected as the BFN for best QoS-enabled data transfer.

9.4 Results and Discussion

The proposed routing strategy has considered several QoS and consistency limitations to achieve best next cloud node selection that guarantees optimum direction-finding decision. By considering the advantage of dynamic quality of link, rank of congestion, and rate of data transfer, the parameter named cumulative grade vector (CGV) was found to determine the appropriateness of a cloud node to become best next forwarding cloud. The proposed scheme has been established based on the IEEE 802.11a.

The simulation considers each network node as a cloud having enough resources such as buffer capacity and residual energy. The simulation parameters used are presented in Table 9.1.

The performance assessment of the developed scheme has been carried out in terms of data delivery ratio (DDR) of the real-time data and non-real-time data and overall deadline miss ratio.

In Fig. 9.3, it is observed that the proposed routing protocol outclasses other zone-based routing techniques like zone-based geographical multicast routing (ZGMR) and Virtual Base Station-zone-based routing strategy (VBS-ZRP) in terms of advanced and better DDR performance. For real-time data, it shows approximately 97.92% of data delivery ratio, which is more than other approaches like ZGMR (88.9%) and VBS-ZRP (82.0%). The reason is that the ZGMR and VBS-ZRP do not have the multi-buffer establishment and related scheduling of resource, and hence it has caused into lower data delivery ratio performance.

Table 9.1 Simulation parameters

Parameter	Specification
Programming	Matlab scripting
Physical	IEEE 802.11PHY
MAC	IEEE 802.11MAC
Cloud nodes	10, 20, 30, 40, .., 300
Data transfer size	512 bytes
Deadline time	10 s
Weight parameters	$\omega_1 = 0.33$, $\omega_2 = 0.33$, $\omega_3 = 0.33$
Simulation period	1000 s
Payload	250, 500, 750, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000.

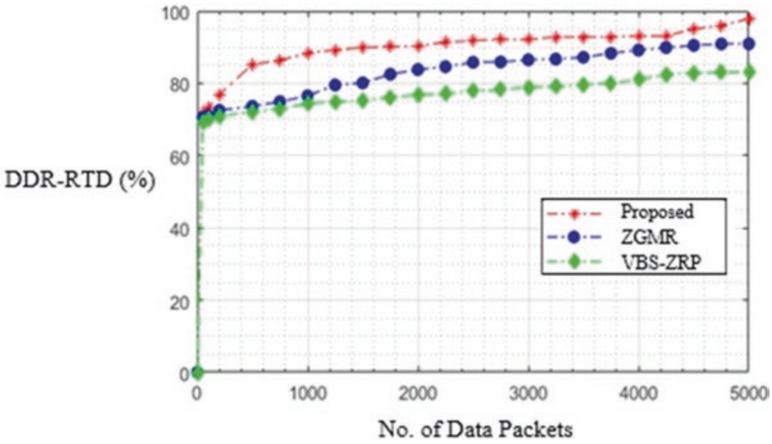


Fig. 9.3 Data delivery ratio performance of real-time data

Also, in Fig. 9.4, the proposed routing strategy showcases about 92% of data delivery ratio, while the classical zone-based geographical routing achieves maximum rate of 89%, which is about 3% less than the former. Following, the Virtual Base Station-zone-based routing strategy achieves about 82% of PDR for non-real-time data.

In Fig. 9.5, the minimum deadline miss performance is better for the proposed inter-cloud routing approach compared to others.

The attained results uphold that the proposed routing strategy as it accomplishes reliable transfer of data, which affirms its suitability for real-time data and non-real-time data traffic.

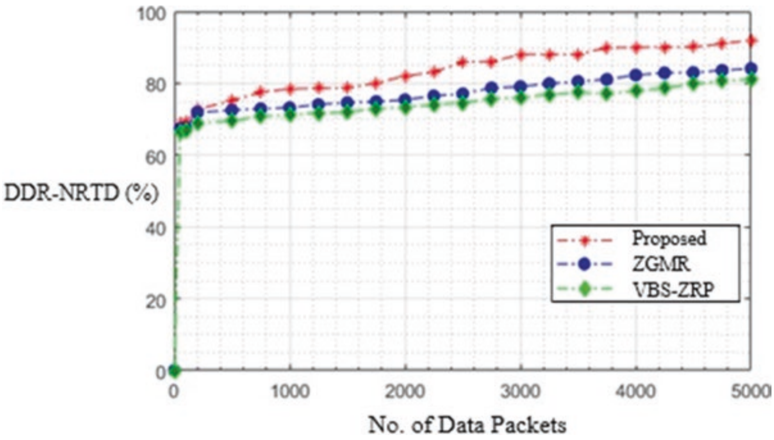


Fig. 9.4 Data delivery ratio performance of non-real-time data

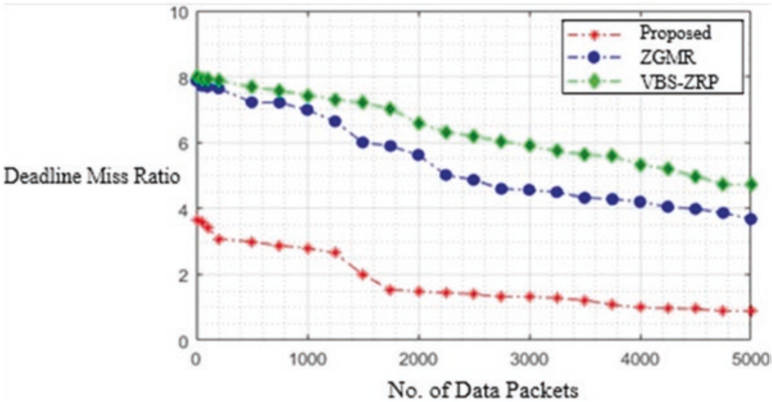


Fig. 9.5 Deadline miss performance ratio

9.5 Conclusion

The dynamically varying network constraints such as bandwidth, quality of the link, residual energy, and congestion can enforce vast changes in time and cost during transfer of data and during requesting for resources from one cloud node to another. These dissimilarities can force network to experience outrage or sometimes loss of data. These network parameters and actions are intermittently recorded by the zone-based routing strategy. To improve such problems, in this chapter, a robust strategy with augmented network-aware and multiparameter support for data transfers in inter-cloud networking environment has been designed. The strategy established a cross-layer approach to achieve the finest best next forwarding cloud node. The proposed strategy performed detection of congestion and differentiation of service along with considering data transfer rate and quality of link estimation that are collectively recorded to predict best next cloud node for forwarding the data in the inter-cloud networking environment. This ultimately yields better Quality of Service (QoS) provision. The strategy uses a cumulative grade vector (CGV) estimation that uses link quality, congestion degree, and data transfer rate with weighted parameter to decide on the node rank index (NRI) value for the cloud node. The results in the simulation show that the proposed inter-cloud zonal routing strategy accomplishes maximum data delivery ratio of 98% for real-time data while preserving approximately 92% of data delivery ratio for non-real-time data traffic. It also implies the strength of the proposed multiparameter assisted based route selection. The complete outcome encourages the appropriateness of the proposed strategy for real-time data in the inter-cloud communication environment.

References

1. Corson, S., & Macker, J. (1999). Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations. https://doi.org/10.17487/rfc2501.
2. Aazam, M., & Huh, E. N. (2014). Inter-cloud media storage and media cloud architecture for inter-cloud communication. 2014 IEEE 7th international conference on cloud computing. https://doi.org/10.1109/cloud.2014.151.
3. Emeakaroha, V. C., Healy, P., Fatema, K., & Morrison, J. P. (2013). Analysis of data interchange formats for interoperable and efficient data communication in clouds. 2013 IEEE/ACM 6th international conference on utility and cloud computing. https://doi.org/10.1109/ucc.2013.79.
4. Kumar, V., Ravikumar, K., Aravinth, S. S., & Rajkumar, B. (2014). A message passing interface to support fast data access in distributed cloud environment along with master and slave communication. Second international conference on current trends in engineering and technology – ICCTET 2014. https://doi.org/10.1109/icctet.2014.6966307.
5. Maliszewski, A. M., Vogel, A., Griebler, D., Roloff, E., Fernandes, L. G., & Navaux Philippe, O. A. (2019). Minimizing communication overheads in container-based clouds for HPC applications. 2019 IEEE symposium on computers and communications (ISCC). https://doi.org/10.1109/iscc47284.2019.8969716.
6. Yang, J., Liu, C., Shang, Y., Mao, Z., & Chen, J. (2013). A communication-aware deployment method for communication-intensive applications in service clouds. 2013 international conference on cloud computing and big data. https://doi.org/10.1109/cloudcom-asia.2013.14.
7. Fan, P., Wang, J., Zheng, Z., & Lyu, M. R. (2011). Toward optimal deployment of communication-intensive cloud applications. 2011 IEEE 4th international conference on cloud computing. https://doi.org/10.1109/cloud.2011.54.
8. Dhiyanesh, B. (2012). Dynamic resource allocation for machine to cloud communications robotics cloud. 2012 international conference on emerging trends in electrical engineering and energy management (ICETEEEM). https://doi.org/10.1109/iceteeem.2012.6494498.
9. Hans, R., Lampe, U., & Steinmetz, R. (2013). QoS-aware, cost-efficient selection of cloud data centers. 2013 IEEE sixth international conference on cloud computing. https://doi.org/10.1109/cloud.2013.113.
10. Mthunzi, S. N., Benkhelifa, E., Alsmirat, M. A., & Jararweh, Y. (2018). Analysis of VM communication for VM-based cloud security systems. 2018 fifth international conference on software defined systems (SDS). https://doi.org/10.1109/sds.2018.8370441.

11. Kliazovich, D., Bouvry, P., & Khan, S. U. (2012). Simulating communication processes in energy-efficient cloud computing systems. 2012 IEEE 1st international conference on cloud networking (CLOUDNET). https://doi.org/10.1109/cloudnet.2012.6483687.
12. Ekanayake, J., Jackson, J., Lu, W., Barga, R., & Balkir, A. S. (2011). A scalable communication runtime for clouds. 2011 IEEE 4th international conference on cloud computing. https://doi.org/10.1109/cloud.2011.21.
13. Chen, S., Nepal, S., & Liu, R. (2011). Secure connectivity for intra-cloud and inter-cloud communication. 2011 40th international conference on parallel processing workshops. https://doi.org/10.1109/icppw.2011.54.
14. Dulinski, Z., Stankiewicz, R., Rzym, G., & Wydrych, P. (2020). Dynamic traffic management for SD-WAN inter-cloud communication. *IEEE Journal on Selected Areas in Communications, 38*(7), 1335–1351. https://doi.org/10.1109/jsac.2020.2986957
15. Chellathurai, A. S., & Raj, E. G. (2013). EZRP: Evolutionary zone routing protocol. 2013 international conference on advanced computing and communication systems. https://doi.org/10.1109/icaccs.2013.6938740.
16. Sreerangaraju, M. N., & Mungara, J. (2010). Tuning ZRP framework for CR networks and MANETs. International Performance Computing and Communications Conference. https://doi.org/10.1109/pccc.2010.5682299.
17. Benni, N. S., Manvi, S. S., & Anup T. H. (2015). A performance study of hybrid routing protocols for variation of nodes in wireless mesh networks. 2015 international conference on communications and signal processing (ICCSP). https://doi.org/10.1109/iccsp.2015.7322732.
18. Maranur, J. R., & Mathpati, B. (2017). VANET: Vehicle to vehicle communication using moving zone-based routing protocol. 2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEECCOT). https://doi.org/10.1109/iceeccot.2017.8284577.
19. Lin, D., Kang, J., Squicciarini, A., Wu, Y., Gurung, S., & Tonguz, O. (2017). MoZo: A moving zone based routing protocol using pure V2V communication in VANETs. *IEEE Transactions on Mobile Computing, 16*(5), 1357–1370. https://doi.org/10.1109/tmc.2016.2592915
20. Mahboub, A., En-Naimi, E. M., Arioua, M., Ez-Zazi, I., & Oualkadi, A. E. (2016). Multi-zonal approach clustering based on stable election protocol in heterogeneous wireless sensor networks. 2016 4th IEEE international colloquium on information science and technology (CiSt). https://doi.org/10.1109/cist.2016.7805018.
21. Mafakheri, M., & Hosseinzadeh, S. (2015). A fuzzy clustering-based mobility-adaptive routing protocol for wireless sensor networks. 2015 7th conference on information and knowledge technology (IKT). https://doi.org/10.1109/ikt.2015.7288768.
22. Kusumamba, S., & Kumar, S. D. (2015). A reliable cross layer routing scheme (CL-RS) for wireless sensor networks to prolong network lifetime. 2015 IEEE international advance computing conference (IACC). https://doi.org/10.1109/iadcc.2015.7154865.
23. Villaverde, B. C., Rea, S., & Pesch, D. (2009). Multi-objective cross-layer algorithm for routing over wireless sensor networks. 2009 third international conference on sensor technologies and applications. https://doi.org/10.1109/sensorcomm.2009.94.
24. Wei, Y., Xie, G., & Li, Z. (2007). A hierarchical cross-layer protocol for group communication in MANET. 2007 IEEE international conference on telecommunications and Malaysia international conference on communications. https://doi.org/10.1109/ictmicc.2007.4448610.
25. Fahmy, I. M. A., Nassef, L., Saroit, I. A. & Ahmed, S. H.. (2010). QoS parameters improvement for the hybrid zone-based routing protocol in MANET. 2010 the 7th international conference on informatics and systems (INFOS).
26. Shankar, R., & Ilavarasan, E. (2014). Geographie multicast routing protocol for achieving efficient and scalable group communication over MANET. Proceedings of IEEE international conference on computer communication and systems ICCCS14. https://doi.org/10.1109/icccs.2014.7068170.

27. Li, Z., & Yang, X. (2016). A reliability-oriented web service discovery scheme with cross-layer design in MANET. 2016 IEEE international conference on web services (ICWS). https://doi.org/10.1109/icws.2016.59.
28. Rath, M., Pati, B., & Pattanayak, B. K. (2017). Cross layer based QoS platform for multimedia transmission in MANET. 2017 11th international conference on intelligent systems and control (ISCO). https://doi.org/10.1109/isco.2017.7856026.

Part III
Next Generation Technologies for Multi-cloud

Chapter 10
An Intense Study on Intelligent Service Provisioning for Multi-Cloud Based on Machine Learning Techniques

C. D. Anisha and K. G. Saranya

10.1 Introduction

The introduction section provides overview on services of cloud computing and need, terminologies, and challenges of multi-cloud computing.

10.1.1 Cloud Computing and Its Services

Cloud computing is a term that represents a distributed model which allows sharing of resources in form of services without any restriction of time (anytime), location (anywhere), and user (anyone). This environment provides the services to users in pay as you go manner, i.e., on a subscription basis. The basic requirement to use cloud services is Internet connectivity. The services related to education, business, and governance are provided through online wherein users can access though web browser. The most prominent cloud service providers are Amazon, Google, and Microsoft [1]. The datacenter which consists of cloud server that has data and software programs stored in it provides resources to any user in the world. The approach of cloud computing enables effective use of resources and to procure and update data of a user without any purchasing of license.

C. D. Anisha (✉)
Research Scholar (Ph.D), Department of CSE, PSG College of Technology, Coimbatore, India

K. G. Saranya
Assistant Professor (Sr.Gr), Department of CSE, PSG College of Technology, Coimbatore, India
e-mail: kgs.cse@psgtech.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_10

10.1.2 Multi-Cloud Computing: Need, Terminologies, and Challenges

Multi-cloud is referred to as cloud system in which applications are hosted in blocks among a diverse network of distinct cloud.

The need for multi-cloud is as follows:

1. *Choice of service*: Single cloud service provider (CSP) vendor will not provide all services essential for the organization. Integrating utilization of services from multiple CSP is made possible through multi-cloud.
2. *Latency*: The service can be chosen based on the distance. The service which is nearer to the user can be provided in multi-cloud environment.
3. *Increased disaster recovery*: Multi-cloud environment allows the replicas of applications in two or more clouds. This capability allows the user to access another cloud if there is downtime in one cloud.

Terminologies and its associated tools used in multi-cloud are as follows:

1. *Library-based approaches:*
   (a) *jclouds*: It is an open-source java library used to provide mobility of java applications.
   (b) *libcloud*: It is a python library that provides detachment of various differences among the programming interfaces.
2. *Service-based approaches:*
   (a) *Hosted services*: These are commercial services, and the most commonly hosted services are RightScale, enStratus, and Kaavo.
   (b) *Deployable services*: These are open-source projects, and the most commonly used deployable services are Aeolus, mOSAIC, and optimis.

Challenges in multi-cloud are as follows:

1. *Cost management*: The biggest challenge in multi-cloud is the management of cost, as different cloud service providers (CSP) fix price with different meters. The construction of multi-cloud with available cost in organization is a huge challenge in multi-cloud.
2. *Assets management*: Organization will get access from developed CSPs and developing CSPs. Complexity of managing asset increases when integrating the developed and developing CSPs in multi-cloud.
3. *Performance management*: Organization performance will be affected if the services are chosen in a substandard manner.
4. *User management*: Managing access rights among users becomes a challenge in multi-cloud environment.

10.2 Classification of Services Provisioning in Multi-Cloud

Service provisioning is the process of allocating the on-demand resources to the user. The service provisioning provides four major services, namely, SaaS (Software as a Service), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Security as a Service (SecaaS). Service provisioning involves monitoring and orchestration.

The service provisioning in multi-cloud can be classified based on the following:

1. The characteristics, namely, workload, elasticity, and location (placement and consolidation)
2. The approaches for service provisioning:
   (a) Service provisioning models
   (b) Brokerage aided provisioning
   (c) Policy ensure service level agreements (SLAs)
   (d) Heuristic and holistic perspective
   (e) Multi-criteria decision-making (MCDM)
   (f) Algorithmic techniques
3. Services orchestration at each service level:
   (a) IaaS service orchestration
   (b) PaaS service orchestration
   (c) SaaS service orchestration

10.2.1 Objectives, Topologies, Requirements, and Procedures in Service Provisioning of Multi-Cloud

10.2.1.1 Objectives in Service Provisioning of Multi-Cloud

The objectives in service provisioning of multi-cloud includes self-service provisioning, autonomous workload distribution, elasticity, and removal of latency constraint.

Self-Service Provisioning and Autonomous Service Provisioning

The main objective of provisioning in multi-cloud is to provide self-service provisioning which means the user can select the service as per the requirement with less intervention from the cloud service provider (CSP). The service provisioning can also be autonomous which provides service based on user requirements with less user intervention in service selection process.

Autonomous Workload Distribution

The workload distribution which means allocating and releasing resources should be handled in efficient manner in order to avoid infrastructure failures.

Elasticity

The elasticity is also key objective which ensures computational resources in multi-cloud are scaled with flexibility. To ensure elasticity, there are certain approaches to be adopted, namely, load balancing, application scaling, and migration.

Removal of Latency Constraint: Location (Placement and Consolidation)

The latency constraint is another most vital objective which can be removed in multi-cloud environment since the location of data center is provided based on the customer geographical parameters which yield higher availability without interruption.

10.2.1.2 Topologies in Service Provisioning of Multi-Cloud

The topologies associated with service provisioning based on approaches criteria are service provisioning model, brokerage-aided provisioning, policy ensure service level agreements (SLAs), heuristic and holistic perspective, multi-criteria decision-making (MCDM), and algorithmic techniques.

Service Provisioning Model

The Service Measurement Index (SMI) can be used to meet the objective of self-provisioning. This index is a service measurement model which is mainly based on the business model of the International Standard Organization (ISO). SMI model allows the users to choose based on the dimensions provided with various cloud offerings.

Brokerage-Aided Provisioning

The cloud brokers play a prominent role in satisfying the autonomous service provisioning objective. In this brokerage-aided provisioning method, the broker first collects all details of the services of each CSP, analyzes, and creates an index which is utilized when a user request is provided. The cloud performs matching task on the index created and provides the best service to the user.

Policy Ensure Service Level Agreements (SLAs)

Service level agreements (SLAs) are the terms and conditions to which consumer and provider have a mutual agreement. In service provisioning of multi-cloud environment, the process of SLA has to be automated or semi-automated in order to meet the objective of elasticity which in turn reduces the cost and increases the trustworthiness of service provider.

Heuristic and Holistic Perspective

Frameworks and tools have been existing in providing holistic and heuristic task selection and resource allocation options with better resource utilization and energy aware features.

Multi-criteria Decision-Making (MCDM)

This MCDM method is used to select the best service based on the performance measurements provided by the third part monitoring tools. This method is proven to be used for real-world complex problems of service selection.

Algorithmic Techniques

Genetic algorithm and intelligent service provisioning using machine learning techniques have been used as a best search tool for service selection. These techniques have also been used for task scheduling in a dynamic manner and deployment of services in an optimal manner.

Requirements in Service Provisioning of Multi-Cloud

Requirements about multi-cloud is unique for each organization, but patterns of requirements can be identified which is of two broad categories: patterns related to distributed deployment of applications and patterns related to redundant deployment of applications.

Patterns Related to Distributed Deployment of Applications

The concept of this pattern is mainly about the integrations of public clouds from various cloud service providers. Partitioned multi-cloud pattern is a best example under this category. The mechanism of partitioned multi-cloud pattern is to execute an application A in one public cloud of one vendor (Amazon Web Service), whereas application B in another cloud owned by different vendor (Azure). The advantages are avoidance of vendor lock in, and shifting of workloads in different computing environment has been simplified.

Patterns Related to Redundant Deployment of Applications

The concept of this pattern is focused on deployment of replicas of applications in multiple cloud environments in order to increase scalability and availability.

10.2.1.3 Procedures in Service Provisioning of Multi-Cloud

The various procedures involved in the process of service provisioning of multi-cloud are user request scheduling, service selection, service composition, service monitoring, and orchestration.

User Request Scheduling

Cloud services are requested by the cloud consumer which is sent as tasks or jobs at the datacenter of the cloud service provider (CSP). The multi-cloud environment is set up to ensure availability higher than the single cloud environment. Sanjaya et al. [2] have proposed task scheduling algorithm which is allocation aware using traditional Min-Min and Min-Max algorithm for multi-cloud environment.

Service Selection

Service selection is the first and foremost procedure in service provisioning. The user selects the service based on the standards, price, and flexibility. In autonomous service selection process wherein the user intervention is less, the service is selected based on the rank and prediction obtained from the analysis done on the service performance, price, and its features.

Service Composition

Cloud service composition is a procedure of composing distant meta-services which is simpler and satisfies the consumer requirements. Cloud service composition can be categorized as two types, namely, functional and non-functional known as Quality of Service (QoS).

Service Monitoring and Orchestration

Service monitoring is the process of providing measurement information which is required for pricing and also for performance analysis.

Service orchestration is essential for coordinating the process between the services, namely, IaaS, PaaS, and SaaS.

At IaaS level service orchestration in multi-cloud, it requires automation and abstraction to handle different standards of CSP and heterogenous API of each CSP. The automation and abstraction are provided using libraries, standards, and models and cloud orchestration tools. The common libraries used for abstraction are Apache j clouds, Apache Lib cloud, and fog. The standards and models used for automation and abstraction are Open Cloud Computing Interface (OCCI), OASIS TOSCA. The cloud orchestration tools used for abstraction and automation are Apache Brooklyn and Cloudify.

At PaaS level service orchestration in multi-cloud, it focuses on application-centric resources and pre-defined environments. CloudFoundary and OpenShift are used as an API abstracting layer for PaaS.

Cross-level service orchestration is a novel area wherein orchestration is provided across different level of service rather than one service level (IaaS or PaaS). CloudML provides cross-level service orchestration.

10.3 Intelligent Service Provisioning (ISP) in Multi-Cloud

10.3.1 ISP: Methodologies, Advantages, and Limitations in Multi-Cloud Environment

The review on intelligent service provisioning (ISP) in multi-cloud is presented into three categories. The problems and solutions using machine learning techniques in each category has been specified. The three categories incorporated in review are characteristics of service provisioning, approaches of service provisioning, and procedures of service provisioning in multi-cloud. Table 10.1 presents the classification of Intelligent Service Provisioning (ISP).

10.3.1.1 ISP Based on Characteristics of Service Provisioning

The ISP based on characteristics of service provisioning involves workload management, elasticity, and removing latency constraint.

Table 10.1 Classification of ISP

S. no	ISP – classification category	ISP – attributes	Machine learning techniques used
1.	Characteristics of service provisioning	Workload management	Deep belief networks (DBN)
		Elasticity	Reinforcement learning -Q learning
		Removing latency constraint (placement and consolidation)	Support vector machine (SVM)
2.	Approaches of service provisioning	Service provisioning models	1. Non-hierarchical clustering 2. Content-based filtering 3. Behavioral and collaborative filtering 4. Informational filtering
		Brokerage aided provisioning	Multi-learning cloud broker
		Policy ensured service level agreements (SLAs)	Support vector machine (SVM)
		Heuristic and holistic perspective	Reinforcement learning (RL)
		Multi-criteria decision-making (MCDM)	Principal factor analysis (PFA)
		Algorithmic techniques	Deep learning technique – Long short-term memory (LSTM)
3.	Procedures for service provisioning	User request scheduling	Deep reinforcement learning (DRL)
		Service selection procedure	Hierarchical clustering algorithm
		Service composition	Bayesian based model
		Service monitoring and orchestration	1. Random Forest 2. Adaptive boosting classifier 3. Binary logistic regression 4. Neural network 5. Extreme gradient boosting (XgBoost)

Intelligent Service Provisioning for Workload Management

Workload management is the method of managing resources in cloud system. The resources management is of two types, and they are pro-active and reactive management. Reactive management is only providing analysis based on monitored data which is a time-consuming process, whereas proactive management is providing future workload prior for effective management of resources in multi-cloud environment. Intelligent service provisioning (ISP) focusing on workload management is essential for effective resource utilization with Quality of Service (QoS) and less power consumption at the multi-cloud environment. The intelligent workload management process means providing workload prediction and forecasting. In workload prediction and forecasting, it provides the workload of VM in advance which aids VM to auto-scale its resources to fulfill Quality of Service (QoS) and saves consumption of energy. Deep belief network (DBN) is a deep learning technique which has been used in image classification, audio classification, and speech recognition. There are correlations among VMs in multi-cloud environment whose workload can be predicted using deep learning model which consists of DBN and logistic regression. The input given to the deep learning model is the previous CPU utilization, and the top layer which is the logistic regression layer predicts the future workload of all VMs. The output given by the deep learning model is the future CPU utilization of all VMs from which the VMs can auto-scale its resources appropriately. The advantage of workload prediction using deep learning model is that its performance is high since the inherent features of the workload are used rather than using only monitored data [3].

Intelligent Service Provisioning for Providing Elasticity

Elasticity in multi-cloud can enable using auto-scaling mechanisms. Reinforcement learning is one of the auto-scaling mechanism which consists of a decision-making agent which makes decisions based on the experiences and provides the appropriate actions to execute which can be addition and deduction of resources and gaining benefits such as maximum throughput and less response time [4].

Barett et al. [5] incorporated reinforcement learning known as Q learning to provide scaling policies which is optimal in nature, and a Q learning in parallel version has been proposed to reduce the execution time.

Intelligent Service Provisioning for Removing Latency Constraint (Placement and Consolidation)

The placement and consolidation of resources has to be performed efficiently in order to remove the latency constraint. T. Miyazawa et al. [6] provide a way to select resources of non-urgent virtual network for migration automatically and dynamically using support vector machine (SVM) and Q learning with satisfying QoS requirements.

10.3.1.2 ISP Based on Approaches of Service Provisioning

The ISP based on approaches of service provisioning involves service provisioning models, brokerage aided provisioning, policy ensure service level agreements (SLAs), heuristic and holistic perspective, multi-criteria decision-making (MCDM) algorithm, and algorithmic techniques.

ISP Based on Service Provisioning Models

Cloud services selection is a challenging and tedious task for the cloud users especially in a multi-cloud environment. To overcome this challenging task, Rahma et al. [7] have developed a cloud service recommendation system (prototype model) named "USTHB-CLOUD." This model provides recommendations based on content-based and behavior-based analysis from the customer preferences using machine learning techniques, namely, non-hierarchical clustering methods, content-based filtering, behavioral collaborative filtering, and informational filtering. The limitations of this model are that data about service level agreements (SLAs), infrastructure saturation, and energy consumption has not been incorporated in the recommendation system.

ISP Based on Brokerage-Aided Provisioning

Kiran Bala et al. [8] present a machine learning-based broker for decision-making for scheduling the tasks (requests) made by the user at the data center. In multi-learning process of the broker, it learns from all the previous mishandled user requests which are procured as results from single learning. Incorporation of multi-learning method to the cloud broker is made possible through machine learning technique. This machine learning-based cloud broker enhances the accuracy of the decision-making process for each user request for services.

ISP Based on Policy Ensure Service Level Agreements (SLAs)

SLA templates are the special forms of SLAs wherein cloud providers present offers and cloud consumers present requirements before the negotiations for legally signing SLA. SLA mainly consists of three prominent elements, namely, SLA metrics, SLA parameters, and Service Level Objectives (SLOs). They are two types of SLA templates, and they are private SLA which is prescribed for the market customers and public SLA which is formulated for the trade of products in market. C. Redl et al. [9] present an automatic way of matching SLA using machine learning techniques. The intelligent SLA management is possible with proper representation of knowledge from the requirements of the users. Case-based reasoning (CBR) has been used for learning semantically from the requirements. Support vector machine (SVM) has been used as a machine learning technique for providing autonomous SLA matching and autonomous provider selection with cost reduction.

ISP Based on Heuristic and Holistic Perspective

Ali Pahlevan et al. [10] present heuristic and machine learning (ML) approach for provisioning of virtual machines (VMs) into the datacenter. The proposed method in this paper consists of two-level ML approach wherein in the first level of ML, K-means clustering is used for generation of classes and the selection of appropriate classes is performed using heuristic approach. In the second level of ML, value iteration algorithm which is a type of reinforcement learning (RL) is used for classification of classes for provisioning VMs.

ISP Based on Multi-criteria Decision-Making (MCDM)

Analytical Hierarchy Process (AHA) and Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) are the commonly used MCDM methods. AHA provides a comparison of elements in pairwise manner which is structured in hierarchical relationship. Muhammad Umer Wasim et al. [11] have proposed a self-regulated MCDM which used an integrated approach of MCDM and machine learning technique for ranking the service providers. Communality which comes under the category of factor analysis, a type of unsupervised machine learning technique, has been used. Structural Equation Modeling (SEM) has been used for estimation of communality. Commonly used SEM methods are principal factor analysis (PFA) and maximum likelihood. The datasets which had been used for evaluation are customer reviews of cloud service providers (CSP) and simulated dataset of feedback from servers of broker architecture. The benefit of the MCDM with machine learning approach is that it provides ranking based on objective criteria which eliminates error in providing irrelevant services to the customers rather than subjective judgements which is mainly based on insufficient domain knowledge.

ISP Based on Algorithmic Techniques

Service composition is one of the important processes to be performed in multi-cloud environment. Cloud consumers expect the services to be economically feasible. Economically driven service composition is a long process, and it requires exhaustive search. In order to optimize the process of service composition, Samar et al. [12] have presented a deep learning-based service composition (DLSC) framework which is an integration of long short-term network (LSTN) and particle swarm optimization (PSO) algorithm.

10.3.1.3 ISP Based on Procedures of Service Provisioning

The ISP based on procedures of service provisioning involves user request scheduling, service selection procedure, service composition, service monitoring, and orchestration.

ISP Based on User Request Scheduling

Each user request for service to the cloud is considered as a task at the datacenter. Efficient scheduling and service provisioning have to be incorporated at the data center owned by the cloud service provider (CSP) in order to minimize the energy cost at larger scale. Mingxi Cheng et al. [13] proposed a deep reinforcement learning (DRL) approach as a solution to minimize energy cost at the datacenter wherein the user requests for service are scheduled.

ISP Based on Service Selection Procedure

Service selection is the vital procedure in service provisioning. Selecting appropriate services from various vendors in multi-cloud environment is a challenging task. Yan Wang et al. [14] have presented a CC-PSM model which is preference aware service selection model based on customer community. In CC-PSM model, the initial step includes data mining process for categorizing customers based on bipartite network. The second step in this model consists of incorporation of improved hierarchical clustering algorithm which is an unsupervised machine learning technique; the outcome of the second step is to discover consumer community based on preferences. Finally, prediction model is used to predict the customer evaluation on unknown service. The advantage of this model is that it replaces the traditional way of service selection which performs collaborative filtering to find a match between customer requirements and QoS.

ISP Based on Service Composition

Cloud service composition is long term and economic driven. Zhen et al. [15] have proposed a Bayesian-based model to represent the economic perspective of consumer. A novel influence diagram-based model has been presented as a cloud service composition methodology.

ISP Based on Service Monitoring and Orchestration

Service monitoring and orchestration have been achieved using Key Performance Indicators (KPIs). The correlation between resources usage and application of Key Performance Indicators (KPIs) is required for the operation engineer to understand performance bottlenecks, scalability, and degradation of QoS issues. Johannes Grohmann et al. [16] have presented a monitorless model which has a machine learning model trained with platform-related data retrieved from the containerized services to infer KPI. The inferred KPI can be used by the operation engineers to improve the architectural frameworks of multi-cloud in order to fulfill Service Level Objectives (SLOs). The labels used are saturated and non-saturated levels. The machine learning models used in the "monitorless" model are adaptive boosting (AdaBoost) classifier, binary logistic regression, XGBoost, Random Forest (RF), and Neural Network (NN) (Table 10.1).

10.3.2 Comparison of Various Intelligent Middleware for Management of Multi-Cloud Services

A middleware in multi-cloud is defined as a tool which acts as medium between cloud consumer and cloud service provider. A middleware is said to be known as "intelligent" if it has incorporated the computational intelligence techniques, namely, Artificial Intelligence (AI), soft computing, and data mining. Table 10.2 presents the comparison of various intelligent middleware used for management of multi-cloud services.

10.4 Benchmark Case Studies: ISP Provisioning in Multi-Cloud

The two benchmark case studies associated with ISP provisioning in multi-cloud are healthcare and smart city services.

10.4.1 Case Study 1: Multi-Cloud Framework with ISP in Healthcare

Multi-cloud framework has been adopted by healthcare organization to provide agility which is essential to meet the demands of the healthcare sectors for the complete care and support of the patient population. The most popular service healthcare sectors use Software as a Service (SaaS) wherein healthcare acts as a broker to host and maintain applications. Some healthcare sectors have begun to adopt services, namely, Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) in order to deploy cloud native applications.

Ahmed Abdelaziza et al. [30] present a novel model for healthcare services which select the optimal number of virtual machines for processing the medical requests using parallel particle swarm optimization (PPSO), and a prediction model of chronic kidney disease (CKD) is provided using hybrid machine learning techniques, namely, linear regression and Neural Network (NN).

Healthcare requires a large-scale processing of data especially when the data is of image type. Processing large data requires healthcare sectors to adopt multi-cloud systems to provide efficiency, scalability, and high availability. Massive data analysis in multi-cloud of healthcare with more security and less computational costs requires machine learning to be incorporated. Mbarek Marwan et al. [31] proposed a novel approach of enhancing security in multi-cloud of healthcare using machine learning techniques, namely, support vector machines (SVMs) and Fuzzy C-means clustering.

Table 10.2 Comparison of intelligent middleware of multi-cloud services

SI. no	Intelligent middleware	Methodology	Advantage	Disadvantage
1	Cloud management broker (CMB)	Cloud management broker (CMB) is used to manage resources in multi-cloud using two level reinforcement learning (RL) resource allocation algorithm [17]	It provides increased scalability and flexibility in managing resources in multi-cloud with quick response to user requests	The state space size has to be minimized, and scalability algorithms has to be improved using state aggregation mechanisms
2	Daleel – decision-making framework	Daleel is a decision-making framework which provides evidence-based knowledge to customers, and it helps in service selection for customers. Regression-based method has been incorporated in Daleel framework [18]	It provides adaptive decision-making due to machine learning techniques based on response time	The response time of each services of cloud service providers (CSP) has been considered, but other characteristics memory, processor, and behavioral data have not been considered
3	Scalable hierarchical framework	The scalable hierarchical framework is used for resource allocation and power management using deep reinforcement learning (DRL) [19]	It reduces online computational complexity and improves parallelism	Experimental results are provided only with Google cluster traces
4	Brokerage-based cloud service selection	K-nearest neighbor (KNN) is used to search the services of cloud service provider (CSP) index created using B+ tree for providing the cloud service selection [20]	The efficiency of the algorithm is evaluated using real-time and synthetic data	The service level agreements (SLAs) negotiated by the customer at the request time have not been considered
5	Agent-based intelligent cloud service selection	The agent is incorporated with intelligence using unsupervised machine learning technique known as Q learning which provides the appropriate services to the customers through performance checking from the customer feedback [21]	The customer feedback has been considered for service selection which enhances the system to an errorless state	The detailed architecture of the agent has not been explored

(continued)

Table 10.2 (continued)

SI. no	Intelligent middleware	Methodology	Advantage	Disadvantage
6	Trust-based agent learning model for service composition (TALMSC)	TALMSC consists of two-staged fuzzy C-means learning (FCM) mechanism. The efficiency of the mechanism has been tested on JADE which is a multi-agent-based learning system wherein four related mechanism has been tested, namely, two-staged improved FCM, FCM, K- means clustering, and random transaction [22]	TALMSC improves the user satisfaction	The integration of the system with service matching, prediction, and forecasting has yet to be considered and developed
7	Trust enabled self-learning agent model for service matching (TSLAM)	TSLAM is a three-layered agent model which consists of brokers with learning module developed using decision tree algorithm [23]	TSLAM enhances the transaction success rate and improves the user satisfaction	There is saturation level, i.e., the number of cloud service providers to be handled by brokers is limited which has to be increased to enhance efficiency
8	QoS prediction model	A Quality of Service (QoS) prediction model for service composition based on hidden Markov model (HMM) [24]	QoS satisfied service provisioning is provided to the customers through HMM-based QoS prediction model	The prediction model is based on only homogenous services, and heterogenous services were not considered
9	NLUBroker	A reinforcement learning (RL)-based agent is incorporated in natural language understanding (NLU) broker system which maps the customer requirements to the cloud services available from the requirements analysis provided in user language [25]	This NLUBroker provides flexibility to users in providing services requirements	The exploration of natural language processing (NLU) is yet to be done

(continued)

Table 10.2 (continued)

SI. no	Intelligent middleware	Methodology	Advantage	Disadvantage
10	Preference-based cLoud service recommender (PuLSaR)	PuLSar is a cloud service recommender using multi-criteria decision-making approach [26]	It enhances the capabilities of cloud service broker with increased scalability	This recommender has yet to be deployed in real-time cloud environments with exposure of handling heterogenous services in distributed and dynamic environment
11.	Service-oriented broker framework	Service-oriented broker framework consists of three modules, namely, user portal for requirements gathering, service discovery and composition, and service provisioning [27]	The framework improves the main functionalities of cloud	The broker has to be incorporated with machine learning techniques to make it more intuitive in decision-making
12	Cloud broker framework for infrastructure service discovery using semantic network	The broker-based cloud framework provides a user interface wherein the user specifies the request in numerical terms and the broker constructs ontologies and semantic network is formed based on the intersection with the discovered services [28]	The broker-based approach provides a good accuracy in providing service recommendation	The user requirements are specified using numerical representation, and linguistic representation is not accepted
13	Broker-based cloud service model	The broker-based cloud service model executes functions, namely, service discovery and service composition for provisioning [29]	The broker-based cloud service model performs additional services, namely, ranking based on QoS specification	The incorporation of intelligence in broker for federated cloud environment is yet to be developed

10.4.2 Case Study 2: Multi-Cloud Framework with ISP in Industrial IOT and Smart City Services

Basheer Qolomany et al. [32] have proposed an intelligent polynomial time heuristic which selects machine learning models from a superset of model which maximizes the trustworthiness of the model. This selected model is used as prediction model by cloud service providers (CSP) in order to process the data obtained from industrial-based IOT devices and smart city services connected to the cloud.

10.5 Review on State-of-the-Art ML Algorithms for Service Provisioning in Multi-Cloud Environment

Machine learning (ML) techniques come under the category of Artificial Intelligence (AI). Deep learning (DL) is the sub-category of machine learning (ML). Intelligent service provisioning is possible only through the incorporation of machine learning (ML) technique at any procedural level of provisioning or at any service level of provisioning. Machine learning (ML) technique is of two types: supervised machine learning techniques and unsupervised machine learning techniques.

10.5.1 Neural Network (NN)

Working Principle: Neural Network (NN) consists of input layer, hidden layer, and output layer. The Neural Network (NN) is a machine learning algorithm, and its structure is inspired from the biological neurons. It consists of collections of interconnected neurons. The input layer provides input to the hidden layer which computes the output based on activation function, and the final output layer produces the result as the weighted aggregation of all output of hidden layers [33].

Usage: Neural Network ML model is used for multi-cloud service monitoring and orchestration based on Key Performance Indicators (KPIs).

10.5.2 Reinforcement Learning

Working principle: Reinforcement learning (RL) is a type of machine learning technique which is used to control a system to improve its performance stated in numbers. The learner incorporated with RL learns f through the trial-and-error method when exposed to the dynamic environment. There are two approaches used to provide solutions for reinforcement learning problems. The first approach is to search through the dynamic environment wherein genetic algorithms are proved to be effective. The second approach is to use statistical techniques and dynamic programming methods [34].

Usage: The reinforcement learning (RL) ML model is used for multi-cloud user request scheduling in an energy-efficient manner and for provisioning of VMs.

10.5.3 Support Vector Machine (SVM)

Working principle: Support vector machine uses the hyperplane to classify the service instances. The projection of instances to higher-dimensional space occurs if the instances are of non-linear type. Different types of kernels can be used in SVM, namely, Gaussian kernel, polynomial kernel, and radial basis function (RBF) kernel [35].

Usage: The SVM ML model is used for autonomous service selection in multi-cloud based on service level agreement (SLA) attributes.

10.5.4 Deep Belief Network (DBN)

Working principle: The construction of deep belief network (DBN) is in the form of stack wherein each individual unit in stack is built using restricted Boltzmann machine. The structure of restricted Boltzmann machine is a feed forward graph structure with two layers, namely, visible layer which is Gaussian or binary units and hidden layer which is binary unit [36].

Usage: The important aspect to be maintained in multi-cloud is workload management which is provided intelligently using deep belief network (DBN) model.

10.5.5 Principal Factor Analysis (PFA)

Working principle: Principal factor analysis (PFA) is mainly used to reduce the dimensionality of the dataset based on the correlation factor. It mainly focuses on the reduction of non-diagonal elements in the dataset [37].

Usage: The service selection in multi-cloud for user is made at ease by incorporating PFA in the multi-cloud framework which provides user with relevant services based on reviews provided by other users.

10.5.6 Random Forest, AdaBoost and XgBoost

Working principle: Random Forest is an ensemble-based machine learning algorithm. It is a homogenous ensemble classifier since it consists of ensemble with one type of machine learning model, namely, decision trees. It combines the results of many decision trees rather than combining predictions of different machine learning models [38]. Adaptive Boosting (AdaBoost) classifier is a sequential based ensemble classifier, which rectifies error in consequent iterations by giving more weight to the error samples [39]. Extreme Gradient Boosting (XgBoost) classifier is an ensemble classifier which works in parallel and is faster in execution than AdaBoost classifier [40].

Usage: The Random Forest (RF), Adaptive Boosting (AdaBoost),Extreme Gradient Boosting (XgBoost) model is mainly used for service monitoring and orchestration in multi-cloud.

10.5.7 Hierarchical Clustering

Working principle: Hierarchical clustering comes under the unsupervised machine learning algorithm. In hierarchical clustering, the data is grouped into clusters which forms a tree, and the split or merge operations cannot be restructured. There are two types of hierarchical clustering, namely, agglomerative clustering and divisive clustering. The outcome of the hierarchical clustering is in the form of dendrogram which is in the form of tree [41, 42].

Usage: The service selection in multi-cloud based on Quality of Service (QoS) and customer requirements is provided using hierarchical clustering ML model.

10.6 Framework for Intelligent Service Provisioning in Multi-Cloud

The intelligent service provisioning model in the framework consists of process models, namely, service request scheduler, service composition, service monitoring and orchestration, and service selection whose data are analyzed using machine learning (ML)-based decision-making system, and data from cloud service providers, cloud consumers, and process models are stored in knowledge base which is in turn used by decision-making system and process models to satisfy consumer requests.

The framework for intelligent service provisioning in multi-cloud is provided in Fig. 10.1.

The cloud consumer requests are processed by the intelligent service provisioning (ISP) model. The outcomes of process models are as follows:

1. Service request scheduler – sends requests to the corresponding process model
2. Service Composition – The service composition suggestions provided by decision-making system: Services of cloud service providers (CSP) appropriate to user requests
3. Service monitoring and orchestration: The service request status
4. Service selection: Recommendation of services offered by CSP appropriate to user requests provided by the decision-making system

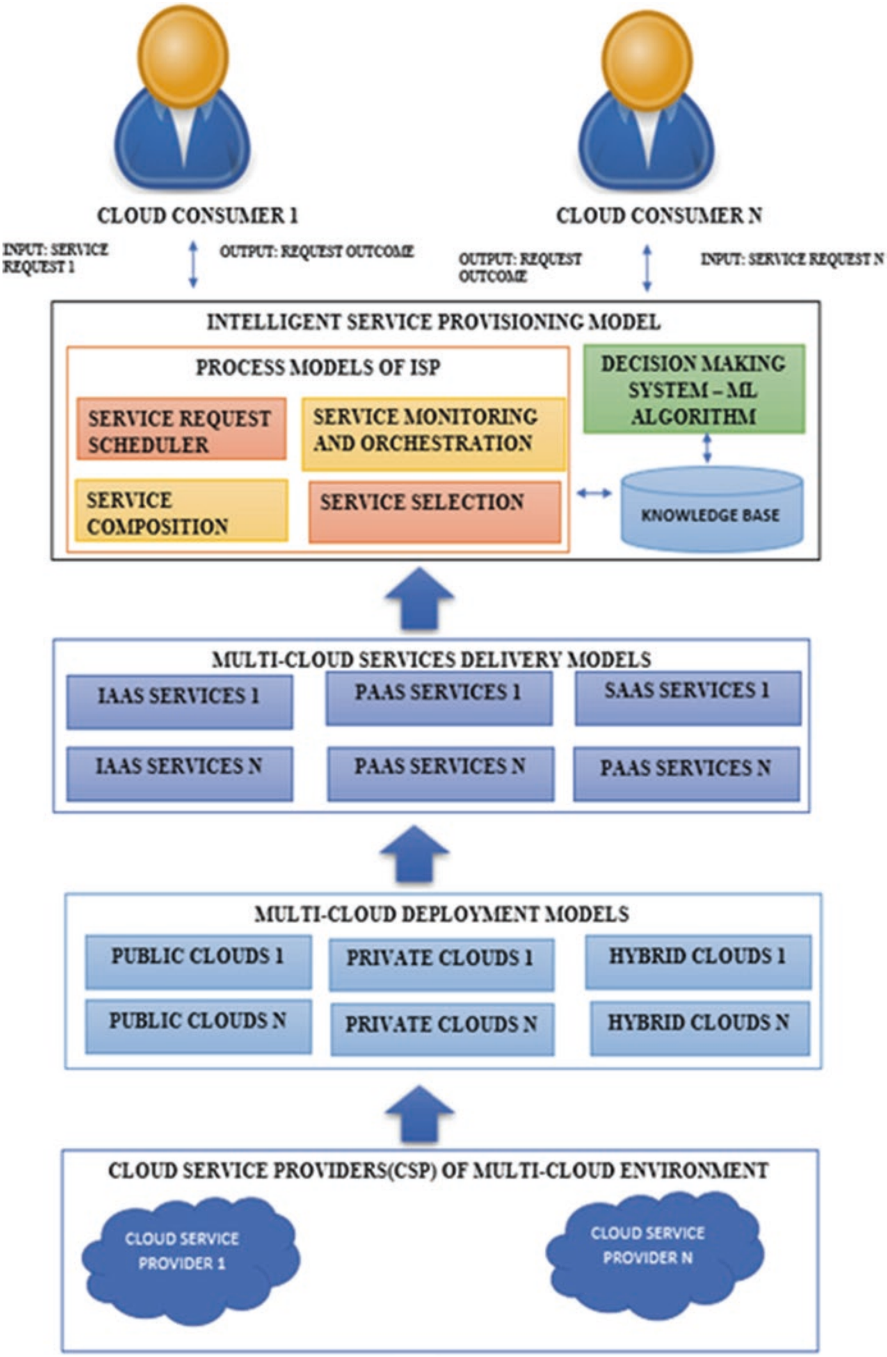


Fig. 10.1 Framework of intelligent service provisioning (ISP)

10.7 Challenges and Future Prospects

The challenges and future prospects of intelligent service provisioning (ISP) in multi-cloud environment have been addressed in this section.

10.7.1 Challenges of Intelligent Service Provisioning (ISP) in Multi-Cloud Environment

1. Provisioning of services with security is a challenge in multi-cloud environment especially sectors associated with finance and healthcare.
2. The costs of services vary, and provisioning of services with optimal cost is a challenging task as it integrates services from various vendor.
3. Service monitoring is a challenging task in multi-cloud environment because the decision has to be made to handle performance degradation.

10.7.2 Future Prospects of Intelligent Service Provisioning (ISP) in Multi-Cloud Environment

1. Integration of block chain with ISP: Security can be enhanced by integrating block chain technology to the ISP framework in multi-cloud environment.
2. Cost: In order to provide optimal cost for services, each services of different vendors has to be registered, and the information has to be stored in knowledge base. The knowledge base has to be dynamic and updated whenever changes in services cost occur.
3. Incorporation of AI: The Deep Learning (DL) algorithm has to be incorporated as decision maker for service monitoring since DL can handle large complex data, and it is faster to provide solutions. DL can be mainly used in healthcare multi-cloud systems wherein complex images have to be stored, processed, analyzed and maintained in cloud.

10.8 Conclusion

Multi-cloud environment provides integrated services from various cloud service providers (CSP). Intelligent service provisioning (ISP) which incorporates machine learning (ML) techniques helps the cloud consumers to procure appropriate services as per the need of the customers. This chapter provides classification of ISP with respect to certain attributes, comparison of various middleware tools used in ISP, and case studied related to ISP.

References

1. Youssef, A. E. (2012). Exploring cloud computing services and applications. *Journal of Emerging Trends in Computing and Information Sciences, 3*(6). ISSN 2079-8407.
2. Panda, S. K., Gupta, I., & Jana, P. K. (2019), Task scheduling algorithms for multi-cloud systems: Allocation-aware approach. *Information Systems Frontiers*. https://doi.org/10.1007/s10796-017-9742-6.
3. Qiu, F. (2016). A deep learning approach for VM workload prediction in the cloud IEEE SNP 2016, Shanghai, China.
4. Dutreilh, X., Kirgizov, S., Melekhova, O., Malenfant, J., Rivierre, N., & Truck, I. (2011). Using reinforcement learning for autonomic resource allocation in clouds: Towards a fully automated workflow. In *7th international conference on autonomic and autonomous systems (ICAS'2011)* (pp. 67–74). IARIA.
5. Barrett, E., Howley, E., & Duggan, J. (2013). Applying reinforcement learning towards automating resource allocation and application scalability in the cloud. *Concurrency and Computation: Practice and Experience, 25*(12), 1656–1674.
6. Miyazawa, T., Kafle, V. P., & Harai, H. (2017). Reinforcement learning based dynamic resource migration for virtual networks. In Proc. IFIP/IEEE symposium on integrated network and service management (IM) (pp. 428–434).
7. Rahma, D. (2017). A novel cloud services recommendation system based on automatic learning techniques. International conference on new trends in computing sciences (ICTCS) https://doi.org/10.1109/ICTCS.2017.58.
8. Bala, K., & Vashist, S. (2014). Machine learning based decision making by brokers in cloud computing. *International Journal of Application or Innovation in Engineering & Management (IJAIEM), 3*(7), 269–273. ISSN 2319-4847.
9. Redl, C., Breskovic, I., Brandic, I., & Dustdar, S. (2012). Automatic SLA matching and provider selection in grid and cloud computing markets. In Proceedings of the 13th ACM/IEEE international conference on grid computing (Grid '12) (pp. 85–94). Beijing, China.
10. Pahlevan, A., Zapater, M., & Atienza, D. (2017). Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers. IEEE transactions on computer-aided design of integrated circuits and systems.
11. Wasim, M. U., Ibrahim, A. A. Z. A., Bouvry, P., Limba, T. (2018). Cloud service providers optimized ranking algorithm based on machine learning and multi-criteria decision analysis. https://doi.org/10.20944/preprints201801.0125.v1.
12. Haytamy, S, & Omara, F. (2020). A deep learning based framework for optimizing cloud consumer QoS-based service composition. Springer, Computing. https://doi.org/10.1007/s00607-019-00784-7.
13. Cheng, M. (2018). DRL-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers. IEEE.
14. Wang, Y., Zhou, J.-T., & Tan, H.-Y. (2015). CC-PSM: A preference-aware selection model for cloud service based on consumer community. Hindawi Publishing Corporation Mathematical Problems in Engineering (Vol. 2015, Article ID 170656, 13 p). https://doi.org/10.1155/2015/170656.
15. Ye, Z., Bouguettaya, A., & Zhou, X. (2012). QoS-aware cloud service composition based on economic models. In International conference on service-oriented computing. Springer (pp. 111–126).
16. Grohmann, J., Nicholson, P. K., Iglesias, J. O., & Lugones, D. (2019). Monitorless: Predicting performance degradation in cloud applications with machine learning association for computing machinery. ACM ISBN 978-1-4503-7009-7/19/12. https://doi.org/10.1145/3361525.3361543.
17. Antonio, P., Stefano, B., Francisco, F., Alessandro, G., Guido, O., Martina, P., & Vincenzo, S. (2015). Resource Management in Multi-Cloud Scenarios via reinforcement learning. Proceedings of the 34th Chinese control conference, Hangzhou, China.

18. Samreen, F., Elkhatib, Y., Rowe, M., & Blair, G. S. (2016). Daleel: Simplifying cloud instance selection using machine learning. NOMS 2016–2016 IEEE/IFIP network operations and management symposium, Istanbul (pp. 557–563). https://doi.org/10.1109/NOMS.2016.7502858.
19. Liu, N., Li, Z., Xu, Z., Xu, J., Lin, S., Qiu, Q., Tang, J., & Wang, Y. (2017). A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning. arXiv:1703.04221v2 [cs.DC].
20. Sundareswaran, S., Squicciarini, A., & Lin, D. (2012). A brokerage-based approach for cloud service selection. IEEE fifth international conference on cloud computing.
21. Rabbani, I. M., Muhammad, A., & Martinez Enriquez, A. M. (2013). Intelligent cloud service selection using agents. In P. Meesad, H. Unger, & S. Boonkrong (Eds.), *The 9th international conference on computing and information technology (IC2IT2013). Advances in intelligent systems and computing* (Vol. 209). Springer. https://doi.org/10.1007/978-3-642-37371-8_14
22. Li, W., Cao, J., Hu, K., Xu, J., & Buyya, R. (2019). A trust-based agent learning model for service composition in Mobile cloud computing environments, Special section on mobile service computing with internet of things. https://doi.org/10.1109/ACCESS.2019.2904081.
23. Li, W. , Cao, J., Hu, K., Xu, J., & Buyya, R. (2019). TSLAM: A trust-enabled self-learning agent model for service matching in the cloud market. *ACM Transactions on Autonomous and Adaptive Systems,13*(4).
24. Wu, Q., Zhang, M., Zheng, R., Lou, Y., & Wei, W. (2013). A QoS-satisfied prediction model for cloud-service composition based on a hidden markov model. *Hindawi Publishing Corporation, Mathematical Problems in Engineering*. https://doi.org/10.1155/2013/387083.
25. Xu, L., Iyengar, A., & Shi, W. (2019). NLUBroker: A flexible and responsive broker for cloud-based natural language understanding services. HotCloud'19: Proceedings of the 11th USENIX conference on hot topics in cloud computing.
26. Patiniotakis, I., Verginadis, Y. & Mentzas, G.(2015), PuLSaR: Preference-based cloud service selection for cloud service brokers. *Journal of Internet Services and Applications*. https://doi.org/10.1186/s13174-015-0042-4.
27. Nagarajan, R., & Thirunavukarasu, R. (2020). Service-oriented broker for effective provisioning of cloud services–a survey. *International Journal of Computing and Digital Systems, 10*, 2–17.
28. Nagarajan, R., & Thirunavukarasu, R. (2017). A cloud broker framework for infrastructure service discovery using semantic network. *International Journal of Intelligent Engineering and Systems, 11*(3). https://doi.org/10.22266/ijies2018.0630.02.
29. Rajganesh, N., & Thirunavukarasu, R. (2016). A review on broker based cloud service model, CIT. *Journal of Computing and Information Technology, 24*(3), 283–292.
30. Abdelaziz, A., Elhoseny, M., Salama, A. S., & Riad, A. M. (2018). A machine learning model for improving healthcare services on cloud computing environment. *Measurement, 119*, 117–128. https://doi.org/10.1016/j.measurement.2018.01.022
31. Marwan, M., Kartit, A., & Ouahmane, H. (2018). Security enhancement in healthcare cloud using machine learning. *Procedia Computer Science*.
32. Qolomany, B., Mohammed, I., Al-Fuqaha, A., Guizan, M, & Qadir, J. (2020). Trust-based cloud machine learning model selection for industrial IoT and smart city services. arXiv:2008.05042v1 [cs.LG].
33. Trivedi, M., & Somani, H. (2016). A survey on resource provisioning using machine learning in cloud computing. IJEDR. ISSN: 2321-9939.
34. Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*.
35. Smola, A. J., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*, 199–222.
36. Khan, A., et al. (2016). Deep belief networks. https://doi.org/10.13140/RG.2.2.17217.15200.
37. Chapter 420 - factor analysis, ncss statistical software. NCSS.com.
38. Biau, G. (2012). Analysis of a random forests model. *Journal of Machine Learning Research*.

39. Ma, C., Chen, F., Liu, J., & Duan, J. (2014). A new method of cloud detection based on cascaded AdaBoost. 8th international symposium of the digital earth (ISDE8) IOP publishing. IOP conf. series: Earth and environmental science. https://doi.org/10.1088/1755-1315/18/1/012026.
40. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. ACM. ISBN 978-14503-4232-2/16/08. https://doi.org/10.1145/2939672.2939785.
41. Rani, Y., & Rohi, H. (2013). A study of hierarchical clustering algorithm. *International Journal of Information and Computation Technology, 3*(11), 1225–1232. ISSN 0974-2239.
42. Pawar, P. S., Sajjad, A., Dimitrakos, T., & Chadwick, D. W. (2015). Security-as-a-service in multi-cloud and federated cloud environments. In *IFIP international conference on trust management* (pp. 251–261). Springer.

Chapter 11
Fuzzy-Based Workflow Scheduling in Multi-Cloud Environment

J. Angela Jennifa Sujana, R. Venitta Raj, and T. Revathi

11.1 Introduction

Nowadays with the introduction of many Internet of Things (IoT)-based real time applications, the need for integrating those applications with the cloud environment has become essential. Also, there is a great need for accessing the data from IoT-based smart applications for doing meaning analysis for inferring useful information and automation of the process. At the same time, the user may want to use different cloud services for their application. In such situations, multi-cloud environment is the only solution. Hence, with multi-cloud environment the user or the developer is having enormous freedom in choosing the best cloud service provider from a set of cloud providers.

Multi-cloud computing aggregates large pool of resources and share them among vast cloud users. It is a gifted technique for systems integration. Multi-cloud computing can also be defined as a new version of collaborative environment, in which scalable and virtualized resources are provided as a service over the Internet [1]. In general, the service workflow is organized as the collection of services to ease the requirements and automation of large-scale distributed systems [2]. Although most commercial cloud services are operated and owned by distributed and heterogeneous organizations in multi-cloud computing environments, the inherent uncertainty and unreliability of large-scale organizations often pose threats to the operation of workflow applications. Hence, in addition to execution time and cost factors, we need to think about trust factor.

J. A. J. Sujana (✉) · R. V. Raj · T. Revathi
Department of Information Technology, Mepco Schlenk Engineering College, Sivakasi, Tamil Nadu, India
e-mail: ang_jenefa@mepcoeng.ac.in; venittaraj@mepcoeng.ac.in; trevathi@mepcoeng.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_11

Initially this multi-cloud environment is not supported for the cloud users, because of the vendor lock-in problem [3]. The Open Virtualization Format (OVF) [4, 5] is a vendor-independent format which supports portability and deployment with different vendors. With the collaborative effort of Dell, Microsoft, HP, IBM, VMWare, and XenSource, the system images can be imported and deployed on multiple platforms, thus enabling cross-platform portability. The introduction of this platform-independent virtualization format is one reason for the emergence of multi-cloud environment. Also, users and developers started to use multi-cloud based on their optimal quality criteria satisfaction.

In many of the applications which use the cloud services, the Quality of Service (QoS) plays a vital role. The QoS mainly depends on the user request and the application being used. It can be noted many enterprise applications and IoT-based smart applications can be represented as workflows [6]. Rajganesh et al. [7] have proposed a service context-aware cloud broker which uses the service details from the contextual information of cloud services and computes service similarities on the basis of QoS values. Each task in the workflow represents a module in the enterprise application. Normally these workflows are represented as Directed Acyclic Graph (DAG). Each node in the DAG represents the task in the workflow. Fuzzy decision-based models can be adopted for multi-cloud environments [8, 9]. A fuzzy logic-based intelligent cloud broker is proposed to find the imprecise state of the inexperienced cloud user when deciding the infrastructure service requirements [10]. An intelligent cloud broker which uses the MapReduce framework for the effective pre-processing of cloud users' feedback was proposed by Rajganesh Nagarajan et al. [11]. Also, they propose a fuzzy logic-based trust evaluation system for accepting the user's feedback in terms of fuzzy linguistic [12].

This chapter focuses on finding the suitable cloud service provider based on the Quality of Service of the user or developer of enterprise applications and IoT-based smart applications. The main QoS parameters considered are time, cost, and trust. This multi-objective QoS-based workflow scheduling is based on fuzzy model. The fuzzy membership function is defined for each objective, and collectively the decisions are made. Also, hence this chapter proposes a multi-objective QoS-based workflow scheduling in multi-cloud environment with fuzzy logic with the fuzzy logic-based workflow scheduling (FLWS) algorithm.

11.2 System Architecture

The system architecture of the proposed fuzzy-based workflow scheduling in multi-cloud environment is represented in Fig. 11.1. This chapter envisages a fuzzy-based multi-objective model for dispatching the tasks in the workflow to suitable cloud provider in accord with the user Quality of Service (QoS). The user will submit the workflow, which will be handled by the QoS-based Resource Management layer. This module gets the workflow from the user and the QoS weightage for each objective. The objectives addressed are time, cost, and trust. This QoS-based Resource Management module can also be viewed as an agent, helping in the user for

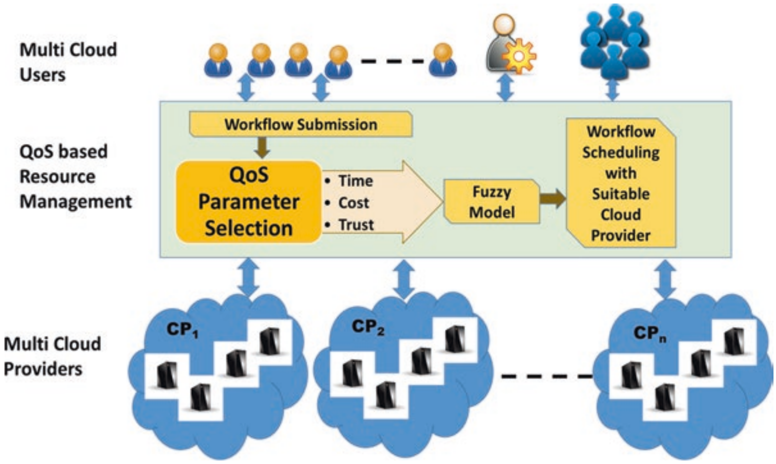
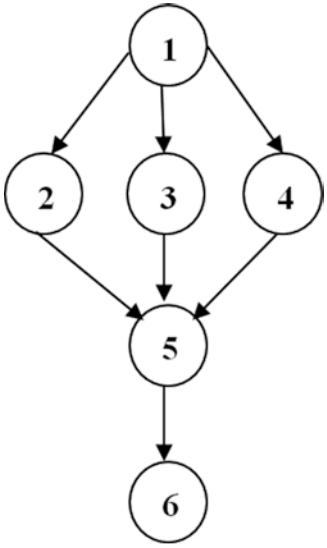


Fig. 11.1 System architecture

Fig. 11.2 Sample workflow



identifying the suitable services for them. The fuzzy model is used to make the decision in selecting the suitable cloud provider for the workflow. Here we provide a fine-grained selection module. That is, the user can select different services for each task in the workflow from different cloud providers. For each task the best suitable cloud provider's service will be considered, and it will be scheduled accordingly.

Figure 11.2 represents a sample workflow, which can be represented as the Directed Acyclic Graph (DAG) [13]. The sample workflow consists of six tasks represented by the nodes, and the edges between the tasks represent the dependency among the tasks. The edges represent the data transfer time. If both the ancestor and descendant task get executed in the same virtual machine, then the data transfer time between these tasks becomes zero.

11.3 Multi-objective Fuzzy Decision Model

In this chapter, we introduce fuzzy-based decision model for selecting the suitable cloud provider in multi-cloud. The clouds' providers considered in the multi-cloud environment are denoted as CP = $\{cp_1, cp_2, \ldots, cp_n\}$. The workflow is modelled as a directed acyclic graph, W = (TK, E), where TK denotes the tasks' set TK = $\{tk_1, tk_2, \ldots, tk_m\}$ and each task $tk_i \in$ TK, $1 \leq i \leq m$ needs a cloud service from the cloud provider. Similarly E represents the set of communication link edges between tasks, and each edge $e(i,j) \in E$ represents that the task tk_j depends on the task tk_i, i.e., task tk_i should be executed prior to the execution of the task tk_j. The task with an in-degree 0 is designated as tk_{entry} entry task, and the task without-degree 0 is designated as tk_{exit} exit task. The parent tasks of the task tk_i are represented *ancestor*(tk_i), and the child tasks of a task tk_i are represented by *descendant*(tk_j). In multi-cloud environment the provider has enormous services. The user pays for the services.

This aids in selecting the best optimal cloud service provider for the tasks in the workflow according to the user's QoS demand. The fuzzy decision model will consider three objectives, namely, time, cost, and trust.

Let n be the total no. of candidate cloud providers in multi-cloud. There are a set of S_j^i services, which are available for each task in the workflow. More specifically S_j^i denotes the service from cloud provider cp_j, delivered for task tk_i. Hence for the same task tk_i there will be "m" number of services available from the cloud providers CP. These services may have varied processing capabilities, processing time, and prices. Let t_j^i be the total processing time and C_j^i be the total processing cost for the ith task on the jth service.

To get optimized solution on multi-objective, we propose fuzzy model as follows. Using max-min as the operator [14], the membership function of objectives can be formulated by separating each objective into its maximum values and minimum values. The membership function for time and cost objectives is given by Eq. (11.1) and for trust is given by Eq. (11.2).

$$u(x) = \begin{cases} 1, Z_K \leq Z_K^{\min} \\ \dfrac{Z_K^{\max} - Z_K(x)}{Z_K^{\max} - Z_K^{\min}}, Z_K^{\min} \leq Z_K \leq Z_K^{\max} \\ 0, Z_k^{\max} \leq Z_k \end{cases} \tag{11.1}$$

$$u(x) = \begin{cases} 0, Z_l \leq Z_l^{\min} \\ \dfrac{Z_l(x) - Z_l^{\min}}{Z_l^{\max} - Z_l^{\min}}, Z_l^{\min} \leq Z_l \leq Z_l^{\max} \\ 1, Z_l^{\max} \leq Z_l \end{cases} \tag{11.2}$$

Here the workflow scheduling problem is considered as a multi-objective planning problem which focuses in optimizing the conflicting objectives [15]. The following notations are used to represent the objectives. The deadline of a task is denoted by D, budget that can be spent by B, and trust level by Tr. The time taken by jth service assigned for ith task is denoted by t_j^i. Let total number of tasks be n and total number of cloud provider services available for the ith task be m_i. To denote assigning the ith task to jth service is given by y_j^i; if the jth service is assigned for ith task, then $y = 1$; else $y = 0$.

The objectives are thus described as follows in Eq. (11.3a), (11.3b), and (11.3c).

$$Minimize : Z_1 = \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_j^i c_j^i \tag{11.3a}$$

$$Minimize : Z_2 = \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_j^i t_j^i \tag{11.3b}$$

$$Maximize : Z_3 = \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_j^i tr_j^i \tag{11.3c}$$

$$\text{Subject to} : Z_1 \;''\; B, Z_2 \;''\; D, Z_3 \;''\; Tr$$

where Z_1 and Z_2 represent the minimization objective for time and cost and Z_3 represents the maximization objective for trust. Our proposed fuzzy logic-based workflow scheduling (FLWS) model considers three different objectives related to time, cost, and trust constraint [10].

11.3.1 Membership Function for Trust Evaluation

The general trust metric, which is the combination of direct trust (DT) and recommendation trust (RT), can be defined as given in Eq. (11.4),

$$Tr(S_i) = w_i * \mathrm{DT}(S_i) + (1 - w_i) * \mathrm{RT}(S_i) \tag{11.4}$$

where the weight w_i assigned for DT is calculated by

$$w_i = 1 - \frac{1}{e^k} \tag{11.5}$$

where k is the number of times the ith service is used by the client user. The direct trust (DT) is calculated by Eq. (11.6).

$$\mathrm{DT}(S_i) = \frac{n_i + 1}{N_i + 2} \tag{11.6}$$

where $\mathrm{DT}(S_i)$ is the direct trust of the ith service which the active user experiences based on the history and $\mathrm{RT}(S_i)$ is the recommendation trust of ith service by other users. w_i is the weight of the direct trust and recommendation trust which is calculated as in Eq. (11.5).

Recommendation trust value can be calculated as given in Eq. (11.7). Here we consider the rating given by other users for a cloud provider's service. The user can record a rating in the 1–5 scale. Value 1 for poor service and value 5 for the best service from the cloud provider. A sample rating table with five users for four cloud providers is shown in Table 11.1. The missing values are represented by ? and it is calculated by the recommendation trust. The recommendation trust is calculated as the weighted sum of the user's rating for the service.

$$\mathrm{RT}\left(S_i\right) = avg\left(v_a\right) + \frac{\sum_{i=1}^{n} w_{ai}\left(v_{ij} - \overline{v_i}\right)}{\sum_{i=1}^{n} \left|w_{ai}\right|} \tag{11.7}$$

$avg(v_i)$ can be calculated as follows:

$$avg\left(v_i\right) = \frac{1}{\left|S_i\right|} * \sum_{j \in S_i} v_{ij} \tag{11.8}$$

$avg(v_i)$ be the average rating given by user i. v_{ij} is the rating by user i to the jth cloud provider's service. We use the *Pearson Correlation Coefficient (PCC)* for calculating the similarity between user a and i. This can be calculated as using the Eq. (11.9).

$$w_{ai} = \frac{\sum_{j \in S}\left(v_{aj} - \overline{v_a}\right)\left(v_{ij} - \overline{v_i}\right)}{\sqrt{\sum_{j \in S}\left(v_{aj} - \overline{v_a}\right)^2 \left(v_{ij} - \overline{v_i}\right)^2}} \tag{11.9}$$

Membership function for trust calculation is formulated as given in Eq. (11.10).

$$U_{tk_i}\left(tr_x\right) = \begin{cases} \dfrac{tr_x^i - tr_{\min}^i}{tr_{\max}^i - tr_{\min}^i}, & if\ tr_{\min}^i \le tr_x^i \le tr_{\max}^i \\ 0, & if\ tr_{\min}^i = tr_x^i \\ 1, & if\ tr_{\max}^i = tr_x^i \end{cases} \tag{11.10}$$

Table 11.1 Sample ratings by the users for the cloud services

	Us_1	Us_2	Us_3	Us_4	Us_5
CP_1	?	2	3	4	5
CP_2	3	4	5	3	?
CP_3	1	5	?	2	2
CP_4	5	1	2	5	5

11.3.2 Membership Function for Execution Time

Using max-min as the operator membership function for execution time is calculated as given in Eq. (11.11).

$$U_{tk_i}\left(t_x\right)=\begin{cases}\dfrac{t_{\max}^i-t_x^i}{t_{\max}^i-t_{\min}^i}, & if\, t_{\min}^i\le t_x^i\le t_{\max}^i\\ 0, & if\, t_{\max}^i=t_x^i\\ 1, & if\, t_{\min}^i=t_x^i\end{cases} \tag{11.11}$$

11.3.3 Membership Function for Execution Cost

Similarly, the membership function for cost the user has to pay for the service is calculated as given in Eq. (11.12).

$$U_{tk_i}\left(c_x\right)=\begin{cases}\dfrac{c_{\max}^i-c_x^i}{c_{\max}^i-c_{\min}^i}, & if\, c_{\min}^i\le c_x^i\le c_{\max}^i\\ 0, & if\, c_{\max}^i=c_x^i\\ 1, & if\, c_{\min}^i=c_x^i\end{cases} \tag{11.12}$$

11.3.3.1 Fuzzy Decision

To convert to crisp model from the fuzzy model, we use max-min as the operator as follows:

$$Maximize \quad \lambda_i$$

$$U_{tk_i}\left(t_x\right)\ge\lambda_i, U_{tk_i}\left(c_x\right)\ge\lambda_i, U_{tk_i}\left(tr_x\right)\ge\lambda_i \tag{11.13}$$

The overall satisfaction degree can be defined as the minimum of overall satisfaction of the given membership values for the trust, time, and cost.

The overall satisfaction by the fuzzy decision is given in Eq. (11.14).

$$\lambda_j^i=\min\left\{U_{tk_i}\left(t_x\right), U_{tk_i}\left(c_x\right), U_{tk_i}\left(tr_x\right)\right\} \tag{11.14}$$

The selection of cloud provider's service for task is given by the maximum of all degrees of satisfaction, as defined in Eq. (11.15),

$$\lambda_k^i = \max\left\{\lambda_j^i\right\} \tag{11.15}$$

However, the user may want to specify their own preference among the objectives time, cost, and trust. This can be done by adding the weight factor for each objectives time, cost, and trust. Thus, we introduce a weight factor for objectives time, cost, and trust. They are represented as w_t, w_c, and w_{tr},respectively. The final selection of the cloud providers' service is done by the weighted arithmetic mean operator and given in Eq. (11.16).

$$Maximize U * W' \tag{11.16}$$

where W is the weight factor for objectives time, cost, and trust. $W = \{w_t, w_c, w_{tr}\}$. The sum of the weight factor must be 1. i.e. $w_t + w_c + w_{tr} = 1$ and the weight factor values must be in the interval [0,1].

11.4 Schedule Primitives

To schedule the workflow with minimum execution time, we use the Earliest Completion Time (ECT) of a task with a cloud provider as the heuristic technique. In this regard the following primitives are considered.

The execution time of the task tk_i on the jth CP is termed as $\mathrm{ET}(tk_i, cp_j)$, and it is calculated using Eq. (11.17). The computation capacity of the service S_j in cloud provider cp_j is denoted as $x(cp_j)$. The execution time of a task tk_i is denoted by $x(tk_j)$.

$$\mathrm{ET}\left(tk_i, cp_j\right) = \frac{x\left(tk_i\right)}{x\left(cp_j\right)} \tag{11.17}$$

The average execution time of the task tk_i is given in Eq. (11.18).

$$\mathrm{ET}\left(tk_i\right) = \frac{\sum_{j=1}^{m} \mathrm{ET}\left(tk_i, cp_j\right)}{m} \tag{11.18}$$

The communication cost of an edge $e(i,j)$ can be calculated as given in the Eq. (11.19). Since the services are available from different cloud providers, there will be some latency in hosting the workflow tasks' in the virtual machines of the cloud providers. The late(cp_j) denotes the latency that a cloud provider will have for executing any task. The $x(e(i,k))$ is the amount of data transferred between the two tasks i and k. bw_k is the bandwidth of the link between the cloud providers.

$$\mathrm{CC}\left(tk_i, tk_j\right) = \sum_{j=1}^{m} \mathrm{late}\left(cp_j\right) + \frac{x\left(e\left(i,j\right)\right)}{\sum_{k=1}^{n} \mathrm{bw}_k} \tag{11.19}$$

The Earliest Start Time (EST) is the earliest possible start time of a task on a cloud provider cp_j. This can be computed by the Eq. (11.20)

$$\begin{aligned} EST\left(tk_i, cp_j\right) = max\{ReadyTime\left(tk_i, cp_j\right), \\ \max_{t_m \in \mathrm{ancestor}(t_i)} \left\{ECT\left(tk_m, cp_j\right) + CC\left(tk_m, tk_i\right)\right\} \end{aligned} \tag{11.20}$$

$$\mathrm{ReadyTime}\left(tk_{\mathrm{entry}}, cp_j\right) = 0$$

The ReadyTime(tk_i, cp_j) denotes the time in which the jth VM is ready for executing the task tk_i. The same cloud provider cp_j can be used to execute multiple tasks, provided it has completed its earlier task and ready for executing the next task and the maximum satisfaction level. The ready time of the entry task is considered as 0. The Earliest Completion Time of a task is computed by considering the earliest start time of a task and the execution of the task on a CP. The EST of a task is computed by finding the maximum of the ready time of a CP for the task and the earliest end time of the entire ancestral task along with the data transfer time (Communication Cost). If any two tasks are allotted with the same cloud provider CP, then the communication cost of those two tasks CC(tk_m, tk_i) = 0.

The Earliest Completion Time (ECT) of a task is the earliest possible completion time of a workflow task on a cloud provider. This is computed by the Eq. (11.21)

$$\mathrm{ECT}\left(tk_i, cp_j\right) = \mathrm{EST}\left(tk_i, cp_j\right) + \frac{x\left(tk_i\right)}{x\left(cp_j\right)} \tag{11.21}$$

Makespan is the one representing the schedule length, and it is found by considering the ECT of exit task and it is denoted in Eq. (11.10). The objective of the schedule is to minimize this makespan.

$$\mathrm{makespan} = \mathrm{ECT}\left(tk_{\mathrm{exit}}, cp_j\right) \tag{11.22}$$

11.5 Fuzzy Logic-Based Workflow Scheduling

The fuzzy logic-based workflow scheduling (FLWS) model is given in the Algorithm.

Algorithm: The FLWS Algorithm

Algorithm: The FLWS Algorithm.

Input: Input: W = (TK, E) //W - Workflow DAG ,

Set of Cloud Providers- CP,

QoS Parameters – (time, cost, and trust) values as tuple

Output: A Schedule ***Sch*** with workflow tasks $tk_i \in TK$ mapped to Cloud Provider $cp_j \in CP$

1. $Sch = \emptyset$ **//** ***Sch*** is the schedule
2. Calculate $ECT(tk_i, cp_j)$ using equation (21).
3. Calculate $U_{tk_i}(tr_x), U_{tk_i}(t_x)$ *and* $U_{tk_i}(c_x)$ using equation (10), (11) and (12) respectively
4. $TaskList \leftarrow tk_{entry}$
5. **While** $TaskList \neq \emptyset$ **do**
6. **for all** available Cloud Provider cp_j providing service for task tk_i **do**
7. Find $ECT(tk_i, cp_j)$value using insertion-based scheduling policy
8. Compute $\lambda_k = \max\{\sum_{j=0}^{2} U_{Z_{j(x)}}^{T_i} * w_j\}$
9. **end for**
10. $Sch = Sch \cup Map(tk_i, S_k)$
11. Update *TaskList* with next level of tasks from W
12. **end while**
13. **Return** *Sch*

The algorithm fuzzy logic-based workflow scheduling (FLWS) gets the Workflow W as DAG, the set of Cloud Providers CP that are readily providing services for the tasks in the workflow, and the QoS Parameters values as integer for time, cost, and trust as tuple. Initially the schedule **Sch** is initialized as NULL. Calculate Earliest Completion Time ECT(tk_i, cp_j) for all the tasks in each Cloud Provider providing the service. The QoS Parameters values for (time, cost, and trust) are got as tuple for each service in Cloud Providers. With these values $U_{tk_i}(tr_x), U_{tk_i}(t_x)$ and $U_{tk_i}(c_x)$ can be calculated. Then for generating the schedule initially the first task is considered. Then with the loops at step 5 and 6, the decision-making process is repeated for all the tasks. The ECT value is computed with insertion-based scheduling policy. Step 8 does the calculation for making the fuzzy decision for finding the cloud service best suited for the task as per the QoS requirement of the user. The mapped service to the task Map(tk_i, S_k) is appended to the schedule Sch. These steps are repeated until all the tasks are done with. Finally, the schedule Sch is given as the output.

11.6 Results and Discussions

To illustrate the working of our fuzzy logic-based workflow scheduling (FLWS) model, we present a table with sample values for the six tasks in workflow shown in Fig. 11.2. Table 11.2 shows the sample QoS Parameters (time, cost, and trust) values as tuple for four Cloud Providers. The steps are worked out for the first task in the workflow.

Step 1

Using max-min operator, compute vectors of ($U_{T_i}(t_x)$, $U_{T_i}(c_x)$,$U_{T_i}(tr_x)$) for the first task. Here $t^1_{\max}$ = 4,$t^1_{\min}$=3; $c^1_{\max}$=5, $c^1_{\min}$=2; $tr^1_{\max}$=5, $tr^1_{\min}$=2 . For this task three cloud providers are providing their services. Hence the calculations are as shown below.

$$U^1_{tk_1} - \left(\frac{4-4}{4-3} \quad \frac{5-5}{5-2} \quad \frac{3-2}{5-2} \right) = \left(0 \quad 0 \quad 1/3\right)$$

$$U^2_{tk_1} - \left(\frac{4-3}{4-3} \quad \frac{5-2}{5-2} \quad \frac{5-2}{5-2} \right) = \left(1 \quad 1 \quad 1\right)$$

$$U^3_{tk_1} - \left(\frac{4-3}{4-3} \quad \frac{5-4}{5-2} \quad \frac{2-2}{5-2} \right) = \left(1 \quad 1/3 \quad 0\right)$$

Table 11.2 Sample QoS parameter values from multi-cloud providers for their services

Workflow task no.	(Time, cost, and trust) values by cloud providers			
	CP_1	CP_2	CP_3	CP_4
1	(4, 5, 3)	(3, 2, 5)	(3, 4, 2)	–
2	(3, 2, 1)	–	(3, 2, 5)	(2, 1, 1)
3	(4, 7, 3)	(5, 6, 4)	(6, 5, 5)	(7, 4, 4)
4	(2, 3, 6)	(3, 4, 1)	–	(6, 3, 4)
5	(5, 2, 1)	–	(4, 6, 5)	(6, 2, 7)
6	–	(4, 3, 2)	(3, 5, 4)	(5, 2, 2)

Step 2

The selection of cloud provider's service for task is computed by the maximum value of all three objectives based on weight factor. Maximize $U * W'$. As given in step 8 (Algorithm FLWS) do the calculations. Simply this can be denoted as $(U_{T_i}(t_x)\ U_{T_i}(c_x)\ U_{T_i}(tr_x))^*\ (w_t w_c w_{tr})'$. Here four cases are presented. In case 1 all the three objectives are given the same weightage. In case 2, time is given the priority and the other two factors are not considered. Similarly, in case 3, cost is given the priority and the other two factors are not considered. Also, in case 4, trust is given the priority and the other two factors are not considered.

Case 1

Consider $W = (\frac{1}{3}\ \frac{1}{3}\ \frac{1}{3})$ to the set $W = (w_t w_c w_{tr})$ for test.

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{3} \\ 1 & 1 & 1 \\ 1 & \frac{1}{3} & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{9} \\ 1 \\ \frac{4}{9} \end{pmatrix}$$

As per the results got, it could be understood that the second cloud provider's service $\mathbf{CP_2}$ is the best service for the first task considering equal weightage for all the three objectives time, cost, and trust.

Case 2

Let $W = (1, 0, 0,)$,

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{3} \\ 1 & 1 & 1 \\ 1 & \frac{1}{3} & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

As per the results, it is noted that both second and third cloud provider's service $\mathbf{CP_2}$ **and** $\mathbf{CP_3}$ are the best service for the first task considering time alone among the three objectives time, cost, and trust.

Case 3

Let $W = (0, 1, 0,)$,

$$\begin{pmatrix} U^1_{tk_1} \\ U^2_{tk_1} \\ U^3_{tk_1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \frac{1}{3} \\ 1 & 1 & 1 \\ 1 & \frac{1}{3} & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \frac{1}{3} \end{pmatrix}$$

When cost alone is considered among the three objectives, it is noted that cloud provider's service $\mathbf{CP_2}$ is the best service for the first task.

Case 4
Let $W = (0\ 0\ 1)$,

$$\begin{pmatrix} U_{tk_1}^{1} \\ U_{tk_1}^{2} \\ U_{tk_1}^{3} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1/3 \\ 1 & 1 & 1 \\ 1 & 1/3 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1 \\ 0 \end{pmatrix}$$

When trust alone is considered among the three objectives, it is noted that cloud provider's service $\mathbf{CP_2}$ is the best service for the first task.

The performance graph is also shown below. The algorithm is implemented in Cloudsim. In the simulation experiments, graphs are generated using DagGen library. DagGen is a synthetic task graph generator tool. This tool generates random and synthetic task graphs for the purpose of simulation. This is useful, for instance, to evaluate scheduling algorithms that must be tested over a wide range of application configurations. The comparison of FLWS, Minimum Critical Path (MCP), and greedy-cost models is presented by using the DAG generated. The experiments are done to sequentially test the three algorithms with different sizes. The number of tasks varies from 10 to 100 with an increment of 20. The comparisons of execution time and cost of FLWS, MCP, and greedy cost are shown in Figs. 11.3 and 11.4, respectively. As for the FLWS algorithm, time and cost are both considered simultaneously, which enables the user to compromise requirements to yield a genuinely better solution. MCP is good with execution time among the algorithms but with the highest cost. Conversely, greedy cost has the least cost but takes the longest execution time. As for the FLWS approach, time and cost are both considered simultaneously, which enables the user to get a better solution with their QoS.

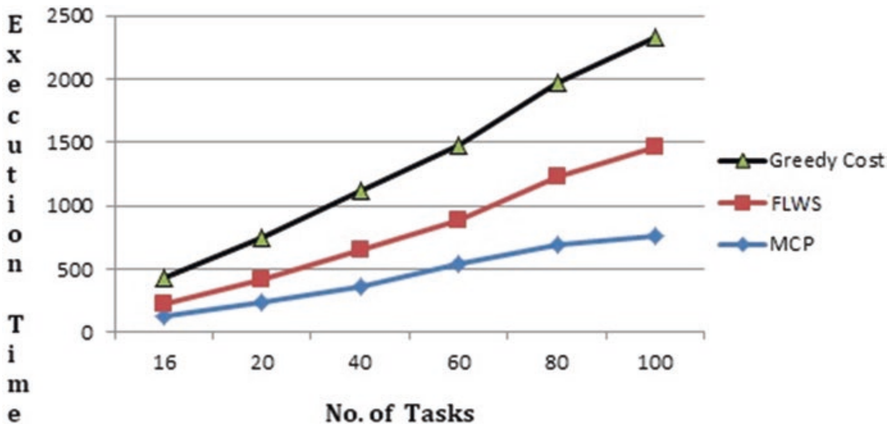


Fig. 11.3 Comparison of execution time

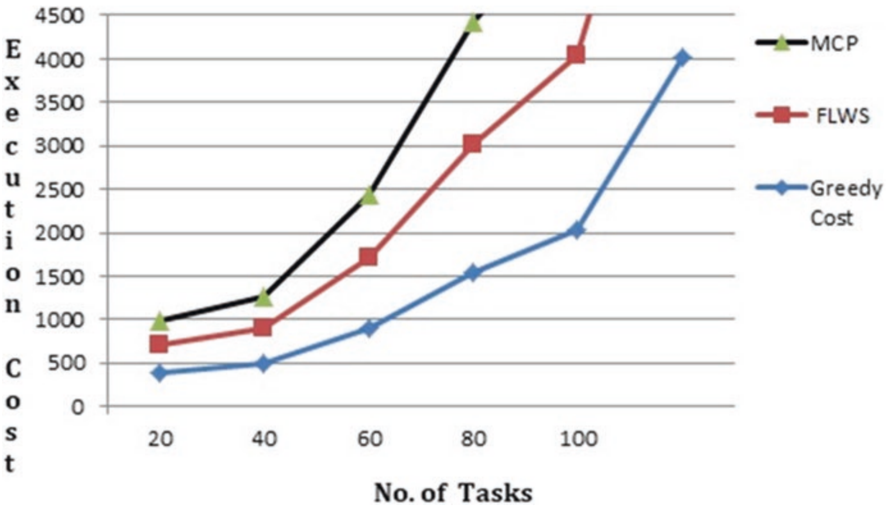


Fig. 11.4 Comparison of execution cost

11.7 Conclusion

This chapter discusses the problems related to workflow scheduling (WFS) in multi-cloud computing environment. A fuzzy logic-based decision for solving the multi-objective problem is proposed. The fuzzy logic-based workflow scheduling (FLWS) provides an optimal schedule for the given workflow with user QoS being satisfied in terms of time, cost, and trust. Thus, this FLWS algorithm provides cooperative model for workflow scheduling in multi-cloud environments along with the user quality of service being achieved.

References

1. Diaz-Montes, J., Diaz-Granados, M., Zou, M., Tao, S., & Parashar, M. (2018). Supporting data-intensive workflows in software-defined federated multi-clouds. *IEEE Transactions on Cloud Computing, 6*, 250–263. https://doi.org/10.1109/TCC.2015.2481410
2. Ranaldo, N., & Zimeo, E. (2009). Time and cost-driven scheduling of data parallel tasks in grid workflows. *IEEE System Journal, 3*, 104–120.
3. Buyya, R., Christian, V., & Tamarai Selvi, S. (2013). Mastering cloud computing. TMGH.
4. Crosby, S., et al. (2010). Open virtualization format specification. Distributed management task force. [online document] Retrieved from https://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf
5. Hwang, K., Fox, G. C., & Dongarra, J. G. (2012). *Distributed and cloud computing, from parallel processing to the internet of Things*. Morgan Kaufmann Publishers.
6. Tan, W. A., Sun, Y., Li, L. X., Lu, G. Z., & Wang, T. (2014). A trust service-oriented scheduling model for workflow applications in cloud computing. *IEEE Systems Journal, 8*, 868–878. https://doi.org/10.1109/JSYST.2013.2260072

7. Nagarajan, R., & Thirunavukarasu, R. (2020). A service context-aware QoS prediction and recommendation of cloud infrastructure services. *Arabian Journal for Science and Engineering, 45*, 2929–2943. https://doi.org/10.1007/s13369-019-04218-6
8. Farid, M., Latip, R., Hussin, M., & Abdul Hamid, N. A. W. (2020). Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment. *IEEE Access, 8*, 24309–24322. https://doi.org/10.1109/ACCESS.2020.2970475
9. Sujana, J. A. J., Revathi, T., & Rajanayagam, S. J. (2020). Fuzzy based security-driven optimistic scheduling of scientific workflows in cloud computing. *IETE Journal of Research, 66*(2), 224–241. https://doi.org/10.1080/03772063.2018.1486740
10. Nagarajan, R., & Thirunavukarasu, R. (2019). A fuzzy-based decision-making broker for effective identification and selection of cloud infrastructure services. *Soft Computing, 23*, 9669–9683.
11. Nagarajan, R., Thirunavukarasu, R., & Shanmugam, S. (2018). A fuzzy-based intelligent cloud broker with MapReduce framework to evaluate the trust level of cloud services using customer feedback. *International Journal of Fuzzy Systems, 20*, 339–347. https://doi.org/10.1007/s40815-017-0347-5
12. Nagarajan, R., Kumaran, S. S., & Thirunavukarasu, R. (2017). A fuzzy logic based trust evaluation model for the selection of cloud services. IEEE international conference on computer communication and informatics (ICCCI), Coimbatore. https://doi.org/10.1109/ICCCI.2017.8117686.
13. Sujana, J. A. J., Revathi, T., & Malarvizhili, M. (2015). Scheduling of scientific workflows in cloud with replication. *Applied Mathematical Sciences, 9*, 2273–2280.
14. Zadeh, L. (1965). Fuzzy sets. *Information and Control, 8*, 338–353.
15. Xu, L. (1988). A fuzzy multi-objective programming algorithm in decision support systems. *Annals of Operations Research, 12*, 315–320.

Chapter 12
Performance Evaluation and Comparison of Hypervisors in a Multi-Cloud Environment

Nalin Reddy, R. K. Nadesh, R. Srinivasa Perumal, Nikhil Chakravarthy Mallela, and K. Arivuselvan

12.1 Introduction

Many cloud service providers and the arising business need have created an enormous demand for computing power, platform, and applications across the globe. Digital Business Technology (DBT) is a combination of Operation Technology (OT) and Information Technology (IT). Everything as a Service (XAAS) can be easily provisioned with the help of a virtual machine. All the information technology infrastructure needs can be seamlessly provisioned as a virtual environment to the remote users with the support of virtualization. Cloud computing offers software, platform, database, infrastructure, security, and anything in IT as a service. For all kinds of offerings in public, private, and hybrid cloud, virtualization is the basic building block for the multi-cloud environment. A hypervisor is a tool used to create an abstraction of the resources to provide a multi-tenant computing model in a multi-cloud environment.

Virtualization is a broad term that refers to an abstraction of resources across many aspects of computing. One physical machine to support multiple virtual machines that run in parallel is the purpose. Many enterprises prefer to have a virtualized environment because of the drawbacks in the traditional computing environment like too many servers for too little work, aging of hardware and end of usable life, high infrastructure requirements, and the limited flexibility in the shared environments. The core technology behind virtualization is the hypervisor, sometimes referred to as the Virtual Machine Monitor. Hypervisor was introduced in the 1970s

N. Reddy · R. K. Nadesh (✉) · R. Srinivasa Perumal · N. C. Mallela · K. Arivuselvan
School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: rnalin.reddy2017@vitstudent.ac.in; r.srinivasaperumal@vit.ac.in; karivuselvan@vit.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_12

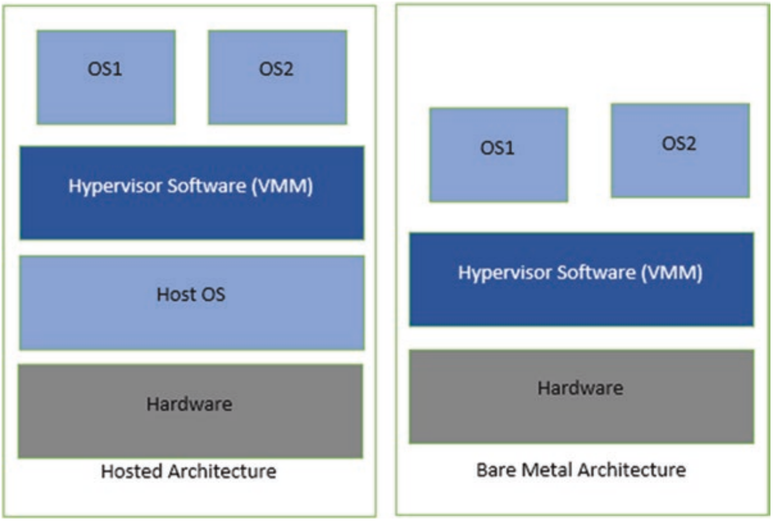


Fig. 12.1 Types of hypervisor

as part of the IBM S/360. A hypervisor is a computing layer that allows multiple operating systems to run a host computer simultaneously. The hypervisors are broadly classified as Type-1 and Type-2 hypervisor. The first type runs directly on the underlying host system, known as a bare-metal hypervisor or native hypervisor. The second of this kind has a host operating system running over the underlying host system and is also known as a hosted hypervisor (Fig. 12.1).

In the bare-metal hypervisor, the abstraction is done at the hardware layer and managed by the Virtual Machine Monitor (VMM). In the hosted system, the virtual machine is created, and the total control of the guest operating system is within the host operating system. This facilitates the option multi-tenant model with different operating systems and applications from multiple physical servers for the multi-cloud environment. Any user of private, public, hybrid cloud models can also share the resources among themselves in several regions where the computing model is called Inter-Cloud Computing. Server virtualization technology helps the enterprise provide platform, software, infrastructure, and security as a service in their private cloud owned by them or managed by the third-party service provider [1]. It is a promising technology to reduce the cost through server consolidation. Microsoft Hyper-V, VMware ESX are server-centric tools that tend to focus on the virtualization of the infrastructure. The other famous virtualization tools from VMware, Citrix, Oracle, and Amazon also play a significant role in hypervising the hardware. Each hypervisor has its features and limits, and choosing the appropriate hypervisors for digital business technology today is always challenging. Different virtualization types include server virtualization, storage virtualization, network virtualization, memory virtualization, and data virtualization in a multi-cloud computing environment. The multi-tenant deployed modes in data centers and application servers help the end user use the services from a fully isolated server, virtualized server, or shared virtualized server. The tenants can also share the database, schema, and server [2].

Virtualization can be done at any level, starting from the instruction set architecture level, hardware abstraction level, operating system level, user level API, and application level. Therefore, estimating CPU performance, memory, and disk helps the enterprise to choose the appropriate virtualization model by analyzing the number of iterations, number of threads, number of floating-point operations per second, and number of input/output operations per second. The memory performance is measured based upon the number of iteration, number of threads, throughput, size, and latency. The disk performance is calculated based on the read and write operations, size of the file, and time taken for read and write operations [3].

12.2 Related Works

Cloud computing continues to dominate by making the data and computing facilities available in an unprecedented economy, thereby increasing the end-user reliability and scalability. Nimbus and Open Nebula provide full freedom to build and manage enterprise cloud with advanced features for multi-tenancy, resource provision, and elasticity with virtualized services and applications. Open Nebula can integrate any hypervisor for any workload and server deployments. Xen and KVM are the two virtual managers taken for the performance measures, and results obtained prove that Xen outperforms in most cases than the KVM [4]. There are multiple challenges in configuring the hypervisor parameters to optimize its performance in the data center. The normal practice of tuning the scheduler parameter will not be effective in the highly dynamic workloads and utilization of the host machine resource. Discover, optimize, and observe were the three steps used to improve efficiency and termed as dynamic scheduler reconfigurations [5]. Cloud data centers have a collection of servers available to provide services to users at different times virtually. Continuous and uninterrupted services are essential, and therefore scheduling and maintenance activities in the data center are more difficult when the services are offered. A study was conducted to understand the server resources management, queuing the server requests with proper maintenance schedules for the server using an Open Nebula tool kit. Scheduling and maintenance algorithms are devised and deployed to estimate the system performance and evaluated through the Sunstone GUI server [6]. An experimental study was conducted with virtual machines, operating systems, and hardware and analyzed how configurations influence a deterministic algorithm [7]. The targeted system with multi-processor virtual machines placed on a commodity chip-multiprocessor and the performance degradation in a virtualized environment is analyzed considering the hypervisor slowdowns and the nearby VM neighborhood hypervisor noise. The aim is to reduce the noise level using different hardware and software techniques and ensure that they are ready for the cloud [8]. The use of virtual PC machine is considered in the context of interoperability with Network Controlled System, and specific application requests answer the need for virtualization, services, and security. Xen hypervisor is used to demonstrate the performance of the guest operating system and their control in the virtual environment [9].The online gaming system in the public cloud platforms was implemented as a virtualized cloud gaming platform supported with the latest software and hardware, and extended support is provided for both remote servers and local clients. This has achieved full-fledged deployment of gaming services, optimization, and integration of all the modules [10]. The workload-aware credit scheduling method is used for improving the network I/O performance in the virtual environment that gains better bandwidth and less response time, and fairness is achieved between I/O-intensive and CPU-intensive domains [11]. The distributed performance management schemes in virtual environments using non-linear controls have improved scalability among 10 virtual machines [12]. Variations in performance and scalability in IaaS clouds using multi-tier workloads in three hypervisors, namely, CVM, XEN, and KVM, show that each one has its own individual characteristics and performance according to their configurations [13]. Elastic applications for mobile cloud services with VM were implemented with a suspend and run the model. The front end screen reads the input and sends the same to back end server for processing. The processed result is displayed back on the user screen. This gives the user the feeling that the computation has happened in the mobile phone but was originally computed at the back end server [14]. Throughput based resource allocation model addresses the need for clustering and scheduling in a virtual environment. Virtual machine performance is computed, and based on the need, it is migrated to the next physical server available, thereby optimizing the server performance. This increases the overall efficiency of the virtual environment and has the key characteristics of cloud computing [15].The sharing of resources in a mobile environment with the help of remote method invocation is an added entity for the peer-to-peer computing model, which also supports the multi-cloud under heavy loads [16]. Feedback control for hypervisor-based multi-core is implemented with two controls. One control will limit the sharing of the resources, and the second control maintains the performance. These controls guarantee the execution of critical partitions, and it is recommended for the use of multi-core execution platforms [17]. Software-Defined Networking (SDN) plays a major role in separating the data plane from the control plane. Network Function Virtualization (NFV) allows multiple tenants the flexibility of sharing the physical resources. SDN and NFV are complementary technologies as virtualization allows the SDN networks to leverage the combined benefits of SDN networking and network virtualization. Therefore, centralized and distributed hypervisors are part of the SDN hypervisor [18]. Fault tolerance hypervisor is introduced to have a back-up hypervisor and the primary hypervisor running parallelly, and when the primary VM fails, the back-up hypervisor will start functioning within a time of 10 ms which is emerging approach for time-sensitive applications. Synchronization of both the virtual machines is done to provide economic and fault tolerant solution [19]. A virtual machine with a webserver running is used to study the performance degradation when Denial of Service (DoS) attacks happen in the network and it is observed that the performance degrades up to 23% when compared to non-virtual system. Generally, the performance degradation due to security threat is high when compared to the normal physical resources [20]. Project managers maintaining the infrastructure availability prefer to move virtualization to cloud computing to reduce capital expenses, operational expenses, and the total cost for ownership [21]. Common vulnerabilities, security, and threats during VM image sharing, VM migration, and secure host os were studied and recommended that the hypervisor should analyze the need for VM and ensure that unnecessary operations are not carried out [22]. Browser is the main entity of cloud computing environment where any user can directly connect themselves to the virtual environment. Minimizing the system resources while using the cloud-based application can be supported with browser add-on which is also secured and provides user friendliness [23].

12.3 Methodology

The virtual boxes VMware Workstation, Oracle VirtualBox, and Windows Subsystem for Linux (WSL) are used to study the virtual machine performance. VMware Workstation is a hosted hypervisor that supports the users to set up virtual machines on a single physical machine and run simultaneously. The hosted hypervisor runs on x64 version and x86–32 version of Microsoft Windows, Linux, BSD, and MS-DOS. The VMware Workstation has the free-of-charge version known as VMware Workstation player, and it can be used for non-commercial. However, the operating system needed to use is proprietary like Windows. VMware Workstation helps bridging the existing host network adaptors and sharing of host resources like physical disk drives, USB devices, and mounting of disk drives. The main benefit of using VMware Workstation is snapshot; the current state of the virtual machines can be saved and later resumed. One more added advantage of using VMware Workstation is grouping multiple images in an inventory folder; therefore the machines in the folder can be powered on and powered off as a single object and useful for client-server environments. The cross-platform virtualization software Oracle VM VirtualBox allows the system to run multiple operating systems at the same time. The latest release of Oracle VM VirtualBox 6.0 provides integration with the open source and the cloud development. Therefore, creating and deploying virtual machines can be done everywhere with upload and download option, review, and make changes offline. It also provides the developers to create multiplatform environments and also to develop applications for container within the Oracle VM VirtualBox on a single machine. The virtual machine can further be deployed to Oracle VM server in the case of server virtualization environments. Windows subsystem for Linux is compatible for running Linux binary executable natively on Windows 10 and Windows Server 2019. It uses the Linux kernel through a subset of Hyper-V features and allows the user to run Linux command-line tools alongside with the Windows command-line, desktop, and applications and access the Windows files from Linux. Each of the virtual machines is configured in three different players with certain workloads. For the execution of the task, each virtual machine is designed with one virtual CPU and 1GB of memory. The Windows Server framework was introduced inside the virtual machine. The 64-piece machine is favored since most of the hypervisors are perfect with the x64. HP Pavilion pc designed with an Intel i5-6200 U 2.40GHz processor with 8GB RAM is used for the study. Although the test framework has 8GB of RAM introduced, it is booted with just 2GB of RAM for local tests. Windows Subsystem for Linux is a compatibility layer for running Linux binary executables natively on Windows 10 and Windows Server 2019. The WSL has Full Linux kernel, increased file I/O performance, and full system call compatibility.

12.3.1 Setting Up of the Virtual Machines

1. Set up the initial system with Intel i7 processor and 8GB Ram.
2. Create a Virtual Machine over the host operating system.
3. Create three VM using VMware Workstation, Oracle VirtualBox, and WSL.
4. Kali Linux Operating System installed in all three Virtual Machines.
5. Key VM Configurations.

   Processors: 1VCPUs
   Memory: 2048 MB
   Network Adapter: Internal Virtual Switch
   Percentage of Total System Resources: 25%
   Clustered: No

6. Benchmark Tests, run on each Kali Virtual Machine to test the efficiency of the underlying Hypervisor.

12.3.2 Setting Up of the Benchmark Tests

Benchmark test is introduced to test the effectiveness and efficiency of the asset usage for the virtual environment.

The three benchmark tests are:

- CPU Benchmark
- Memory Benchmark
- Disk Performance Benchmark

12.3.2.1 CPU Test

A CPU benchmark (CPU benchmarking) is a progression of tests intended to quantify the presence of a PC or gadget CPU (or SoC). Many gauges or standard estimations are utilized to look at the exhibition of various frameworks, using similar strategies and conditions. A typical CPU benchmark test will test the framework against the gauges for the sort of CPU utilized. CPU particulars regularly estimated by a benchmark test incorporate the clock speed, the quantity of directions executed, library calls per cycle, and large engineering effectiveness factors. The benchmark norms change between ages of CPU and among Intel and AMD CPU. CPU benchmark programming will likewise assemble and give data on many principle gadgets in a PC framework, such as the processor, motherboard and chipset, and memory.

12.3.2.2 Memory Utilization Test

When a PC program needs to utilize an area of memory to store information, it makes a solicitation to windows for the measure of memory it requires. Windows designates the memory to the program (except if framework assets are deficient) and comes back to the mentioning program, the allotted square's primary memory opening. It is conceivable that a few projects may demand a lot of memory. The "Memory Speed Per Block Size" test, like the "Memory Speed Per Access Step Size" test, is made out of numerous means. During each test's progression, the performance test demands a square of memory and goes through the square estimating the entrance speed. memory benchmarking is a straightforward memory benchmark program, which attempts to quantify the pinnacle transfer speed of successive memory gets and the idleness of irregular memory brings. Data transmission is estimated by running diverse get-together codes for the adjusted memory squares and endeavoring distinctive prefetch techniques.

12.3.2.3 Disk Utilization Test

When PC clients get another hard drive plate, it is mandatory to know the hard drive's particular execution in their PCs. A plate execution test is done to understand the specific performance. Circle benchmarking is the way toward running programs that precisely measure move speeds under different plates in different situations. The point is to deliver figures in Mbps that abridge the speed attributes of a circle. There are a few free programming choices accessible, which you can, without much of a stretch, download and run yourself to benchmark your own drive.

12.4 Implementation

The proposed method is executed to study the CPU performance, memory performance, and disk performance. The virtual boxes VMware Workstation, Oracle VirtualBox, and WSL are used to review the virtual machines' performance. Each of the virtual machines is configured in three different players with certain workloads. All the three virtual machines created are executed to study the CPU performance w.r.t number of iterations, the number of threads, the number of Floating-Point operations per second, and the number of input/output operations second. The memory performance is measured based upon the number of iteration, number of threads, throughput, size, and latency. The disk performance is based on read and write operations, size of the file, and time taken for read and write operations. Kali Linux operating system was uniquely used in all the virtual machine, and the results obtained at the end of each iteration are tabulated.

12.4.1 Running Kali Linux on Windows Subsystem on Linux (Hypervisor—I)

The CPU benchmark testing is done using the code written in C/C++, and the obtained results are tabulated according to the iterations, number of threads, number of floating-point operations per second, and the number of I/O operations per second. A total number of 10 iterations are carried out to study the CPU performance for the hypervisor using Kali Linux on Windows Subsystem on Linux.

The memory performance is tabulated based on the memory benchmarking test parameters like size in bytes, number of threads, and throughput for the read and write operations for both random and sequential access for the number of iterations. The latency is also computed to understand the delay in read and write operations for the given task with the CPU performance for the hypervisor using Kali Linux on Windows Subsystem on Linux (Tables 12.1, 12.2, 12.3, 12.4, 12.5, 12.6, 12.7, 12.8, and 12.9).

The performance of the disk is carried out to estimate the reading and writing rate in Mbps; the size of the file is also considered for the total read and write operations per milliseconds. Disk benchmarking test is conducted along with the CPU and memory test for the hypervisor using Kali Linux on Windows Subsystem on Linux.

Table 12.1 CPU performance

Execution count	Number of iterations	Number of threads (multithreading)	Number of floating-point operations per second (in G-FLOPS)	Number of input/output operations per second (in G-FLOPS)
1.	1,000,000	1	13.761542	13.866962
2.	1,000,000	1	13.924173	13.830580
3.	1,000,000	1	13.414729	12.084710
4.	1,000,000	1	13.668033	13.620210
5.	1,000,000	1	13.266403	13.605371
6.	1,000,000	1	13.361323	13.551228
7.	1,000,000	1	13.550177	13.753762
8.	1,000,000	1	13.720102	13.442020
9.	1,000,000	1	13.578925	13.306751
10.	1,000,000	1	13.662736	13.220409
SUM	10,000,000	10	135.908143	134.282003

12.4.2 Running Kali Linux on VMware Workstation (Hypervisor-II)

The CPU benchmark testing is done using the code written in C/C++, and the obtained results are tabulated according to the iterations, number of threads, number of floating-point operations per second, and the number of I/O operations per second. A total number of 10 iterations are carried out to study the CPU performance for the hypervisor using Kali Linux on VMware Workstation.

The memory performance is tabulated based the memory benchmarking test parameters like Size in bytes, Number of Threads, Throughput for the read and write operations for both random and sequential access for the number of iterations. The latency is also computed to understand the delay in read and write operations for the given task with the CPU Performance for the hypervisor using Kali Linux on VMware Workstation.

The performance of the disk is carried out to estimate the reading and writing rate in Mbps, the size of the file is also considered for the total read and write operations per milliseconds. Disk benchmarking test is conducted along with the CPU and memory test for the hypervisor using Kali Linux on Windows Subsystem on VMware Workstation.

Table 12.2 Memory performance

Execution count	Number of iterations	Number of threads	Size (bytes)	Throughput for seqread()	Throughput for seqwrite()	Throughput for randread()	Throughput for randwrite()	Latency
1.	1,000,000	10	10,240	1093.576680	1163.855930	1249.048243	1028.942180	0.923900
2.	1,000,000	10	10,240	1114.380148	1585.388570	1235.726828	1145.171190	0.890000
3.	1,000,000	10	10,240	1140.562354	1309.574123	1115.328405	1301.770329	0.910900
4.	1,000,000	10	10,240	1110.838498	1203.392437	1358.699061	1236.455398	1.246600
5.	1,000,000	10	10,240	1062.710044	1183.694756	1241.359062	1333.302816	0.987500
6.	1,000,000	10	10,240	1079.948504	1375.542437	1124.116638	1191.022263	0.928900
7.	1,000,000	10	10,240	1212.436839	1230.578571	1204.567490	1242.535845	1.407700
8.	1,000,000	10	10,240	1113.078924	1276.804871	1321.040630	1344.414386	0.975300
9.	1,000,000	10	10,240	1165.537709	1209.499971	1501.075084	1286.280667	0.952300
10.	1,000,000	10	10,240	1122.852707	1290.952293	1242.830390	1289.682061	0.952800
SUM:	10,000,000	100	102,400	11215.922407	12829.283959	12593.791831	12399.577135	10.1759

Table 12.3 Disk performance

Execution count	Size of the file (bytes)	Total write time (ms)	Total read time (ms)	Writing rate (Mbps)	Reading rate (Mbps)
1.	10	77.6674	101.277	0.128759	0.0987387
2.	10	67.5392	101.461	0.148062	0.0985596
3.	10	78.4833	97.7968	0.127416	0.102253
4.	10	75.2111	104.913	0.132959	0.095317
5.	10	80.4484	98.7593	0.124303	0.101256
6.	10	88.9543	114.768	0.112417	0.0871322
7.	10	83.6196	104.838	0.119589	0.0953855
8.	10	89.4208	96.3829	0.111831	0.10386
9.	10	60.0919	101.707	0.166412	0.0983218
10.	10	64.8395	102.022	0.154227	0.0980185
SUM:	100	766.2755	1023.925	1.325975	0.9788423

Table 12.4 CPU performance

Execution count	Number of ıterations	Number of threads (multithreading)	Number of floating-point operations per second (in G-FLOPS)	Number of ınput/output operations per second (in G-FLOPS)
1.	1,000,000	1	13.094113	13.547008
2.	1,000,000	1	13.573411	13.398396
3.	1,000,000	1	13.545767	13.622528
4.	1,000,000	1	13.451017	13.493908
5.	1,000,000	1	13.544296	13.556375
6.	1,000,000	1	13.147545	13.008343
7.	1,000,000	1	13.516356	13.464139
8.	1,000,000	1	13.140898	13.388689
9.	1,000,000	1	13.547628	13.606009
10.	1,000,000	1	13.179020	13.325062
SUM:	10,000,000	10	133.740051	134.410457

12.4.3 Running Kali Linux on Oracle Virtual Box (Hypervisor-III)

The CPU benchmark testing is done using the code written in C/C++, and the obtained results are tabulated according to the iterations, number of threads, number of floating-point operations per second, and the number of I/O operations per second. A total number of 10 iterations are carried out to study the CPU performance for the hypervisor using Kali Linux on Oracle VirtualBox.

The memory performance is tabulated based on the memory benchmarking test parameters like size in bytes, number of threads, and throughput for the read and write operations for both random and sequential access for the number of iterations. The latency is also computed to understand the delay in read and write operations

Table 12.5 Memory performance

Execution count	Number of iterations	Number of threads	Size (bytes)	Throughput for eqread()	Throughput for seqwrite()	Throughput for randread()	Throughput for randwrite()	Latency
1.	1,000,000	10	10,240	936.186777	964.873246	1048.628431	1005.201553	1.0519
2.	1,000,000	10	10,240	1025.652663	1140.252284	1103.009520	1065.409470	1.1450
3.	1,000,000	10	10,240	990.484107	1082.289312	842.771259	1136.113549	1.1146
4.	1,000,000	10	10,240	694.835332	1180.363595	1096.951564	1018.133799	1.1420
5.	1,000,000	10	10,240	864.021094	1023.750061	1128.411084	1075.297134	1.2156
6.	1,000,000	10	10,240	873.340274	1145.984699	1017.738523	1152.091414	1.1927
7.	1,000,000	10	10,240	778.944397	1018.331553	863.558575	1103.531888	1.2161
8.	1,000,000	10	10,240	1262.280005	1073.151162	974.467729	1151.332418	1.2967
9.	1,000,000	10	10,240	805.914995	960.278401	1138.457196	1109.369446	1.3303
10.	1,000,000	10	10,240	835.552014	1078.504500	950.399710	1167.679287	1.2021
SUM:	10,000,000	100	102,400	9067.211658	10667.778813	10164.393591	10984.159958	11.907

Table 12.6 Disk performance

Execution count	Size of the file (bytes)	Total write time (ms)	Total read time (ms)	Writing rate (Mbps)	Reading rate (Mbps)
1.	10	18.9476	3.8338	0.527772	2.60838
2.	10	22.049	3.89774	0.453534	2.56559
3.	10	22.8735	3.84719	0.437188	2.5993
4.	10	21.3802	3.81358	0.467722	2.62221
5.	10	23.6215	3.67262	0.423343	2.72285
6.	10	24.0222	4.06257	0.416281	2.4615
7.	10	23.3403	4.85422	0.428443	2.06006
8.	10	19.2859	3.75868	0.518514	2.6605
9.	10	18.1091	3.79127	0.552209	2.63764
10.	10	18.2845	3.81848	0.54691	2.61884
SUM:	100	211.9138	39.35015	4.771916	25.55687

Table 12.7 CPU performance

Execution count	Number of iterations	Number of threads (multithreading)	Number of floating-point operations per second (in G-FLOPS)	Number of ınput/output operations per second (in G-FLOPS)
1.	100,000,000	1	13.290265	13.336000
2.	100,000,000	1	13.392698	13.228865
3.	100,000,000	1	13.415576	13.240553
4.	100,000,000	1	12.925370	12.646173
5.	100,000,000	1	12.357365	13.339931
6.	100,000,000	1	13.127266	13.099025
7.	100,000,000	1	13.050231	13.145842
8.	100,000,000	1	13.267999	13.165307
9.	100,000,000	1	13.429680	13.074873
10.	100,000,000	1	13.383794	13.627305
SUM:	1,000,000,000	10	131.640244	13.903874

for the given task with the CPU performance for the hypervisor using Kali Linux on Oracle VirtualBox.

The performance of the disk is carried out to estimate the reading and writing rate in Mbps; the size of the file is also considered for the total read and write operations per milliseconds. Disk benchmarking test is conducted along with the CPU and memory test for the hypervisor using Kali Linux on Windows Subsystem on Oracle VirtualBox.

Table 12.8 Memory performance

Execution count	Number of iterations	Number of threads	Size (bytes)	Throughput for seqread()	Throughput for seqwrite()	Throughput for randread()	Throughput for randwrite()	Latency
1.	1,000,000	10	10,240	914.229914	1263.572935	1186.507496	1365.777923	1.1557
2.	1,000,000	10	10,240	1386.180184	1330.511356	1235.799646	1357.204245	1.2348
3.	1,000,000	10	10,240	1385.447579	1323.040818	1374.010352	1589.233101	1.2516
4.	1,000,000	10	10,240	1159.544399	1187.649790	1195.571518	1296.378809	1.2280
5.	1,000,000	10	10,240	1064.490127	1191.902245	1077.341005	1129.687567	1.2341
6.	1,000,000	10	10,240	1378.526260	1377.801721	1401.745873	1360.903310	1.1872
7.	1,000,000	10	10,240	1376.264602	1340.890026	1312.689034	1549.886926	1.2520
8.	1,000,000	10	10,240	510.516809	1653.905363	1381.614072	1522.986202	1.1990
9.	1,000,000	10	10,240	1224.614307	1550.345235	1440.945445	1514.189170	1.4194
10.	1,000,000	10	10,240	1134.331458	1498.822184	1263.953713	1391.976636	1.3616
SUM:	10,000,000	100	102,400	11534.145639	13718.441673	12870.178154	14078.223889	12.5234

Table 12.9 Disk performance

Execution count	Size of the file	Total write time (ms)	Total read time (ms)	Writing rate (Mbps)	Reading rate (Mbps)
1.	10,240	19.4234	5.66896	0.514844	1.76399
2.	10,240	8.74747	5.43169	1.14319	1.84105
3.	10,240	23.3246	4.49415	0.428732	2.22511
4.	10,240	21.6748	5.04888	0.461366	1.98064
5.	10,240	18.7458	6.4755	0.533453	1.54428
6.	10,240	18.7675	8.55579	0.532836	1.1688
7.	10,240	14.3616	6.74879	0.696303	1.48175
8.	10,240	20.0254	4.70196	0.499366	2.12677
9.	10,240	11.0351	6.95651	0.906202	1.4375
10.	10,240	17.0936	7.97497	0.585015	1.25392
SUM:	102,400	173.19927	62.0572	6.301307	16.82381

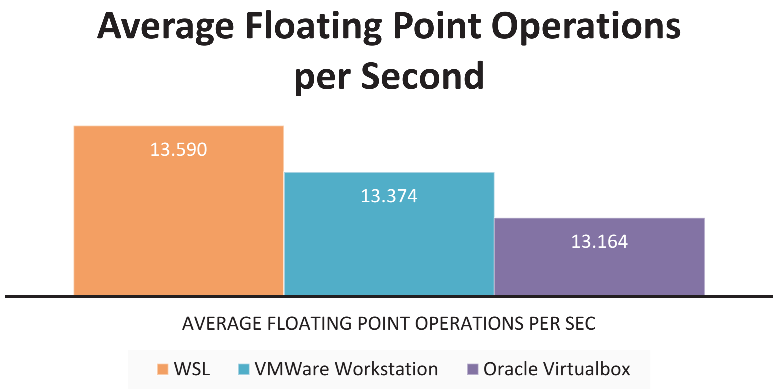


Fig. 12.2 Average floating-point operations per second – CPU performance

12.5 Results and Discussion

The results obtained after the execution of all the three virtual machines are summarized, and the average of each and every parameter is computed. The CPU performance of the three virtual machines is calculated based on the average floating-point operations per second. The averages of WSL, VMware Workstation, and Oracle VirtualBox are computed, and the average input/output operation per second (Fig. 12.2).

It is observed that the average flops obtained are less in Oracle VirtualBox when compared to the VMware Workstation and the WSL. However, the deviation is very less when compared. The I/O operations per second obtained and on computing the average of the input/output operations per second have yielded less in Oracle VirtualBox than the VMware Workstation and the WSL (Fig. 12.3).

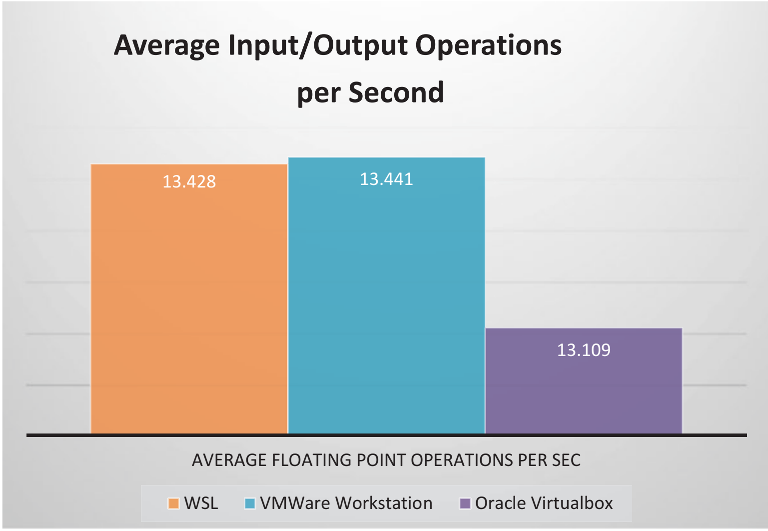


Fig. 12.3 Average I/O operations per second – CPU performance

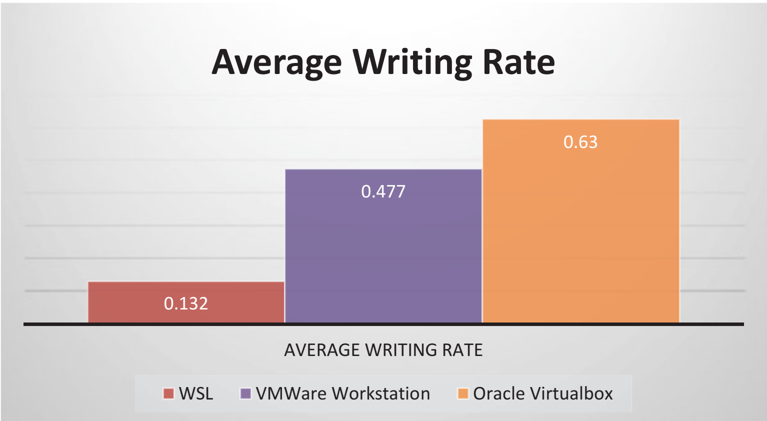


Fig. 12.4 Average writing rate – disk performance

The disk performance is measured by computing the average of the reading and writing rate. The WSL has claimed to have the lowest writing time when compared with the other two virtual boxes (Fig. 12.4).

The WSL reading time creditably has less reading time even though same operations were performed in the other two virtual boxes (Fig. 12.5).

The three virtual environments' overall memory performance and the throughput are calculated for sequential read/write operations and random read/write operations (Fig. 12.6).

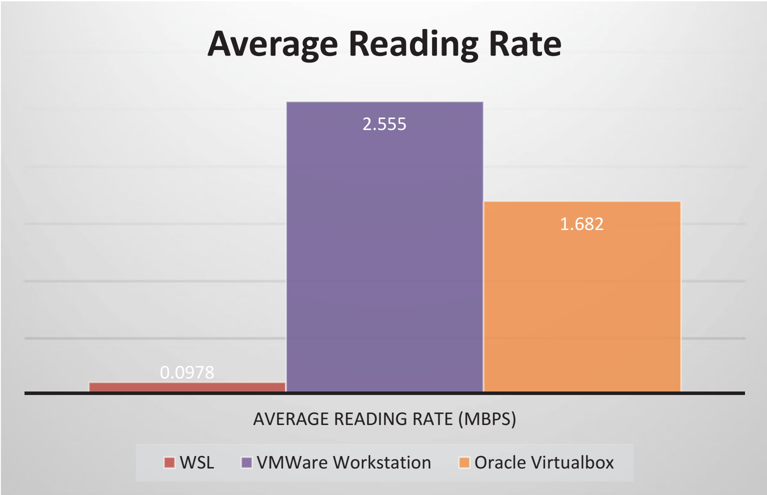


Fig. 12.5 Average reading rate – disk performance

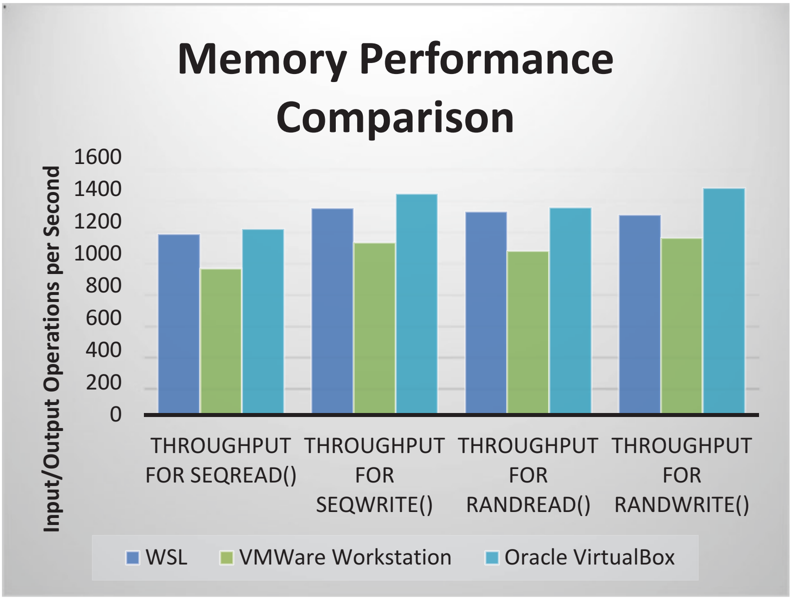


Fig. 12.6 Memory performance

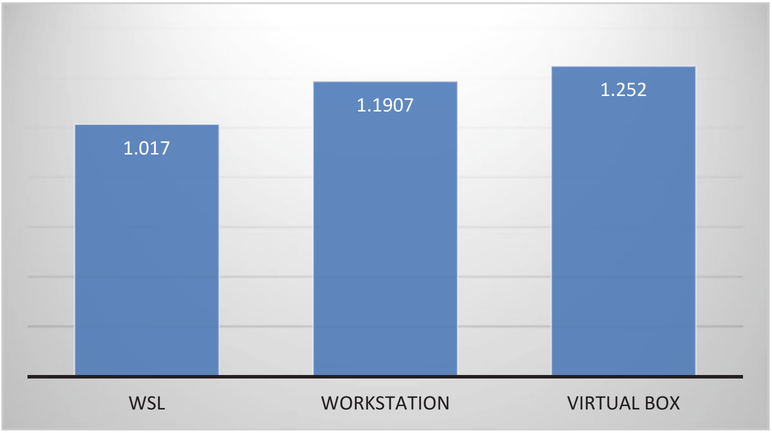


Fig. 12.7 Latency

The latency in the memory performance is measured for all the three virtual boxes, and WSL claims the minimum latency when compared to the other VM boxes (Fig. 12.7).

The average memory performance looks better in Oracle VirtualBox than in the other virtual environments, and the latency is also significantly less. Contrasting over three virtual machines, Oracle VirtualBox has the best memory performance compared to VMware Workstation and WSL. However, WSL tops floating-point operations but is placed second to VMware Workstation in input/output operations in CPU performance. In disk performance, the average writing rate is better in VMware Workstation, but the average reading rate is better in Oracle VirtualBox.

12.6 Conclusion

The performance of the CPU, memory, and disk utilization is considered as the main components in this study. The exhibition of frameworks utilizing Linux as Guest OS is impressively better contrasted with the Windows Host-Guest framework. The experimental results show that each hypervisor has got its own identity. Each one is better than the other in different aspects and can be used according to the enterprise demand and the need for the applications in the virtual environment.

References

1. Alaluna, M., Vial, E., Neves, N., & Ramos, F. M. V. (2019). Secure multi-cloud network virtualization. *Computer Networks, 161*, 45–60. https://doi.org/10.1016/j.comnet.2019.06.004
2. Fischer, A., Botero, J. F., Beck, M. T., et al. (2013). Virtual network embedding: A survey. *IEEE Communication Surveys and Tutorials, 15*, 1888–1906. https://doi.org/10.1109/SURV.2013.013013.00155

3. Nadesh, R. K., & Aramudhan, M. (2018). TRAM-based VM handover with dynamic scheduling for improved QoS of cloud environment. *International Journal of Internet Technology and Secured Transactions, 8*. https://doi.org/10.1504/IJITST.2018.093340
4. Armstrong, D., & Djemame, K. (2011). Performance issues in clouds: An evaluation of virtual image propagation and I/O paravirtualization. *The Computer Journal, 54*, 836–849. https://doi.org/10.1093/comjnl/bxr011
5. Caglar, F., Shekhar, S., & Gokhale, A. S. (2018). ITune: Engineering the performance of xen hypervisor via autonomous and dynamic scheduler reconfiguration. *IEEE Transactions on Services Computing, 11*, 103–116. https://doi.org/10.1109/TSC.2016.2538234
6. Nadesh, R. K., Jagadeesh, M., & Aramudhan, M. (2015). A quantitative study on the performance of cloud data Centre: Dynamic maintenance schedules with effective resource provisioning schema. *Indian Journal of Science and Technology, 8*. https://doi.org/10.17485/ijst/2015/v8i23/51867
7. Bôaventura, R. S., Yamanaka, K., & Oliveira, G. P. (2014). Performance analysis of algorithms for virtualized environments on cloud computing. *IEEE Latin America Transactions, 12*, 792–797. https://doi.org/10.1109/TLA.2014.6868884
8. Nikounia, S. H., & Mohammadi, S. (2018). Hypervisor and neighbors' noise: Performance degradation in virtualized environments. *IEEE Transactions on Services Computing, 11*, 757–767. https://doi.org/10.1109/TSC.2015.2464811
9. Gaj, P., Skrzewski, M., Stój, J., & Flak, J. (2015). Virtualization as a way to distribute PC-based functionalities. *IEEE Transactions on Industrial Informatics, 11*, 763–770. https://doi.org/10.1109/TII.2014.2360499
10. Shea, R., Fu, D., & Liu, J. (2015). Cloud gaming: Understanding the support from advanced virtualization and hardware. *IEEE Transactions on Circuits and Systems for Video Technology, 25*, 2026–2037. https://doi.org/10.1109/TCSVT.2015.2450172
11. Guan, H., Ma, R., & Li, J. (2014). Workload-aware credit scheduler for improving network I/O performance in virtualization environment. *IEEE Transactions on Cloud Computing, 2*, 130–142. https://doi.org/10.1109/tcc.2014.2314649
12. Patikirikorala, T., Wang, L., Colman, A., & Han, J. (2015). Differentiated performance management in virtualized environments using non-linear control. *IEEE Transactions on Network and Service Management, 12*, 101–113. https://doi.org/10.1109/TNSM.2015.2394472
13. Jayasinghe, D., Malkowski, S., Li, J., et al. (2014). Variations in performance and scalability: An experimental study in IaaS clouds using multi-tier workloads. *IEEE Transactions on Services Computing, 7*, 293–306. https://doi.org/10.1109/TSC.2013.46
14. Lakshmi Narayanan, K., & Nadesh, R. K. (2013). Elastic calculator: A mobile application for windows mobile using mobile cloud services. *International Journal of Engineering and Technology, 5*, 2759–2764.
15. Baucas, M. J., & Spachos, P. (2019). Using cloud and fog computing for large scale IoT-based urban sound classification. *Simulation Modelling Practice and Theory, 101*, 102013. https://doi.org/10.1016/j.simpat.2019.102013
16. Nadesh, R. K., & Aramudhan, M. (2018). An empirical study on peer-to-peer sharing of resources in mobile cloud environment. *International Journal of Electrical and Computer Engineering, 8*, 1933–1938. https://doi.org/10.11591/ijece.v8i3.pp1933-1938
17. Crespo, A., Balbastre, P., Simó, J., et al. (2018). Hypervisor-based multicore feedback control of mixed-criticality systems. *IEEE Access, 6*, 50627–50640. https://doi.org/10.1109/ACCESS.2018.2869094
18. Blenk, A., Basta, A., Reisslein, M., & Kellerer, W. (2016). Survey on network virtualization hypervisors for software defined networking. *IEEE Communication Surveys and Tutorials, 18*, 655–685. https://doi.org/10.1109/COMST.2015.2489183
19. Zhu, J., Jiang, Z., Xiao, Z., & Li, X. (2011). Optimizing the performance of virtual machine synchronization for fault tolerance. *IEEE Transactions on Computers, 60*, 1718–1729. https://doi.org/10.1109/TC.2010.224

20. Shea, R., & Liu, J. (2013). Performance of virtual machines under networked denial of service attacks: Experiments and analysis. *IEEE Systems Journal, 7*, 335–345. https://doi.org/10.1109/JSYST.2012.2221998
21. Rajasingh, J. S. J., & Reeves Wesley, J. (2020). Step into the cloud or stop with virtualization - the project manager's dialectic dilemma. *Procedia Computer Science, 172*, 1077–1083. https://doi.org/10.1016/j.procs.2020.05.157
22. Almutairy, N. M., & Al-Shqeerat, K. H. A. (2019). A survey on security challenges of virtualization technology in cloud computing. *International Journal of Computer Science and Information Technologies, 11*, 95–105. https://doi.org/10.5121/ijcsit.2019.11308
23. Jagadeesh, M., & Nadesh, R. K. (2013). Minimization of system resources for cloud based application using web browser add on. *International Journal of Engineering and Technology, 5*, 2777–2784.

Chapter 13
A Manifesto for Modern Fog and Edge Computing: Vision, New Paradigms, Opportunities, and Future Directions

Sukhpal Singh Gill

13.1 Introduction

In modern era, the utilization of Internet of Things (IoT) applications is growing exponentially and generating a large amount of data [1, 2]. Prominent tech giants (Facebook, Microsoft, Google, Apple, and Amazon) are using massive Cloud Datacenters (CDCs) to provide the effective user services to process this data in a proficient and trustworthy manner [3, 4]. Some critical and deadline-oriented IoT applications, such as traffic management, military, or healthcare, need to process their requests with minimum latency and response time [5, 6]. To solve this problem, the two advanced paradigms, fog and edge computing, are introduced to process and manage data quickly [7, 8]. Fog devices aid cloud to process the small user requests without sending it to cloud, while edge devices provide an innovative service to execute user request at edge device [9]. The research in both fog and edge computing is growing day by day since they emerged as groundbreaking computing paradigms [10]. Therefore, there is a need to assess the existing research related to emerging paradigms to find out the possible future directions and research opportunities existing in this field.

S. S. Gill (✉)
School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
e-mail: s.s.gill@qmul.ac.uk

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_13

13.1.1 Motivation

Research in the area of fog and edge computing is growing day by day. There are a few review and survey articles in the literature [1–7], but the research has purposefully grown in modern fog and edge computing. It is important to evaluate, advance, and upgrade the existing research of modern fog and edge computing. This work expands the existing research and presents a fresh manifesto on modern edge and fog computing to identify the further opportunities and future directions for emerging paradigms.

13.1.2 Contributions

In this work, we have summarized the existing research in terms of edge and fog computing and presented the relationship of fog and edge computing with new paradigms. A thorough investigation has been carried out to uncover the existing challenges and research questions. Research opportunities in the area of edge and fog computing are presented.

13.1.3 Chapter Structure

The reminder of the chapter is organized as follows: The background of edge and fog computing is presented in Sect. 13.2. Section 13.3 describes the applications of edge and fog computing. Architecture types of edge and fog computing are discussed in Sect. 13.4. Section 13.5 discusses new paradigms, and Sect. 13.6 presents the research opportunities and future directions. Lastly, Sect. 13.7 summarizes the chapter.

13.2 Background: Fog and Edge Computing

The concept of fog computing emerged in the year 2009 and was adopted in the year 2011 as an assistance to cloud computing [11] and manages the extensive data incoming from IoT devices continuously. Fog computing servers are deployed among edge devices and CDCs to process the small jobs quickly instead of sending to the CDC [12]. Further, edge computing emerged in 2010 and was adopted in the year 2012, which helps users to process their requests at their edge devices instead of sending to fog or cloud server [13]. Both fog and edge computing optimizes energy, execution cost, and time while maintaining the quality of service at required level. Figure 13.1 shows the basic model of fog and edge computing, which

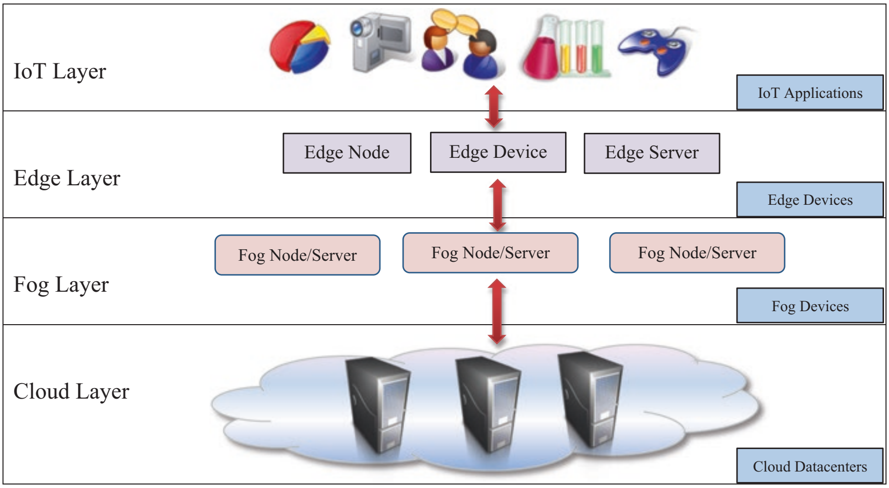


Fig. 13.1 Fog and edge computing model

contains four main layers: cloud, fog, edge, and IoT. Cloud layer consists of data centers and processes the user requests which needs large processing capability to manage big chunk of data. Fog layer contains the fog devices (servers or nodes) to process the small user requests to reduce latency and response time, and it needs less processing capability as compared to cloud computing. Edge layer contains edge devices to process user request at edge device despite of sending it to fog or cloud server. IoT layer comprises of IoT devices related to the various applications such as healthcare, agriculture, traffic monitoring, weather forecasting, or other sensors related to particular application.

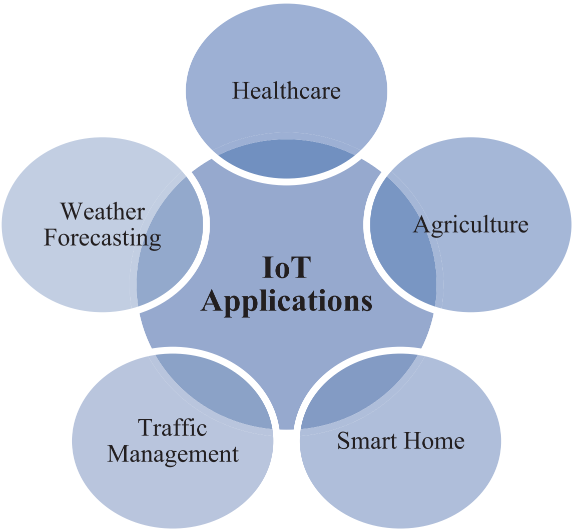
13.3 IoT Applications

IoT applications such as weather forecasting, water supply control, traffic monitoring, smart home and city, agriculture, and healthcare are utilizing fog and edge computing paradigms to process user data to reduce latency and response time. Figure 13.2 shows key IoT applications.

13.3.1 Healthcare

Healthcare is an important IoT application which is developed for different diseases such as cardiac arrest, diabetes, cancer, COVID-19 or pneumonia [14]. For example, an integrated fog and edge computing environment is used to diagnose the

Fig. 13.2 IoT applications



status of a heart patient automatically using various medical sensors [15]. Further, the current healthcare service can be improved by using next-generation technologies such as artificial intelligence or virtual reality to combat future pandemics [16].

13.3.2 Agriculture

Agriculture sector is using advanced technology to process various types of agriculture-related data for the prediction of different parameters such as production rate, water level, or crop quality [17]. For example, cloud-based agriculture system is developed to predict the agriculture status automatically based on the information provided by various IoT or edge devices [18]. Further, a mobile application is developed to manage the enormous volume of data and provide the required information to the farmers at their edge device to enable smart farming.

13.3.3 Smart Home

Electricity is an important source of power, which must be used wisely [19]. With the development of smart home, the owner can control their home appliances from mobile to optimize the use of energy and provide the required security by deploying cameras. For example, cloud and fog computing-based resource management technique is developed to control the edge devices using mobile application and control fans, lights, cameras, room temperature, and voltage using sensors related to various home appliances [19].

13.3.4 Traffic Management

IoT is playing an important role for managing traffic effectively using various sensors and actuators. For example, a smart transportation system is developed for the detection of potholes using IoT devices [20]. Further, various machine learning techniques were used to evaluate its performance in terms of performance parameters. Furthermore, fog and edge computing paradigms can be used to process data quickly for the earlier notification of potholes to avoid accidents.

13.3.5 Weather Forecasting

IoT and cloud computing can contribute toward weather and climate forecasting and science by facilitating observations empowered by IoT devices [21]. Traditionally measurements of weather parameters such as air pollution, humidity, and others are observed, saved, and displayed to the public and are also used by the researchers to improve the science behind the natural phenomena [22]. A sensor-based IoT system can be used for these observations which can send the data to the cloud.

13.4 Type of Architectures

Basically, there are three different types of architectures for fog and edge computing [23–25]: centralized, decentralized, and hierarchical as shown in Fig. 13.3. There is one chief controller in the *centralized* architecture, which controls all the internal activities such as task scheduling, provisioning of resources, scheduling of resources, and its execution. There are two different levels of schedulers in *hierarchical* architecture: upper and lower level. Upper level is a chief broker which manages all the lower-level schedulers, while lower level schedulers perform various internal

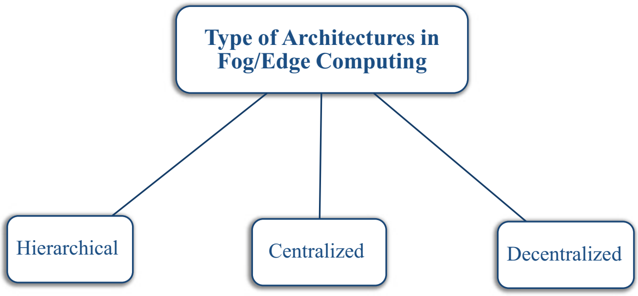


Fig. 13.3 Type of architectures in fog and edge computing

activities independently. *Decentralized* architecture allocates the resources to various user requests based on their specific necessities to complete their task, which also provides a high level of scalability.

13.5 Relevant Issues and Paradigms: A Vision

In this section, a future vision is presented while relevant issues and paradigms are discussed based on the existing research of fog and edge computing. Figure 13.4 shows the relevant issues and paradigms for fog and edge computing.

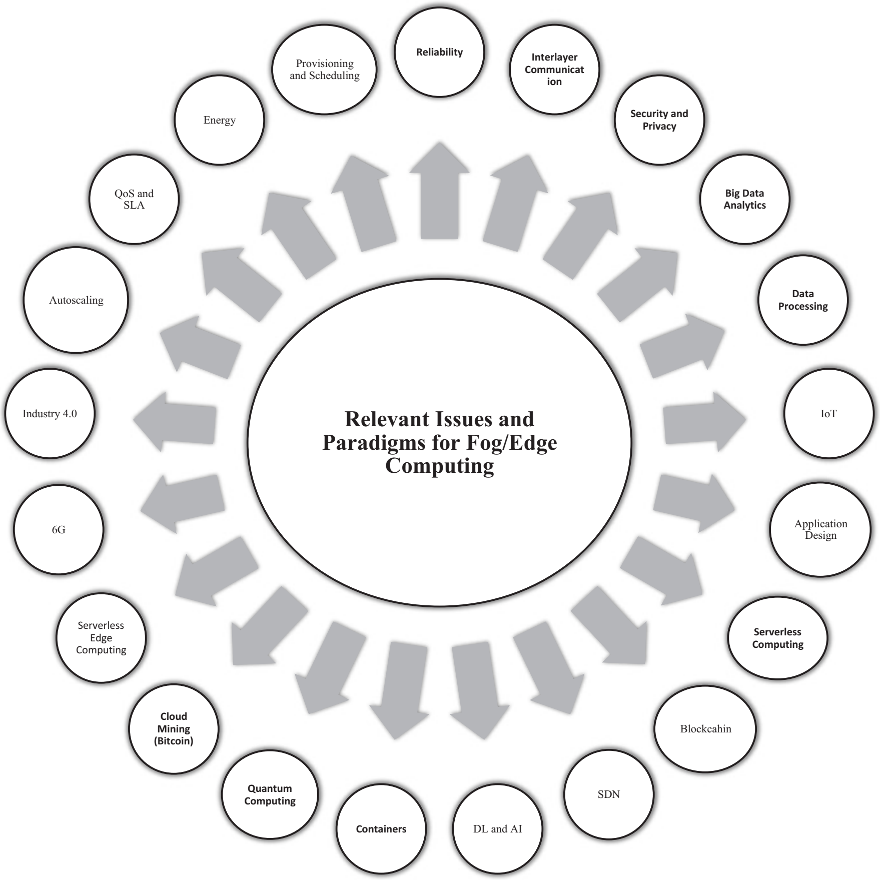


Fig. 13.4 Relevant issues and paradigms for fog and edge computing

13.5.1 Service Level Agreement (SLA) and Quality of Service (QoS)

QoS is a significant measurement parameter, which is used to measure the level of cloud, fog, and edge service [1]. QoS parameter can be reliability, security, availability, latency, elasticity, scalability, cost, and execution or response time [2]. SLA is an authorized contract, which is assured between cloud provider and end user based on different quality of service requirements, and it also contains some SLA deviation based on their negotiation [26, 27]. In case of SLA violation, a penalty can be issued to cloud provider, or compensation can be given to end user [28, 29]. Further, fulfilment of SLA and attainment of required QoS can improve the fog and edge service and increase the system performance.

13.5.2 Energy Efficiency and Sustainability

There is a need to run the cloud data centers continually to deliver the cloud service in a proficient way, which uses enormous volume of energy for computation and cooling [30, 31]. Further, the utilization of large amount of CDC and utilization of massive chunks of energy/power is making an impact on the environment, which leads to climate change by producing carbon footprints [32, 33]. To provide the sustainable and energy-efficient fog and edge service, there is a requirement to develop effective energy-efficient techniques for the management of fog/edge resources using machine learning, deep learning, or artificial intelligence (AI) [34].

13.5.3 Provisioning and Scheduling of Resources

Resource management for fog and edge computing comprises of three main tasks such as provisioning, scheduling, and execution of fog/edge resources [35, 36]. Resource provisioning is defined as the filtering of best matched resources based on the QoS requirements of the requests [37]. Further, resource scheduler manages the provisioned resources using different scheduling policies based on optimization algorithms, machine learning, or AI techniques to optimize the system performance [38]. Finally, resource manager executes the user workloads on scheduled resources.

13.5.4 Fault Tolerance (Reliability)

Fault tolerance is a very important concept to maintain the reliability and performance of the system [39]. Fog and edge computing paradigms provide an effective fault tolerant service to run the system continually without any interruption by fixing the faults dynamically [40]. Further, there is a need to find out the software, network, or hardware failure and their reasons and fixes to improve the reliability of service [41]. Furthermore, effective fault tolerant techniques can help the system to achieve the required objective under predefined conditions in a particular time period.

13.5.5 Interlayer Communication

Interlayer communication plays an important role for effective resource management to enable coordination amongst various tasks and subtasks going inside resource manager to perform resource provisioning and scheduling [42]. In fog-edge computing model, there is a requirement of an effective communication among various layers such as cloud, fog, edge, and IoT as shown in Fig. 13.1, which can improve the quality of service in terms response time and latency [43].

13.5.6 Security and Privacy

Security and privacy are two important challenges in computing system while delivering services to end users [44]. Privacy-aware fog and edge computing system allow authentic users to access their data using various IoT applications [45]. Further, security plays an important role to provide the safe and secure service to user without any interruption [46]. There is a need of proactive privacy and security-aware systems to protect user data and handle security attacks such as Denial of Service (DoS), Distributed-DoS(DDoS), Remote to Local (R2L), User to Root (U2R) and probing [47, 48].

13.5.7 Big Data Analytics

The growth in connected IoT devices is increasing continuously, and generating huge amounts of data in a fast manner, which is called big data [49]. So, big data analytics is an effective way of processing and managing large chunks of data to identify the retrieved information from knowledge base using innovative data management mechanisms [50]. Fog and edge computing paradigms can be used for proficient big data analytics to improve data processing [51].

13.5.8 Internet of Things (IoT)

IoT applications such as weather forecasting, water supply control, traffic monitoring, smart home and city, agriculture, and healthcare are utilizing fog and edge computing paradigms to process user data to reduce latency and response time [52, 53]. Fog and edge computing helps critical IoT applications to maintain the SLA during the processing of user requests in an efficient manner before their deadline [54, 55].

13.5.9 Data Processing

Fog and edge computing helps to gather, process, and handle the user data in near proximity of edge devices [56]. Due to the advancement in IoT devices, the generation is increasing quickly, which needs to be managed in a better way for effective utilization of storage. With the assistance of effective data processing mechanisms, the user data can be managed efficiently and retrieves the requisite information from the enormous volume of data using fog and edge computing [57].

13.5.10 Application Design

The design of an IoT application is very important for providing effective services using fog and edge computing [58]. Application can be designed using two different ways: monolithic or microservices [59]. Monolithic application is developed as a distinct element, while microservices are presented officially with business-oriented APIs and are very effective in offering scalability due to less coupling.

13.5.11 Serverless Computing

This concept enables the easy and faster development of IoT applications by eradicating the necessity to maintain real infrastructure [60]. Further, Serverless computing or Function as a Service (FaaS) runs the code as independent functions by automatic provisioning of resources and helps to manage the required resources dynamically [61]. Furthermore, execution of independent functions improves the scalability of infrastructure at runtime [62].

13.5.12 Blockchain

This is a new concept of storing data in the form of chain of blocks to increase the security of data [63]. Mostly, ledger are using for transactions to secure data in Blockchain [64]. Further, Blockchain technology can be used to protect the data coming from IoT applications [65].

13.5.13 Software-Defined Network (SDN)

The concept of virtualization can be used for fog and edge computing to provide security and offers virtualization in an efficient manner [66]. Software-Defined Network (SDN)-based fog and edge computing can decline energy consumption and increases the utilization of network while processing the data coming from IoT applications [67].

13.5.14 Deep Learning (DL) and Artificial Intelligence (AI)

Artificial Intelligence (AI) is an important branch of computer science, which is working like humans and helps to run various systems automatically [1]. Latest AI or deep learning techniques are effective in managing the resources for fog and edge computing while maintaining the SLA during execution [52]. Further, effective use of AI techniques can enhance the energy efficiency and reliability of the CDC [68, 69].

13.5.15 Containers

Existing systems are using Virtual Machines (VMs) for virtualization in fog and edge computing, which can be replaced with latest technology, i.e., containers [70]. The utilization of containers can enable independence among execution of various applications which are running on the same operating system. Further, the effective use of Docker-based containers can reduce total cost of ownership.

13.5.16 Quantum Computing

The concept of quantum computing can be used to provide aid to various machine learning or deep learning techniques for the predication of resource requirement in fog and edge computing in advance [71]. Further, successful deployment of quantum-based machine learning can help to manage fog and edge resources automatically and improve the resource utilization and energy efficiency.

13.5.17 Cloud Mining (Bitcoin)

The concept of cloud mining is introduced recently, which uses shared processing power to access remotely located cloud data center and produce the new bitcoins using computing power without managing the infrastructure [17]. Further, this concept can be extended toward fog and edge computing paradigms to optimize the utilization of computational or processing power.

13.5.18 Serverless Edge Computing

Serverless edge computing is a new research area to explore where both Serverless and edge computing are utilized together to achieve a single objective [72]. Further, it provides the function as a service and executes individual functions on edge devices, which further saves energy consumption, reduces the latency and response time, and improves the reliability while processing at closest edge device.

13.5.19 6G Technology

6G is the sixth generation and a successor of 5G technology to provision cellular data networks by increasing communication speed to a large extent [73]. The communication amongst IoT, fog, and edge devices can be improved with the help of 6G technology [74]. During the execution of user requests, 6G can offer fast transmission while exchanging data among edge devices to reduce latency and response time.

13.5.20 Industry 4.0

The use of next-generation technology in the area of manufacturing and industrial practices is increasing to automate the entire process for effective management of available resources within industry [44]. Smart industry can use AI or machine leaning-based techniques to incorporate the automatic learning of the system, which can further improve the internal processes such as asset management or resource allocation [75].

13.5.21 Autoscaling

The utilization of Serverless computing or FaaS for fog and edge computing can satisfy the fluctuating demand of various IoT applications dynamically [1]. Further, the concept of autoscaling can improve the self-confirmation of resources, self-optimization of QoS parameters, self-protection from attacks, and self-healing from occurrence of software, hardware, or network faults without manual intervention [76].

13.6 Opportunities and Future Directions

Table 13.1 shows the various possible research opportunities and promising future directions for practitioners, researchers, and academicians.

13.7 Summary and Conclusions

The use of IoT applications is increasing day by day and producing a lot of data in seconds. Cloud platform is effective in data management dynamically, but latest IoT applications need to process data with minimum latency and response time. In this chapter, a manifesto for modern fog and edge computing systems is presented to evaluate the ongoing research in this field. Further, type of architectures and applications for fog and edge computing are presented. Finally, research opportunities and promising future directions are highlighted.

Table 13.1 Opportunities and future directions

Relevant issues and paradigms	Opportunities and future directions
QoS and SLA	How to maintain the SLA and QoS at runtime during the execution of resources and workloads in fog and edge computing?
Energy efficiency and sustainability	How to deliver sustainable and energy-efficient service using AI or machine leaning methods for fog/edge computing?
Resource provisioning and scheduling	How to provision fog and edge resources effectively for different IoT applications before actual scheduling of resources?
Fault tolerance (reliability)	How to maintain the reliability of service while delivering the sustainable service simultaneously?
Interlayer communication	What is the impact of better interlayer communication among various layers of fog-edge computing model on QoS parameters?
Security and privacy	How to maintain the privacy and security of fog/edge computing systems during the data processing of IoT applications?
Big data analytics	How to use AI or machine learning techniques for big data analytics for effective management and analysis of data?
Internet of Things	How to maintain the SLA during the processing of user requests in an efficient manner before their deadline?
Data processing	What is the impact of data processing mechanisms on latency and response time?
Application design	How to design new energy-efficient IoT applications for better utilization of fog/edge resources?
Serverless computing	How to improve the scalability by using Serverless computing for fog/edge computing?
Blockchain	How Blockchain can be used to protect the data for IoT applications?
Software-defined network	How SDN can be used for fog/edge computing to reduce power consumption?
DL and AI	How to improve the deep learning and AI-based resource management techniques?
Containers	How the utilization of containers for virtualizations can improve the QoS in fog/edge computing?
Quantum computing	How to improve machine learning techniques using quantum computing?
Cloud mining (bitcoin)	How fog and edge computing can be used to optimize the utilization of computing or processing power?
Serverless edge computing	How FaaS can be used to improve the QoS in terms of power and utilization of resources?
6G technology	How 6G can be used to offer fast transmission while exchanging data among edge devices to reduce latency and response time?
Industry 4.0	How to perform predicative analysis of industry assets using AI, fog, and edge computing?
Autoscaling	How to provide effective autoscaling of resources to maintain the SLA and QoS at runtime for fog/edge computing?

References

1. Gill, S. S., Tuli, S., Xu, M., Singh, I., Singh, K. V., Lindsay, D., Tuli, S., Smirnova, D., Singh, M., Jain, U., & Pervaiz, H. (2019). Transformative effects of IoT, Blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges. *Internet of Things, 8*, 100118.
2. 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, 11*, 100273.
3. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13–16).
4. Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: Concepts, applications and issues. In Proceedings of the 2015 workshop on mobile big data (pp. 37–42).
5. Yi, S., Hao, Z., Qin, Z., & Li, Q. (2015). Fog computing: Platform and applications. In the Third IEEE workshop on hot topics in web systems and technologies (HotWeb) (pp. 73–78). IEEE.
6. Vaquero, L. M., & Rodero-Merino, L. (2014, October 10). Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review, 44*(5), 27–32.
7. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal, 3*(5), 637–646.
8. Satyanarayanan, M. (2017). The emergence of edge computing. *Computer, 50*(1), 30–39.
9. Shi, W., & Dustdar, S. (2016). The promise of edge computing. *Computer, 49*(5), 78–81.
10. Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2017). Mobile edge computing: A survey. *IEEE Internet of Things Journal, 5*(1), 450–465.
11. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials, 19*(4), 2322–2358.
12. Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. *ETSI White Paper, 11*(11), 1–6.
13. Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems, 97*, 219–235.
14. Tuli, S., Basumatary, N., Gill, S. S., Kahani, M., Arya, R. C., Wander, G. S., & Buyya, R. (2020). Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments. *Future Generation Computer Systems, 104*, 187–200.
15. Gill, S. S., Arya, R. C., Wander, G. S., & Buyya, R. (2018). Fog-based smart healthcare as a big data and cloud service for heart patients using IoT. In *International Conference on Intelligent Data Communication Technologies and Internet of Things* (pp. 1376–1383). Springer.
16. Tuli, S., Tuli, S., Wander, G., Wander, P., Gill, S. S., Dustdar, S., Sakellariou, R., & Rana, O. (2020). Next generation technologies for smart healthcare: Challenges, vision, model, trends and future directions. *Internet Technology Letters, 3*(2), e145.
17. Singh, S., Chana, I., & Buyya, R. (2020). Agri-info: Cloud based autonomic system for delivering agriculture as a service. *Internet of Things, 9*, 100131.
18. Gill, S. S., Chana, I., & Buyya, R. (2017). IoT based agriculture as a cloud and big data service: The beginning of digital India. *Journal of Organizational and End User Computing (JOEUC), 29*(4), 1–23.
19. 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.
20. Bansal, K., Mittal, K., Ahuja, G., Singh, A., & Gill, S. S. (2020). DeepBus: Machine learning based real time pothole detection system for smart transportation using IoT. *Internet Technology Letters, 3*(3), e156.

21. Olivares-Rojas, J. C., Reyes-Archundia, E., Gutiérrez-Gnecchi, J. A., Molina-Moreno, I., Méndez-Patiño, A., & Cerda-Jacobo, J. (2020). Forecasting electricity consumption using weather data in an edge-fog-cloud data analytics architecture. In *International conference on P2P, parallel, grid, cloud and internet computing* (pp. 410–419). Springer.
22. Krishnan, R., Singh, M., Vellore, R., & Mujumdar, M. (2020). Progress and prospects in weather and climate modelling. arXiv preprint arXiv:2011.11353. 2020 November 23.
23. Hong, C. H., & Varghese, B. (2019). Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms. *ACM Computing Surveys (CSUR), 52*(5), 1–37.
24. Varshney, P., & Simmhan, Y. (2017). Demystifying fog computing: Characterizing architectures, applications and abstractions. In IEEE 1st international conference on fog and edge computing (ICFEC) (pp. 115–124). IEEE.
25. Omoniwa, B., Hussain, R., Javed, M. A., Bouk, S. H., & Malik, S. A. (2018). Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues. *IEEE Internet of Things Journal, 6*(3), 4118–4149.
26. Gill, S. S., Chana, I., Singh, M., & Buyya, R. (2018). Chopper: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Cluster Computing, 21*(2), 1203–1241.
27. Singh, S., Chana, I., & Singh, M. (2017). The journey of QoS-aware autonomic cloud computing. *IT Professional, 19*(2), 42–49.
28. Singh, S., & Chana, I. (2016). QoS-aware autonomic resource management in cloud computing: A systematic review. *ACM Computing Surveys (CSUR), 48*(3), 1–46.
29. Singh, S., & Chana, I. (2015). Q-aware: Quality of service based cloud resource provisioning. *Computers & Electrical Engineering, 47*, 138–160.
30. Zhou, Q., Xu, M., Gill, S. S., Gao, C., Tian, W., Xu, C., & Buyya, R. (2020). Energy efficient algorithms based on VM consolidation for cloud computing: comparisons and evaluations. Proceedings of the 20th IEEE/ACM international symposium on cluster, cloud, and internet computing (CCGrid 2020, IEEE CS Press, USA), Melbourne, Australia, May 11–14, 2020.
31. Malik SU, Akram H, Gill SS, Pervaiz H, Malik H. (2020) Effort: Energy efficient framework for offload communication in mobile cloud computing. *Software: Practice and Experience*.
32. Gill, S. S., Garraghan, P., Stankovski, V., Casale, G., Thulasiram, R. K., Ghosh, S. K., Ramamohanarao, K., & Buyya, R. (2019). Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge. *Journal of Systems and Software, 155*, 104–129.
33. Gill, S. S., & Buyya, R. (2018). A taxonomy and future directions for sustainable cloud computing: 360 degree view. *ACM Computing Surveys (CSUR), 51*(5), 1–33.
34. Puthal, D., Obaidat, M. S., Nanda, P., Prasad, M., Mohanty, S. P., & Zomaya, A. Y. (2018). Secure and sustainable load balancing of edge data centers in fog computing. *IEEE Communications Magazine, 56*(5), 60–65.
35. Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing, 14*(2), 217–264.
36. 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.
37. Gill, S. S., Chana, I., Singh, M., & Buyya, R. (2019). RADAR: Self-configuring and self-healing in resource management for enhancing quality of cloud services. *Concurrency and Computation: Practice and Experience, 31*(1), –e4834.
38. 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.
39. Gill, S. S., & Buyya, R. (2018). Failure management for reliable cloud computing: A taxonomy, model and future directions. *Computing in Science & Engineering*.
40. Sharif, A., Nickray, M., & Shahidinejad, A. (2020). Fault-tolerant with load balancing scheduling in a fog-based IoT application. *IET Communications, 14*(16), 2646–2657.

41. Grover, J, & Garimella, R. M. (2018). Reliable and fault-tolerant IoT-edge architecture. In IEEE sensors (pp. 1–4). IEEE.
42. Díaz-de-Arcaya, J., Miñon, R., & Torre-Bastida, A. I. (2019). Towards an architecture for big data analytics leveraging edge/fog paradigms. In Proceedings of the 13th European conference on software architecture (Vol. 2, pp. 173–176).
43. Krishnan, P., Duttagupta, S., & Achuthan, K. (2020). SDN/NFV security framework for fog-to-things computing infrastructure. *Software: Practice and Experience, 50*(5), 757–800.
44. Golec, M., Gill, S. S., Bahsoon, R., & Rana, O. (2020). BioSec: A biometric authentication framework for secure and private communication among edge devices in IoT and industry 4.0. *IEEE Consumer Electronics Magazine*.
45. Gill, S. S., & Shaghaghi, A. (2020). Security-aware autonomic allocation of cloud resources: A model, research trends, and future directions. *Journal of Organizational and End User Computing (JOEUC), 32*(3), 15–22.
46. Gill, S. S., & Buyya, R. (2018). Secure: Self-protection approach in cloud resource management. *IEEE Cloud Computing, 5*(1), 60–72.
47. Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal, 4*(5), 1125–1142.
48. Yi, S., Qin, Z., & Li, Q. (2015). Security and privacy issues of fog computing: A survey. In *International conference on wireless algorithms, systems, and applications* (pp. 685–695). Springer.
49. Gill, S. S., & Buyya, R. (2019). Bio-inspired algorithms for big data analytics: A survey, taxonomy, and open challenges. In Big data analytics for intelligent healthcare management (pp. 1–17). Academic.
50. Badidi, E., Mahrez, Z., & Sabir, E. (2020). Fog computing for smart cities' big data management and analytics: A review. *Future Internet, 12*(11), 190.
51. Hussain, M. M., Beg, M. S., & Alam, M. S. (2020). Fog computing for big data analytics in IoT aided smart grid networks. *Wireless Personal Communications, 114*(4), 3395–3418.
52. Tuli, S., Gill, S. S., Casale, G., & Jennings, N. R. (2020). iThermoFog: IoT-fog based automatic thermal profile creation for cloud data centers using artificial intelligence techniques. *Internet Technology Letters, 3*(5), e198.
53. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13–16).
54. Rahmani, A. M., Liljeberg, P., Preden, J. S., & Jantsch, A. (Eds.). (2017). *Fog computing in the internet of things: Intelligence at the edge*. Springer.
55. Morabito, R. (2017). Virtualization on internet of things edge devices with container technologies: A performance evaluation. *IEEE Access, 5*, 8835–8850.
56. Soo, S., Chang, C., Loke, S. W., & Srirama, S. N. (2018). Proactive mobile fog computing using work stealing: Data processing at the edge. In Fog computing: breakthroughs in research and practice (pp. 264–283). IGI global.
57. Bierzynski, K., Escobar, A., & Eberl, M. (2017). Cloud, fog and edge: Cooperation for the future?. In Second international conference on fog and mobile edge computing (FMEC) (pp. 62–67). IEEE.
58. Gill, S. S., & Buyya, R. (2019). Sustainable cloud computing realization for different applications: A manifesto. In *Digital business* (pp. 95–117). Springer.
59. Pore, M., Chakati, V., Banerjee, A., & Gupta, S. K. (2019). Middleware for fog and edge computing: Design issues. In Fog and edge computing: principles and paradigms. Wiley.
60. Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., Muthusamy, V., Rabbah, R., Slominski, A., & Suter, P. (2017). Serverless computing: Current trends and open problems. In *Research advances in cloud computing* (pp. 1–20). Springer.

61. McGrath, G., & Brenner, P. R. (2017). Serverless computing: Design, implementation, and performance. In IEEE 37th international conference on distributed computing systems workshops (ICDCSW) (pp. 405–410). IEEE.
62. Fox, G. C., Ishakian, V., Muthusamy, V., & Slominski, A. (2017). Status of serverless computing and function-as-a-service (faas) in industry and research. arXiv preprint arXiv:1708.08028.
63. Bouraga, S. (2020). A taxonomy of blockchain consensus protocols: A survey and classification framework. *Expert Systems with Applications*. 114384.
64. Deepa, N., Pham, Q. V., Nguyen, D. C., Bhattacharya, S., Gadekallu, T. R., Maddikunta, P. K., Fang, F, & Pathirana, P. N. (2020). A survey on Blockchain for big data: Approaches, opportunities, and future directions. arXiv preprint arXiv:2009.00858.
65. Ankenbrand, T., Bieri, D., Cortivo, R., Hoehener, J., & Hardjono, T. (2020). Proposal for a comprehensive (crypto) asset taxonomy. In2020 Crypto Valley conference on Blockchain technology (CVCBT) (pp. 16–26). IEEE.
66. Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE, 103*(1), 14–76.
67. Duan, Y., Li, W., Fu, X., Luo, Y., & Yang, L. (2017). A methodology for reliability of WSN based on software defined network in adaptive industrial environment. *IEEE/CAA Journal of Automatica Sinica, 5*(1), 74–82.
68. Dhillon, A., Singh, A., Vohra, H., Ellis, C., Varghese, B., & Gill, S. S. (2020). IoTPulse: Machine learning-based enterprise health information system to predict alcohol addiction in Punjab (India) using IoT and fog computing. *Enterprise Information Systems*. 1–33.
69. Li, L., Ota, K., & Dong, M. (2018). Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics, 14*(10), 4665–4673.
70. Bachiega, N. G., Souza, P. S., Bruschi, S. M., & De Souza, S. D. (2018) Container-based performance evaluation: A survey and challenges. In IEEE international conference on cloud engineering (IC2E) (pp. 398–403). IEEE.
71. Gill, S. S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M., & Buyya, R. (2020). Quantum computing: A taxonomy, systematic review and future directions. arXiv preprint arXiv:2010.15559.
72. Aslanpour, M. S., Toosi, A. N., Cicconetti, C., Javadi, B., Sbarski, P., Taibi, D., Assuncao, M., Gill, S. S., Gaire, R., & Dustdar, S. (2021). Serverless edge computing: Vision and challenges. Proceedings of the 19th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2021), Dunedin, New Zealand.
73. Nawaz, F., Ibrahim, J., Awais, M., Junaid, M., Kousar, S., & Parveen, T. (2020). A review of vision and challenges of 6G technology. *International Journal of Advanced Computer Science and Applications,* 11(2).
74. Stergiou, C. L., Psannis, K. E., & Gupta, B. B. (2020). IoT-based big data secure management in the fog over a 6G wireless network. *IEEE Internet of Things Journal*.
75. Sengupta, J., Ruj, S., & Bit, S. D. (2020). A secure fog based architecture for industrial internet of things and industry 4.0. *IEEE Transactions on Industrial Informatics*.
76. Abdullah, M., Iqbal, W., Mahmood, A., Bukhari, F., & Erradi, A. (2020). Predictive autoscaling of microservices hosted in fog microdata center. *IEEE Systems Journal*.

Part IV
Privacy and Security Issues in Multi-cloud Environment

Chapter 14
Functionalities and Approaches of Multi-cloud Environment

A. Vijayalakshmi and Hridya

14.1 Introduction to Cloud Computing

Cloud computing is an evolving paradigm where various resources and applications are moved to the Internet so that the applications can be remotely shared among different resources. It enables ubiquitous network access to a pool of shared resources [1]. National Institute of Standards and Technology (NIST) defines cloud computing as a model that enables access to the shared pool with various configurable computing resources like networks, servers, applications, and services [2]. In this technological era cloud computing is considered as a very prevailing network architecture that performs complex computations. The various characteristics of cloud computing include infrastructure sharing where a virtualized software model is used in sharing the services and storage, providing services based on demand of the users by allowing access through Internet for PCs, laptops, etc. as shown in Fig. 14.1.

While using the shared resources, users are charged based on their usage in the particular billing period. The advantage of using cloud from a user's perspective is that it is inexpensive and very convenient as application need not be installed in the personal system; rather it can be accessed through the Internet. The various vital components of cloud computing are:

1. Cloud computing provides services for the needy on time. Any consumer who is in need of computing resources, namely, software resources, CPU time, etc., for a particular timeslot can access cloud services without any human interaction with cloud providers.

A. Vijayalakshmi (✉) · Hridya
CHRIST (Deemed to be University), Bangalore, India
e-mail: vijayalakshmi.nair@christuniversity.in; hridya@btech.christuniversity.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_14

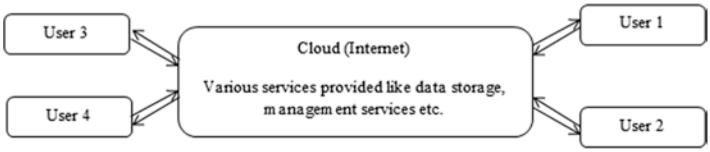


Fig. 14.1 Cloud characteristics

2. The services provided by cloud are accessible over the Internet by applications with diverse platforms that are at the consumer side.
3. The resources in a cloud service provider are pooled together so that multiple consumers are benefitted [3].

14.1.1 Architecture of Cloud

Basic 3 Tier cloud architecture is depicted in Fig. 14.2. The three Tiers in this architecture includes Load balancer server as the top layer, Application server in the second layer, and Database server in the bottom layer. Each of the Tier consists of dedicated server.

14.1.2 Service Models

Cloud computing is a model that is developed with the objective of convenient on demand access and usage of computing resources from a shared pool with minimalizing the interaction from the service provider [4]. The service from cloud server is categorized into three parts as shown in Fig. 14.3.

14.1.2.1 Cloud Infrastructure as a Service (IaaS)

This is a service offered by cloud computing in which the users access servers, storage, and networking resources as shown in Fig. 14.4. The advantage of using this service is that users need not purchase hardware and instead can pay for IaaS on demand and hence saving the cost of purchase and maintenance of hardware resources.

Servers: IaaS providers function to manage physical machines in various data centers around the world for powering the various layers of abstraction so that the services of these machines are made available to end users over the Internet.

Storage: The types of storage provided by cloud are block storage, file storage, and object storage. Block storage is where data files are stored in the cloud server for a fast and efficient data transfer. File storage help in application to access data

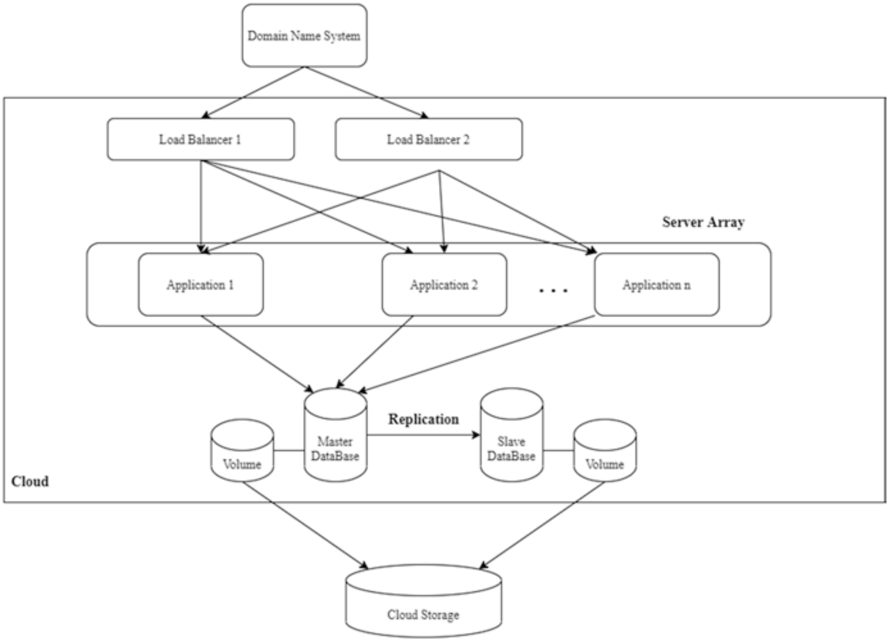


Fig. 14.2 Cloud architecture

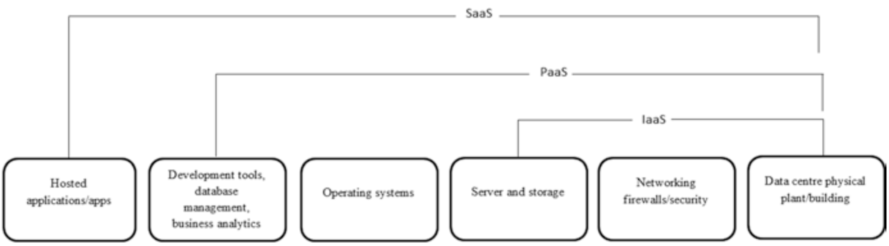


Fig. 14.3 Cloud services

through shared files. Object storage is highly distributed and is the most common mode of storage in cloud.

Network: In networking, the networking hardware like routers and switches are made available through APIs.

14.1.2.2 Cloud Platform as a Service (PaaS)

This is a service where cloud offers consumers with an environment where they can develop and manage various applications as shown in Fig. 14.5. PaaS provides platform for developing and testing applications, hence helping users not to worry on

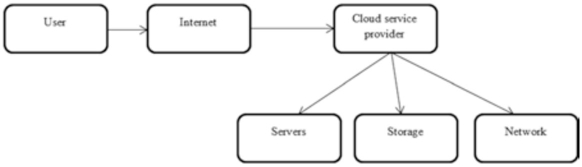


Fig. 14.4 Cloud resources

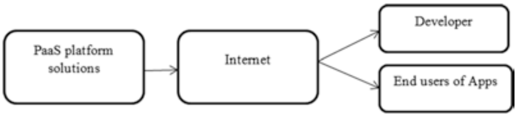


Fig. 14.5 Platform as a service

the infrastructure for the development of applications. This facility accomplishes the management of operating system and security by assisting team work.

14.1.2.3 Cloud Software as a Service (SaaS)

This is a feature offered by cloud server where the consumers can access cloud-based software. The applications that are installed in the cloud networks could be accessed by the users through web or an API. This is provided by the vendors through subscription, and use of resource is completely depending on the need as depicted in Fig. 14.6.

14.1.3 Deployment Models

Cloud community has defined the following deployment models [4].

Private cloud: Here the cloud server is solely operated and managed by a particular organization which can be located at the premise of the organization or outside. Security of data is one of the major concerns of building a private cloud. Apart from security, the cost incurred in transferring data from local system to public cloud is also a factor in developing a private cloud by organizations.

Community cloud: In this model several organizations with common objective jointly construct and share the same cloud infrastructure. Thus developed cloud infrastructure may be maintained and managed internally or by a third party vendor.

Public cloud: This form of infrastructure is used by normal public consumers whereas the service providers have complete ownership in managing the cloud with a detailed policy, cost, and charging, and hence this model is the most prevailing form of cloud deployment models.

Hybrid cloud: This model is a combination of two or more deployment models which is opted by organization to optimize the resources (Table 14.1).

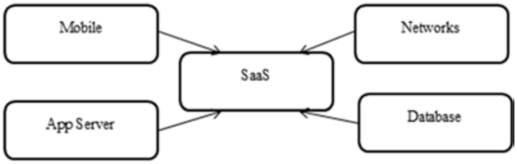


Fig. 14.6 Software as a service

Table 14.1 Pros and cons of various clouds

		Public cloud	Private cloud	Hybrid cloud
Pros	1.	Very simple to implement as well as use public cloud	Organization has a complete control over server updates and patches	Cost-efficient cloud strategy and flexible on utilization
	2.	Public clouds are bound to have negligible straightforward costs	Long-term expenditure is minimal	Hybrid clouds are less prone to extended services
	3.	Server virtualization can lead to utilization efficiency gains	Server virtualization can lead to utilization efficiency gains	Server virtualization can lead to utilization efficiency gains
	4.	Public clouds are widely accessible	–	Hybrid clouds are appropriate for large workloads
	5.	Public clouds do not need any space specifically dedicated for data center	–	–
Cons	1.	Public clouds are most expensive in long term	The upfront costs are large	Implementation is difficult as management schemes are complex
	2.	These types are susceptible to continued service outages	Private clouds are susceptible to continued service outages	Hybrid clouds require adequate space specially committed for data center
	3.	–	Access to the data is restricted	–
	4.	–	The space devoted to data center is large	–
	5.	–	Private clouds are not suited for handling large spikes in work load	–

14.2 Issues in Cloud Computing

Though cloud computing is a boon to the world of smart computing, there are few issues associated with cloud computing. Secure outsourcing of data is a major concern in cloud computing paradigm. The user making use of services provided by the cloud should be aware that the data sent to the cloud as well as the applications deployed in the cloud is controlled by the cloud provider. A strong trust should be

Table 14.2 Cloud computing issues – security strategies

Privacy model	Errors are difficult to find as well as correct System does not provide protection against attackers
Cloud computing privacy intrusion detection	Frequent update is necessary to avoid attackers There is a chance that non-intrusive information is incorrectly detected and terminated
Dirichlet reputation	Depends on complicated strategy that is difficult to implement Performance is dependent on user participation
Anonymous bonus point	The speed of data is reduced Intrusion protection is minimal Can be implemented in wireless application only
Network slicing	May be expensive while implemented in large network

formed between the user and the provider on the security of these data, and security plays a major issue in cloud computing [5–7]. The issues in cloud computing include various attacks on cloud interfaces as well as misusing the cloud services in order to attack other systems in the network [3]. Lists the difficulties in cloud computing with respect to various security strategies in Table 14.2.

Cloud computing provides different supports to the users through the Internet for storage, networking, software, and as server. Cloud storage helps in storing files and data in a remote storage that has access through the web. Cloud storage is preferred by people because it helps in saving the cost of resources for storage, high speed, and efficiency and increases the overall productivity. Cost saving and availability are the advantages of cloud computing. However, it carries along various challenges in terms of security of data. Cloud security is a major concern in people using cloud storage facility. It is very much necessary that proper security is provided to the sensitive data that are stored in cloud servers like credit card or debit card details, health information, etc. Cloud computing should take care of data breaches, loss of data, traffic attacks, distributed denial of service attacks, etc. The security concerns that are to be attended are data prevention, web security, location, identity and access management, recovery, data loss prevention, web and e-mail security, security assessments, regulatory compliance, violation management, event management, encryption, data segregation, business continuity, and disaster recovery. When data is stored in a remote database, the major concern is the security of data. Cloud service provides various security measures for the data stored in cloud [8].

14.3 Introduction to Multi-cloud

Cloud computing is a service provided to host applications and data in remote servers. The change in digital era has driven the companies toward multi-cloud architecture that enables shared services. Multi-cloud is a cloud computing approach made up of two or more cloud environments where applications are presented among the heterogeneous networks. The transition from single cloud to multi-cloud data centers has always benefited the industry. This transition indicates the shift of traditional data centers to a pool of multiple private clouds on demand. Dedicated data servers from a private environment to hybrid and multi-cloud are deployment models that integrate more than one cloud. It differs from hybrid cloud in the cloud infrastructure. Hybrid cloud has a combination of two or more than two different types of clouds, whereas multi-cloud integrates different clouds of similar types. Hence, the term multi-cloud can be referred to as a cloud system where different cloud networks are combined to perform various roles. It is evident from the studies conducted that multi-clouds can address the security issues in cloud computing that are related to date integrity, intrusion, and service [9].

14.3.1 Single Cloud to Multi-cloud

NIST reported that services from different clouds can be used simultaneously or serially from one cloud to another [10]. Different multi-cloud use cases can be used to the maximum to provide flexibility and control over the workloads and cloud stored data. Since multi-cloud offers flexibility in cloud environment, the organizations are intended to meet the requirements of technical and commercial workload by adopting multi-cloud environment. Another advantage of multi-cloud is seen in the case geographical advantage in order to address the problems associated with latency issues. Also, there are times when the organizations use a certain cloud service for a short period in order to achieve their short-term goals. Apart from these, vendor lock-in problem is an issue which is a frequent concern in cloud computing that can be solved when multi-cloud strategy is used. Various multi-cloud use cases can be listed as follows [11].

- To access the services and resources based on various demands
- To improve the quality of service and optimize cost
- To use the services from various providers
- To avoid the dependency on a single provider
- To create backups of data so that disasters are dealt efficiently

14.3.2 Architecture of Multi-cloud

Multi-cloud architecture design is an appropriate solution for a reliable application being stored in cloud. Figure 14.7 shows a prominent multi-cloud architecture strategy where an application component can use different cloud services of other cloud platform and improve the overall performance.

To demonstrate cloudification, the Application-1 uses Amazon Web Service (AWS) for storage and Azure for computation. This improves availability and avoids

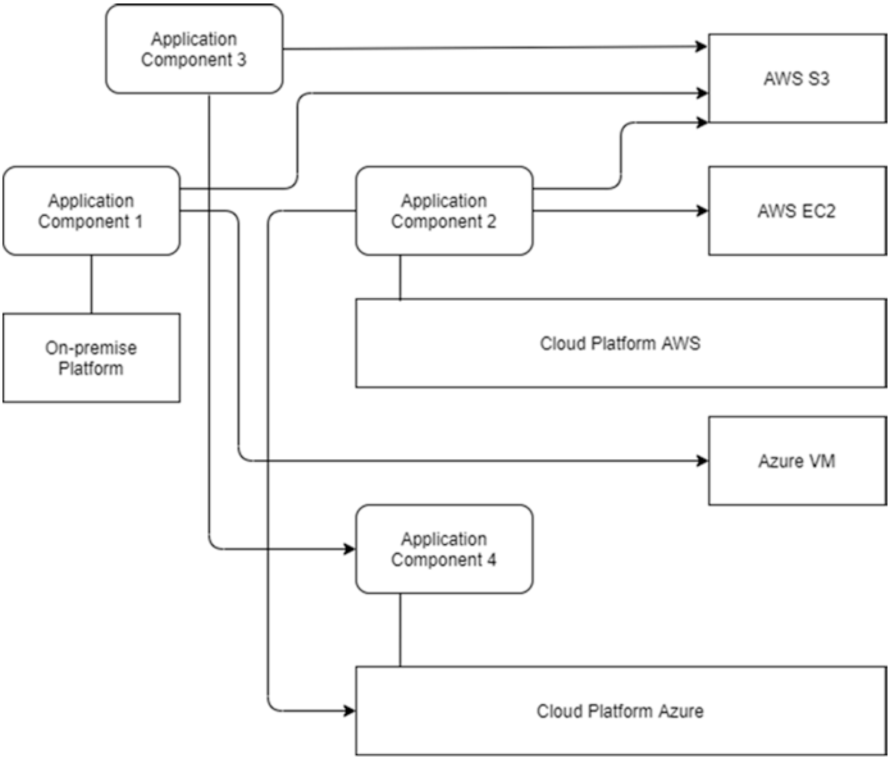


Fig. 14.7 Multi-cloud architecture

vendor lock-in. Application-2 demonstrates multi-cloud relocation where the application is re-hosted on AWS cloud platform and privilege is given to use environmental services provided by Azure. This method enhances the availability as the application that is re-hosted avoids vendor lock-in. Multi-cloud refactor is demonstrated using application component3 deployed on AWS using AWS3 and application4 on Azure so that the application can use any of the cloud service provided by Azure as per the requirement of the application. This method provides optimal performance and quickness to respond to changes [12].

14.3.3 Benefits of Multi-cloud

Multi-cloud offers various advantages and the most important of these is flexibility. With organizations adopting multi-cloud strategy, the organization has flexibility in choosing from various cloud service providers. Multi-cloud allows choosing varied set of services that are provided by various cloud providers. Managing deployment on various clouds is another important functionality of multi-cloud [11]. Ability in distributing applications and services geographically is another benefit of multi-cloud computing. The user experience will increase as the app is closer to the target user [13]. Below are the advantages of multi-cloud listed as per [12].

Disaster recovery: It is difficult and risky when a particular organization has to manage its resources when it uses a cloud service. There can be cyber-attacks that can affect the end users to wait to access the data until the problem is solved. Using multi-cloud can help the users robust against such attacks as there are other clouds that can carry the workload when any one of them is affected.

Vendor lock-in: With multi-cloud, the various organizations have the privilege to choose the best service according to their objective of the organization. Hence while using multi-cloud, organizations can explore various providers and find the best match for each component of their operations.

Managing data: An organization can develop data that need to be accessed in varied intervals. There are cases in which some data may be stored in the database and not accessed for a long time, and some of them will be accessed in frequent intervals. In such cases, rather than storing these data in single cloud, you can make use of the right cloud for right function. An organization can also have data that are to be uploaded to the cloud, analyzed, and downloaded back. In such instances, the organization prefers a cloud service with speed and power. Hence, with multi-cloud, the organization has the privilege to access the right service depending on the frequency in which different data need to be accessed.

Cost optimization: The performance analysis of the workloads in various cloud platforms will help in saving the cost.

Low latency: In an environment with multi-cloud, the data center closest to the end user can serve the data requested from the end user with minimal delay. This can lead to unified experience for the user.

14.4 Security in Multi-cloud Architecture

The objective of cloud computing is to provide various resources to the end user as a service over the Internet. Security of data in cloud is an important aspect when quality of service of cloud is considered. Due to dynamic nature of cloud, various security threats emerge [14]. Cloud servers contain humongous confidential sensitive data. By using multiple distinct clouds and splitting data based on various components, threats associated with tampering of data can be brought down. Thus the confidentiality of data is maintained with multi-cloud strategy. When an organization uses multi-cloud strategy, it has the provision to use multiple distinct clouds at an instant of time so that it can reduce the risk of malicious data and tampering of data. The integration of distinct clouds makes it difficult for an attacker to tamper the data hosted in the cloud server [6, 9]. The concept of using multiple cloud was suggested by Bernstein and Celesti [15, 16] and various instances of cloud computing working together and further recommended grid capabilities in cloud computing.

Considering shared environment like cloud, giving maximum protection to the privacy of data and identity of clients is very necessary. Amazon S3 and Microsoft Azure which provide data as a service (DaaS) are the two very important and widely used cloud service where organizations can store data and can retrieve whenever needed based on various policies for gaining access. XACML is an access control model that specifies identity attributes like an individual's email address, age, location access, etc. to provide fine-grained data access. With cloud computing, data theft and privacy breach is not limited to the circle of an organization [17].

14.4.1 Algorithm for Data Security in Cloud

14.4.1.1 Encryption Algorithm

Encryption algorithms like RSA (Rivest, Shamir, Adleman), AES (Advanced Encryption Standard), XOR (Exclusive OR), and DES (Data Encryption Standard) have been a great advantage while considering security of data in cloud [18]. AES is an encryption standard, where the operation speed is fast and offers high safety on the data. This is a symmetric encryption algorithm which performs both encryption and decryption. The encryption procedure changes with respect to the key that changes the operation of AES algorithm. The encryption formula for AES is

$C = E(K, P)$where C is the cipher text, K is the key, E is the encryption function, and P is the plaintext.

The AES algorithm executes for Nr-1 loops on a 128-bit block of data. Nr (10,12 or 14) depends on the key length which can be 128, 192, or 256 bits in length. The steps in encryption are listed below.

Step 1: Sub Bytes.

This step is known as substitute bytes transformation which includes a 16 × 16 matrix as a table look called s-box. The matrix has the combination of 8 bits sequence. The left most 4 bits are used, the leftmost nibble of the byte is used to specify a particular row of the s-box, and the rightmost 4 bits indicate a column.

Step 2: Shift Row Transformation.

In this step, the rows of a particular state are shifted in a circular way toward left.

Step 3: Mixing of column transformation.

In this stage, a substitution takes place where columns are individually operated and every byte of a particular column is mapped with a new value that is a function of all the four bytes in the column.

14.4.1.2 Data De-duplication Technology

Data De-duplication is the process in which all the duplicate data are eliminated to reduce the storage required for these data. This in turn reduces the bandwidth for transfer of data across the network. This technique is a great benefit to the cloud users in managing their data without duplication and it saves the storage. For secure data de-duplication, Rabin block algorithm is used which involves block-wise de-duplication of data.

14.4.1.3 Cloud Storage Technology

Cloud storage is a file server with complex functionality that can be helpful in storing and accessing stored data as and when required. The end user or the organizations upload the data and store the same on cloud servers that ensure security of these data. If there is a server failure, accessing the stored data will be difficult. To solve this, all the data nodes are added to the node at the backup to make sure data is available.

14.4.2 Good Practices in Multi-cloud Strategy

Organizations are increasingly adopting multi-cloud strategy with the fast development in the field [19]. If multi-cloud strategy is used with identical operations on two clouds, security settings should be the same across both the clouds. This is achieved by synchronizing policies across the cloud providers:

- If an organization is making use of varied workloads, independent security policies have to be created for every service.
- Automate various tasks of an organization to reduce the risk factor and ensure security.
- Choose the appropriate tool such that it allows to synchronize the security protocols across various cloud service providers.
- Create a strategy to monitor security that can consolidate logs and alerts from various platforms in one location.

14.5 Conclusion

The Internet has emerged into accumulation of humongous amount of data all around the world. Cloud computing has been of great advantage in order to store these data and access when in need. Data stored in cloud can be accessed from a remote server with the help of various services offered by cloud service providers. Transition from single cloud to multi-cloud is a smart venture for the organizations to achieve greater performances. This chapter discusses the transition of cloud computing from single cloud toward multi-cloud. It also drives into the benefits of using multi-cloud strategy.

References

1. Ferry, N., Rossini, A., Chauvel, F., Morin, B., & Solberg, A. (2013, June). Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In 2013 IEEE sixth international conference on cloud computing (pp. 887–894). IEEE.
2. Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.
3. Hu, F., Qiu, M., Li, J., Grant, T., Taylor, D., McCaleb, S., … Hamner, R. (2011). A review on cloud computing: Design challenges in architecture and security. *Journal of Computing and Information Technology, 19*(1), 25–55.
4. Dillon, T., Wu, C., & Chang, E. (2010, April). Cloud computing: Issues and challenges. In 2010 24th IEEE international conference on advanced information networking and applications (pp. 27–33). IEEE.
5. Hong, J., Dreibholz, T., Schenkel, J. A., & Hu, J. A. (2019, March). An overview of multi-cloud computing. In *Workshops of the international conference on advanced information networking and applications* (pp. 1055–1068). Springer.
6. Bohli, J. M., Gruschka, N., Jensen, M., Iacono, L. L., & Marnau, N. (2013). Security and privacy-enhancing multicloud architectures. *IEEE Transactions on Dependable and Secure Computing, 10*(4), 212–224.
7. Singh, Y., Kandah, F., & Zhang, W. (2011, April). A secured cost-effective multi-cloud storage in cloud computing. In 2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS) (pp. 619–624). IEEE.
8. Liu, W. (2012, April). Research on cloud computing security problem and strategy. In 2012 2nd international conference on consumer electronics, communications and networks (CECNet) (pp. 1216–1219). IEEE.
9. AlZain, M. A., Pardede, E., Soh, B., & Thom, J. A. (2012, January). Cloud computing security: From single to multi-clouds. In 2012 45th Hawaii international conference on system sciences (pp. 5490–5499). IEEE.
10. Sokol, A. W., & Hogan, M. D. (2013). NIST cloud computing standards roadmap.
11. Petcu, D. (2013, April). Multi-cloud: Expectations and current approaches. In Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds (pp. 1–6).
12. 6 Multi-Cloud Architecture Designs for an Effective Cloud Strategy. *sim*. https://www.simform.com/multi-cloud-architecture/.
13. Akinrolabu, O., New, S., & Martin, A. (2019, June). Assessing the security risks of multicloud saas applications: A real-world case study. In 2019 6th IEEE international conference on cyber security and cloud computing (CSCloud)/2019 5th IEEE international conference on edge computing and scalable cloud (EdgeCom) (pp. 81–88). IEEE.
14. Pawar, P. S., Sajjad, A., Dimitrakos, T., & Chadwick, D. W. (2015, May). Security-as-a-service in multi-cloud and federated cloud environments. In *IFIP international conference on trust management* (pp. 251–261). Springer.
15. Bernstein, D., Ludvigson, E., Sankar, K., Diamond, S., & Morrow, M. (2009, May). Blueprint for the intercloud-protocols and formats for cloud computing interoperability. In 2009 fourth international conference on internet and web applications and services (pp. 328–336). IEEE.
16. Celesti, A., Tusa, F., Villari, M., & Puliafito, A. (2010, July). How to enhance cloud architectures to enable cross-federation. In 2010 IEEE 3rd international conference on cloud computing (pp. 337–345). IEEE.
17. Singhal, M., Chandrasekhar, S., Ge, T., Sandhu, R., Krishnan, R., Ahn, G. J., & Bertino, E. (2013). Collaboration in multicloud computing environments: Framework and security issues. *Computer, 46*(2), 76–84.
18. Yao, J., & Jiang, X. (2020, March). Research on multi cloud dynamic secure storage technology. In 2020 International conference on computer engineering and application (ICCEA) (pp. 72–78). IEEE.
19. 8 Best Practices for Multi-Cloud Security. https://blog.checkpoint.com/2019/12/20/8-best-practices-for-multi-cloud-security/.

Chapter 15
Quality, Security Issues, and Challenges in Multi-cloud Environment: A Comprehensive Review

M. G. Kavitha and D. Radha

15.1 Introduction

Cloud computing has changed the manner in which IT administrations are conveyed and devoured. Every affiliation needs to re-fit and reserve data and use database which is used as an assistance. The expense for information stockpiling including information recuperation is amazingly large for organizations making it a practical option in contrast to customary capacity technique. This has caused an ongoing pattern in the corporate world, and numerous associations are changing to a more distributed computing-based systems administration agreement. It is a bunch of problematic innovations that have encouraged new plans of action and another industry design. Associations hope to exploit its nimbleness, usefulness, adaptability, and money saving advantages. The cloud empowers modularization, normalization, and centralization of center skills, amplifying an incentive for all entertainers included. The utilization of multi-cloud is continually developing the same number of ventures use it to satisfy distinctive business needs. The use of various mists has additionally offered ascent to the development of multi-cloud applications.

With multi-cloud and multi-cloud applications, cloud customers and application owners can mix particular help commitments and influence on the middle characteristics of different Cloud Service Providers (CSP) to meet their varying necessities. The security stresses in multi-fogs ooze from the heterogeneous thought of the

M. G. Kavitha (✉)
Department of Computer Science and Engineering, University College of Engineering, Pattukkottai, India

D. Radha
Department of Computer Science and Engineering, Arifa Institute of Technology, Esanoor, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_15

multi-cloud environment, as it has different attack surfaces, which open it to a couple of risks. This makes the multi-cloud environment complex to manage. It similarly makes it difficult to evaluate the security status and all in all execution of the multi-cloud application fragments similarly as the entire multi-cloud environment.

Multi-cloud access control innovation is a security instrument that guarantees asset sharing and secure interoperability in multi-cloud. It deals with the entrance authorization to forestall interruption of illicit clients and authentic client's abuse. The significant destinations of multi-cloud access control innovation are (1) to accomplish asset sharing and interoperability among different mists; (2) to keep unlawful clients from getting to ensured asset in multi-cloud; and (3) to keep real clients from getting to unapproved asset in multi-cloud.

There are no strategy clashes in a solitary cloud climate, since entrée control strategy is standardized and overseen consistently. Notwithstanding, the heterogeneity of multi-cloud impacts asset distribution and protected interoperability is noticeable. Consequently, it is critical to take care of the issue of multi-cloud access control strategy incorporation. Today, there is a fast advancement of worldwide and neighborhood organizations. The real issue of present-day networks is the yearly development of united traffic. As per the scientific information of the main organization gear sellers, for example, Cisco and Huawei, the capacity of transfer is increasing yearly by 20–35%. The traffic development is fundamentally due to digital assaults on these corporative cloud administrations and applications.

Evaluating the vector of assaults on the foundation that bolsters the activity of data frameworks in huge organizations, it tends to be presumed that the real clients are pointed toward confining admittance to the fundamental assault to key applications (35%), an infringement of the cycle gear (35%), and obliteration or robbery of classified information (23%). Corporate clients want to take digital protection offices for lease. To do this, corporate data assets are situated in server farms. In one server farm, figuring limits typically lease a few organizations without a moment's delay.

The principal assignment of server farm administrators is to gather the necessities of shared clients to guarantee the security of rented framework from outside impacts, just as to guarantee nature of administration as indicated by the Service Level Agreement (SLA). Be that as it may, given the measure of information entering the server farm and the quantity of clients working at the same time with a similar hardware, there is a danger of blunders in the arrangement of organization gear. This can prompt negative ramifications for the server farm of the administrator, just as for corporate customers it serves. In this manner, administrators of server farms like to confine the foundation of corporate clients with utilizing multi-cloud stages.

15.2 Related Work

15.2.1 Cloud Computing and Cloud Systems

The idea of cloud computing targets giving registering administrations to cloud clients at whatever point they want. The cloud computing representation guarantees that endeavors can zero in on their center fitness and devour cloud assets as indicated by their necessities.

15.2.1.1 Safety Challenge in Cloud Computing

The guarantee of resources is the principal objective of security. With regard to data security, the primary objectives are secrecy, respectability, and accessibility. In distributed computing, it has gotten important to stretch out these objectives to oblige assault versatility and danger location attributable to the high danger of openness of the cloud. The solution safety challenges are introduced in the following sections.

15.2.1.2 Multi-clouds and Multi-cloud Applications

Multi-clouds include the utilization of numerous cloud administrations by a venture to accomplish a business objective. In this arrangement, a specific service provider might be utilized for giving distinctive cloud administration models or numerous service providers for giving a similar cloud administration model. In particular, unique service provider and cloud administration models are blended in an ideal way that causes the endeavor to meet its goal.

Multi-cloud likewise helps to fulfill information administration prerequisites especially in situations where information should dwell in specific areas attributable to information guidelines. The utilization of multi-cloud makes it conceivable to store the information in service provider offices inside the predetermined locale, while other application parts are facilitated in another as wanted. This guarantees that cloud clients can get the best blend of various CSP contributions in understanding how they fulfill their necessities [1].

15.2.2 Security Challenges in Multi-cloud Computing

The safety challenges in multi-cloud can be ascribed to the degree of enthusiasm and intricacy inside the climate. In multi-clouds, there will be a few interfaces and end sections, which brings about various assault surfaces and subsequently expands the degree of security dangers and weaknesses. Pernicious cloud clients and aggressors exploit this to do destructive acts [2].

It hence gets essential to guarantee satisfactory straightforwardness and security attention to multi-cloud climate segments to distinguish malignant practices or activities, appropriately secure the interfaces, and forestall any pernicious demonstrations. The fundamental security challenges in multi-mists as recognized in various examinations are introduced straightaway.

15.2.3 Cloud Methods

The Software-as-a-Service (SaaS) model permits the purchasers to benefit the predefined composite activities through the Internet from the cloud specialist organization. The goal here is to convey the function of administrations to the clients according to their necessities. Subsequently, the end clients need not stress over the issues, for example, establishment and future upkeep overheads.

The subsequent assistance called Platform-as-a-Service (PaaS) is an improbable representation for contribution to the cloud client, the registering stage that incorporates working frameworks, program creating climate, web workers, and database systems. From this model, a stage asset can be shared and re-utilized among the buyers. The third assistance, in particular, Infrastructure-as-a-Service (IaaS) implies conveying the distributed computing framework, for example, substantial PCs, stockpiling, organization, and related assets for the sending of client working framework and application programming. Through this, the clients can possess the peripherals and can design them according to their desire. Other than the three administrations, a cloud client can benefit such an administration, considered to be called as Anything as a Service (XaaS).

Infrastructure-as-a-Service (IaaS), from various suppliers. Notwithstanding, the current cloud expediting plans infrequently consider the advancement procedures of sending cloud administrations concerning different models, for example, price and execution. This paper, suggests a hereditary based method to deal with multi-cloud administration issue. For example, choosing and renting virtual machines for cloud clients at minimum expense in addition to organization latency. [3].

15.2.4 Distributed Computing and XACML

Cloud service organizations and free source people group have planned for the framework engineering and specialized plan of way in manage to guarantee the security of cloud administrations. Amazon Web Services upholds access strategy for various systems, URLs. Access strategy systems are comprehensively classified into asset-based approaches and client arrangements. Google Cloud [4, 5] have two significant alternatives accessible for receiving items and compartments, and the choice Access Control Lists (ACLs) and sign URLs. OpenStack has a free segment

(Keystone) for validation and approval, which gives access control components, like Temporary URLs, RBAC, and LDAP.

Cloud Stack in along with Eucalyptus don't have free segment for admission control, however they execute access control by situation passage policy and surge rules for Virtual machines. Asset area oversees client, security gathering, virtual machine, and organization access in Cloud Stack. Cloud regulator is responsible for character validation and client administration in Eucalyptus.

15.2.5 Distributed Storage Intimidation and Attack Surfaces

Conveyed Storage Services (CSS) provide gigantic violence to surfaces, in solicitation gives viably open Application programming interface driven shows for instance Hypertext transfer protocol. The revealed show aggression surfaces are fundamental to identify with while arranging and passing on applications and organizations that devour conveyed capacity.

While the greater part of these dangers are after effect of mis-approach blunders by center passed on limit sections, for example, holders, different assaults, insider assaults are additionally conceivable. Essentially, wrongly arranged or assets (e.g., nonattendance of adherence to the standard of least favored index) increases odds of bigger assaults [6].

Data reliability and insurance are the two essential fundamental safekeeping issues connected to customer in order. Normally, information owners obtain the genuine kind of a contraption anywhere the data is being kept, for instance, a hard drive. Hereafter, we could acknowledge data is in safe hands with no outside social occasions incorporation. In inverse, into shade amassing, the data is taken care of on a practical zone of expert association's storerooms [7]. The trust level between these two social occasions is regulated by the methodologies deduced by the expert associations.

15.2.6 Cloud Services Brokerage

In distributed computing situation, a middle coating that encourages a downy connection among the cloud supplier and client assumes a functioning job, and it prompts the idea of cloud administrations business. The administration business layer upgrades center cloud functionalities, for example, administration determination, administration coordination, administration total, administration customization, quality confirmation, and administration streamlining. Other than the job of administration determination, a cloud offers different piles of administrations for administration purchasers. We have recognized the conceivable exploration issues of specialist-based assistance determination in distributed computing. Despite the detail, the purpose of a section of the works are endeavored to get better idea to the financier. It ought to be advanced to help the cloud clients in the part of their calculation for framework arrangement and prospect assumptions.

15.2.6.1 Insightful Cloud Broker

A cloud specialist is a product function, which is projected to fill in as an arbitrator among the shopper and supplier. The cloud intermediary offers different options worth included for the sake of the administrations gatherings; it is more fitting to assemble the agent with the incorporation of smart computational procedures.

15.2.6.2 Customer Portal

This entrance is intended to streamline the prerequisites determination of cloud clients. Since client necessities are planned regarding administration needs, it is fitting to assemble them prior to starting the choice process.

15.3 Security Issues

15.3.1 Interoperability Issues

Interoperability of one cloud specialist to another cloud specialist is imperative to make client to look for furthermore from additional from distributed computing. Without normalization interoperability of explicit application and administration usefulness starting with one cloud then onto the next is unthinkable and it confine the usage causing precariousness in territory, for example, security. The technical interoperability quality viewpoints allude to the limit of at least two administrations offered by various suppliers to convey through basic conventions and together to ensure a specific nature of administration.

15.3.2 Security or Confidentiality

As the administrations are teaming up together, security of the information is very significant in light of the fact that there is a development towards various specialist organizations so that there must be an administration agreement (SLA) or terms and condition to be applied, so that a client of another cloud specialist cannot be able to change the information or data. SLAs (Service Level Agreements) assume a focal function in the administration lifecycle. SLAs are arrangements restricted to depiction of desires and obligations. An SLA can't ensure that client will get the administration it portrays; a similar time, an SLA can alleviate the danger of picking an awful assistance.

15.3.3 Trustworthiness

The information put away in the cloud might be tainted or harmed during progress activities from or to the distributed storage supplier. Keeping up consistency of information is significant.

15.3.4 Administration Availability or High Availability

The client's web administration may end under any circumstances if any of the client's documents break the distributed storage strategy. Administration accessibility in multi-cloud can be ensured by keeping up the reinforcement duplicate in another cloud so that on the off chance that one cloud is down, the information can be recovered from another cloud; in this manner the client will get administrations on all.

15.3.5 Confirmation and Authorization

Another safety exposure that possibly will ensue with a cloud seller, for example, the Amazon cloud administration, is a hack undisclosed phrase or data interruption. Somebody accesses a secret word; they will have the option to get to the entirety of the record's examples and assets. On the off chance it a customer validation got hacked a fakes client can login and can be able to degenerate the information.

Safety vulnerability may occur with a cloud contractor, for example, the Amazon cloud administration, is a hacked undisclosed phrase or data interruption. If somebody accesses secret phrase; they will have the option to get to the entirety of the record's cases and assets.

15.3.6 Information Proprietor

The client transfers the information to the cloud utilizing the ambiguity key. After getting the solicitation from the beneficiary, the information proprietor shares the mystery key and document name. The significant part of the information proprietor is to keep up the approved client's rundown alongside mystery keys.

15.3.6.1 Mystery Key

There is the adaptability to deal with the mystery keys in cloud. It has been done in three steps: first one is next to the administration supplier side, second one is outsider worker region, and the third one is the information proprietor side. The future system guarantees the information parceling, ordering, and scrambling of information, stockpiling in the multi-cloud. There is no likelihood to get to the information from the cloud without the consent of information proprietor. The proposed system gets the information from the client, and it parts the information by relegating the lists to each part. The high-level encryption calculation has been created to scramble the information. The encoded information parts will store in the multi-cloud climate. In the event that any client needs to get to the information, they need to put the solicitation for the information proprietor. The information proprietor will send the mystery key alongside record name to the mentioned client through a preferred channel. The mentioned client presents the record name to the multi-cloud climate and recovers the encoded information section from the multi-cloud. The decoding component is functional by the mentioned client and recovers the information. The blending activity is performed on the information and provisions them into the nearby appliance of the mentioned user [8].

15.3.7 Cloud Intermediary Design

A cloud intermediary is a product appliance which is being intended to fill in as an umpire linking the customer and supplier. Despite the fact that the cloud intermediary offers different and worthy administrations benefit of the gatherings. It is more suitable to assemble the representative with the incorporation of keen computational methods. Such sorts of strategies encourage clever thinking and dynamic ability to the specialist. From our point of view, the keen merchant is characterized as, “a product element dependent on the cloud financier model that offer cloud administrations with the guide of computational knowledge procedures, for example, computerized reasoning, delicate registering, AI and information mining towards giving a stage to both cloud clients and suppliers.” The proposed intermediary design contains (1) User Interface, (2) cloud cosmology, (3) administration revelation, and (4) IaaS administration vault.

A cosmology is a portrayal of various cloud ideas with its connections to encourage the thinking among a wide range of cloud administrations. It empowers the clients/machines to deal with the data all the more correctly and helpfully. Regarding our situation, the cloud merchant develops a cloud cosmology for the viable portrayal of IaaS sorts of cloud administrations with sensible financial plan.

Administration disclosure in distributed computing prompts locates a reasonable cloud administration that basically alludes to the revelation of administration portrayal. An administration depiction contains the utilitarian, non-useful abilities alongside the quality of an administration. The disclosure is obliged by practical, specialized details of the administrations, and the budgetary requirements with proper security arrangements of the administrations are called “administration ideas.” The importance of administration ideas, for example, its availability among the accessible assistance occasions, is spoken to as semantic nets.

15.4 Security Evaluation Framework

The security assessment structure is pointed toward giving a way to deal with assessing security in multi-cloud application. It comprises of various sections alluded to like motors [9]. These motors will be liable for giving the classification tasks.

15.4.1 Structure Operations

The proposed security assessment structure is comprised of various tasks, which incorporate threat ID and danger investigation, determination of measurements and safety controls, function safety observing, security estimation, and dynamic and security status representation.

15.4.2 Risk Identification and Risk Analysis

Warning ID includes a cautious investigation of a purpose to decide vindictive behavior, exercises, and tasks that may influence the application segments. These noxious activities, exercises, and tasks are viewed to as bullying, and their assurance might be accomplished from side-to-side danger displaying procedures as introduced in threat examination and then again gives a way to assessing and rating the recognized dangers to appraise the degree of criticality of the risks.

15.4.3 Choice of Metric and Safety Controls

Following risk distinguishing proof and threat examination is the choice of measurements and safety controls. The determination of measurements includes indicating experimental and screen capable boundaries needed for acquiring specific data about the application practices [10].

15.4.4 Application Security Monitoring

Application safety measures check includes noticing the multi-cloud function to recognize any type of surprising conduct. In particular, a bunch of security measurements is checked to determine whenever wanted degree of execution is met or if an infringement of the measurements has happened. Measurements infringement is a decent sign for the rise of a danger. The degrees of the safety measurements are observed utilizing security checking tests introduced on the virtual machines in which the request runs.

15.4.5 Security Measurement

Security estimation includes joining security goals. It applies towards the estimation of the rate point of metric satisfaction and the refuge remainder of the multi-cloud function. The safety remainder is a proportion of the safety level of the multi-cloud function.

In particular, safety estimation includes the utilization of numerical principles and formula taking place in the checked safety measurements, to figure the security remainder of the multi-cloud application as far as privacy, respectability and accessibility.

15.4.6 Dynamic and Safety Measures Status Perception

In the dynamic activity, the consequences of the security estimation practice are assessed to set up regularity or protection break.

This is complete from first to last correlation of safety measures estimation outcome with determined measurements edge to find out conformance or infringement [11]. In the event of an measurements infringement is noticed, the cycle of security reins submission is set off to relieve the danger. As in respects to the security position of the representation.

15.4.7 Data Possession in Multi-cloud Storage

Numerous enterprises and associations utilize the strategy of distant reinforcement to guarantee the information accessibility and toughness. They construct various topographically scattered server farms to store their significant information and implement the information consistency. On the off chance that one server farm is crushed, the information can be recuperated from other server farms. To lessen the expense of server farm fabricating and keeping up, a few associations lease multi-public distributed storage administrations to store diverse information duplicates, which accomplishes nearly a similar outcome as far-off reinforcement however with much lower speculation.

Since the distributed storage specialist organization isn't completely believed, the information proprietor needs to consider how to confirm the honesty of all information duplicates. This article expects to take care of the issue of information trustworthiness checking for different duplicates on various distributed storage workers. To accomplish this objective, we give the primary character-based PDP convention for numerous duplicates on circulated distributed storage servers [12].

15.5 Construction Mechanism

The planned safety assessment scheme is comprised of the associated segments, safety strategy motor, security estimation motor, capacity checking motor, and choice and inspection motor.

15.5.1 Request Engine

The Request Handling Engine (RHE) addresses the docks of section of the safety assessment system. It fills in as the machinator of the construction and deals with the correspondence between the Framework.

When a security assessment demand is made, it goes directly to the Request, which at that point sends it near the proper motor for handling the security mechanism. The Request likewise gets the reaction from the preparing motors and in the long run makes the security assessment result accessible to the end-client.

The Request is comprised of three parts, to be specific Receiver, Register, and Receptor. The Beneficiary unit is liable for tolerating the solicitation from the end-client, and afterward approves the solicitation by checking the Directory to determine the exactness (Fig. 15.1).

15.5.2 Security System

The security strategy motor empowers the demonstrating of the more than one-cloud function and determination of the data identifying with different motors in the structure. With the application portrayal, data, foundation prerequisites, dangers, security measurements, security controls, and so forth are indicated. The Security system is comprised of four sections, to be specific Inquirer, analogist, effect and

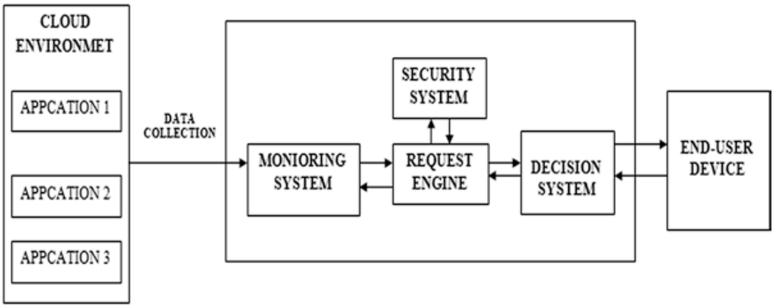


Fig. 15.1 Security framework

Knowledge Base (KB). Approaching solicitations from the Request are acknowledged by the Inquirer and shipped off the rational for additional preparing. The KB contains the safety philosophy, which is questioned by the analogist.

15.5.3 Monitoring System

The monitoring engine empowers security examining of the multi-cloud application. This guides the recognition of any bizarre conduct inside the multi-cloud climate. The monitoring engine arranges the exercises of all the security observing tests introduced on the virtual machines facilitating diverse multi-cloud function segments. It acknowledges stores and amalgamates the security metric qualities revealed by the checking tests. The consolidated measurement esteems are then shipped off the Security engine for safety estimations. The monitoring engine is comprised of three modules, in particular observing worker, size aggregator, and the data set.

15.5.4 Safety Measurement Engine (SME)

The safety measurement engine is the sanctuary mini-computer of the protection assessment system. It figures the rate level of metric satisfaction and the security remainder of the multi-cloud application. The SME applies a bunch of predefined rules and statistical formulae on the safety metric qualities to figure the safety remainder of the multi-cloud function.

15.5.5 Decision System

The decision system is answerable for attractive security assessment choices and showing the safety status of the multi-cloud request to the end-client. In particular, the decision system assesses the security estimation results figured by the Safety engine by contrasting the outcomes and predefined limit esteems to learn regularity or break. Furthermore, it shows the security assessment results on the end-client gadget as diagrams and outlines. The decision system is comprised of three sections, specifically Receptor, Analyzer, and Viewer. The safety estimation results are brought from the security engine by the Receptor.

15.6 Cost of the Administration in a Multi-cloud Environment

It is notable that practically all cloud suppliers can offer similar classes of administrations (virtual machines, virtual capacity, and so forth). From supplier to supplier, administrations are separate as far as the costs/QoS are proposed to the client. With the end goal of this work, the accompanying classifications of administrations will be thought of: Virtual Machines, fixed Public IPs, and Gateways.

Clients can pick any administrations accommodating their application, given that (a) the application can be part into segments, (b) segments can be spread over the islands, and that (c) CTC correspondence can be sufficiently conceded.

15.6.1 Costs Stood to Secure Cloud Administrations

This is the money related expense for leasing VMs, fixed Public IPs, and Gateways.

Security set-up exertion: It is the exertion taken to construct a safe registering climate for the application.

Application set-up exertion: It is the exertion taken to convey the application and accurately set up the systems administration between parts.

Application support exertion: It is the exertion required for some segments to be performed like (a) eliminated from the application, (b) added to the application or (c) moved to an alternate island.

Public IP-based directing (PIPR): All VMs, freely of where they are found, are given a static IP address. Steering among VMs is straightforwardly completed by Internet switches.

Door-based directing: Doors are answerable for steering bundles among VMs dwelling on various islands.

VPN-based routing (VPNR): VPNs are accountable for overseeing directing among VMs having a place with a similar island (intra-island steering) and directing among VMs having a place with various islands (between island directing) [13].

15.6.2 Multi-exhausting in Cloud

The multi-tapering can be accomplished in subsequent manners:

15.6.2.1 Parcel of Use System into Levels

Licenses removal of rationale from information. This gives advantageous fortress other than information spill which may emerge because of issues brought about by application rationale.

15.6.2.2 Segment of Use Information into Fragments

Grants circulation of fine-grained information sections to unique cloud. Few out of every odd cloud has generally able to gain admittance to information which thus guarantees security and classification of information.

15.6.3 Service-Oriented Broker

Web administrations are the type of SOA execution to help interoperable machine-to-machine association over an organization. This interoperability is picked up through a bunch of XML-based open principles. These principles give a typical way to deal with characterizing, distributing, and utilizing web administrations. With regard to SOA, administration revelation is the way toward finding web specialist organizations, and recovering web administrations portrayals that have been recently distributed. An administration vault contains pertinent metadata about accessible and forthcoming administrations just as pointers to the comparing administration contract archives that can incorporate SLAs. Administration disclosure in distributed computing alludes to discover reasonable administrations based on accessible help portrayal.

In SOA, the agreement-based help provisioning has not been centered well, though distributed computing offers these highlights through Service Level Agreements (SLA). Thus, the segment centers towards SLA-based cloud administration provisioning. In a cloud climate, buyers are changing with their necessities, and cloud suppliers are offering distinctive assistance capacities dependent on their promoting techniques. In this manner, recognizing the arrangements for satisfying the client's assumptions is a significant issue and prompts the requirement for a legitimate understanding. An administration level arrangement is a proper understanding, which assists with distinguishing cloud customers' assumptions, explains duties, and encourages correspondence among suppliers and shoppers.

15.6.3.1 Issues Identified with Repair Relocation on Cloud

In offered cloud merchant, the working standards are propelled towards to the ordinary determination and provisioning of administrations beginning a solitary supplier. There is an occasion in the direction of carry out the cycles by playing out the interoperability among the clouds.

15.6.3.2 Issues Identified with Privacy

Notwithstanding, the security arose as a significant issue in distributed computing; it should be appropriately taken care of while planning the intermediary. Next to some degrees of functionalities, the UI or dealing rationale of the proposed representative ought to do whatever it takes not to separate the aim of the necessities while assembling, and post provisioning exercises as well [14].

15.7 Explicit Risk and Cost in Multi-cloud Environment

The dangers depend on examination of value angles that make multi-cloud conditions free from mists given by a solitary supplier. Free suppliers utilize restrictive interfaces and setups. Administrations are additionally exceptionally coordinated with lower-level administrations offered by a similar CSP. Instances of this might be absence of basic SLA, utilization of non-viable advances, absence of similarity in the interchanges convention, absence of shared systems to guarantee information consistency and quality, the presence of administrations which are not carefully same, and missing some significant functionalities.

15.7.1 Multi-cloud Climate

Since through multi-cloud climate, we indicate an alternate arrangement of cloud stockpiling administrations with an intention for progressively indicated to the client at the purpose point furthermore be naturally overseen with no control by the worker. These days, just as obviously a great deal of paid cloud stockpiling administrations, it is conceivable to discover several cloud stockpiling administrations which have free records.

These free records normally accompany a few restrictions, for example, the measure of capacity they give or as far as possible on documents clients can transfer. Accepting that the outsider cloud stockpiling administrations are “accessible” and “dependable” from all perspectives, to choose what the best cloud stockpiling administration arrangement is accessible isn’t simple. This is on the grounds that for this situation the nature of administration isn’t carefully identified with its amount.

15.8 Discussion

The assessment of safety in multi-cloud conditions be a decent method to determine the exhibition and safety measures by the use segments facilitated in numerous mists. The safety assessment can be accomplished during the introduction of structure in this object.

The methodology introduced in the system likewise means to encourage straightforwardness and safety mindfulness. Inside the climate from the very beginning to end the collaboration among various motors the stage particular capacities. Straightforwardness and security mindfulness are essential viewpoints in the security. The executives of multi-cloud conditions, because they will assist with guaranteeing that events inside the multi-cloud climate be able to resolved. With this, the security status of human being segments and the whole multi-cloud climate can be known.

The system motors agreeably achieve consistent assessment of security in numerous mists given by various CSPs paying little mind to their varieties. The functionalities of the various motors contain be deliberately characterized in addition to chosen in favor of the individual motivation behind safety assessment. Merchant lock-in is significant disadvantage in distributed computing climate, this can try not to by team up various suppliers.

A few factors are incorporated while teaming up the administrations since free suppliers will have various norms and setup and client expects the administrations from the cloud all the time (High accessibility). Giving security to the cloud is fundamental.

15.9 Conclusion

Cloud storage frameworks have become alluring focuses for digital lawbreakers because of the gigantic appropriation of distributed storage by ventures and the simplicity of directing these assaults. Cloud storage has been created from a solitary cloud to multi-cloud coordinated effort, which represents another test to the security of information. These assaults are conceivable because of a few security issues, for example, broadened assault surfaces and absence of owing determination by cloud clients. We consequently propose a computerized security danger discovery and episode reaction framework as a strong counter-measure next to these safety issues. Giving security to the cloud is extremely basic. This chapter studied on quality perspective and difficulties in multi-cloud climate, for example, interoperability, security, and simplicity of relocation which is a lot vital while working together in multi-cloud.

References

1. Ferrer, B. R., Lastra, J. L. M., & Afolaranmi, S. O. (2018). A framework for evaluating security in multi-cloud environments. IEEE international conference (pp. 3059–3066). https://doi.org/10.1109/IECON.2018.8591454.
2. Martin, A., Akinrolabu, O., New, S. (2019) Assessing the security risks of multicloud SaaS applications: A real-world case study.6th IEEE international conference on cyber security and cloud computing (pp. 81–88). https://doi.org/10.1109/CSCloud/EdgeCom.2019.00-14.
3. Shi, T., Ma, H., & Chen, G. (2019) A genetic-based approach to location-aware cloud service brokering in multi-cloud environment. IEEE international conference on services computing (SCC) (pp. 146–153). https://doi.org/10.1109/SCC.2019.00034.
4. Thippeswamy, K., & Swetha, D. (2014). Quality aspects and challenges in collaboration of multi-cloud - A study. *International Journal of Engineering Research & Technology (IJERT), 3*(7).
5. Sahbudin, M. A. B., Scarpa, M., & Di Pietro, R. (2019) A web client secure storage approach in multi-cloud environment. IEEE international conference on computing, communications and security (ICCCS). https://doi.org/10.1109/CCCS.2019.8888062.
6. Meinel, C., Cheng, F., Torkura, K. A., & Sukmana, M. I. H. SlingShot - Automated threat detection and incident response in multi cloud storage systems. IEEE 18th international symposium on network computing and applications (NCA). https://doi.org/10.1109/NCA.2019.8935040.
7. Sharma, K. P., & Gupta, S. (2020). A review on applying tier in multi cloud database (Mcdb) for security and service availability. International conference on computer science, engineering and applications (ICCSEA). https://doi.org/10.1109/ICCSEA49143.2020.9132931.
8. Viswanath, G., & Venkata Krishna, P. (2020). Hybrid encryption framework for securing big data storage in multi-cloud environment. Springer-Verlag GmbH Germany, Part of Springer Nature. https://doi.org/10.1007/s12065-020-00404-w.
9. Choi, J., Cho, S., & Shin, Y. (2018). Study on information security sharing system among the industrial IoT service and product provider. IEEE international conference on information networking (ICOIN). https://doi.org/10.1109/ICOIN.2018.8343179.
10. Chen, G., Huang, G.,Shi, J., Kong, T., & Zhou, S. (2020). Research on multi-authority CP-ABE access control model in multicloud. Emerging technologies & applications, China communications (pp. 220–223). https://doi.org/10.23919/JCC.2020.08.018.
11. Wu, A. L., Zha, P., Hong, Z., & Sun, H. (2019) Research on Multicloud access control policy integration framework. China communications (pp. 222–234). https://doi.org/10.23919/JCC.2019.09.017.
12. Li, J., Yan, H., & Zhang, Y. (2019) Efficient identity-based provable multi-copy data possession in multi-cloud storage. IEEE transactions on cloud computing. https://doi.org/10.1109/TCC.2019.2929045.
13. Di Stefano, A., Di Modica, G., Tomarchio, O., & Morana, G. (2019) On the cost of the management of user applications in a multicloud environment. 7th international conference on future internet of things and cloud (pp. 175–181). https://doi.org/10.1109/FiCloud.2019.00032.
14. Parfenov, D. I., & Bolodurina, I. P. (2018) Development and research of the autonomous system for providing security and quality of service for multi-cloud platform.7th mediterranean conference on embedded computing. https://doi.org/10.1109/MECO.2018.8406038

Chapter 16
Trust Management Framework for Handling Security Issues in Multi-cloud Environment

S. Prithi, D. Sumathi, T. Poongodi, and P. Suresh

16.1 Introduction to Access Control Methods

Cloud computing technology facilitates users for storing, processing, and managing data on a remote server instead of maintaining it in a local server or personal computer. Users can avail the resources on-demand in the form of services. According to NIST, cloud computing is defined as a paradigm which enables pervasive, convenient, on-demand access of computing resources with minimum service provider interaction or management effort. The service delivery models are mainly categorized into Infrastructure as a Service (IaaS), Software as a service (SaaS), and Platform as a Service (PaaS). Although there are various categories of services available from numerous cloud vendors, there exists a gap among the stakeholders which leads to deliver inappropriate services to the user. A cloud broker could be used to enhance the functionalities of the cloud services. And, there are four different deployment models available in the cloud environment, namely, private cloud, public cloud, hybrid cloud, and community cloud. Security is a major concern nowadays because the number of users is increasing exponentially. Various techniques

S. Prithi (✉)
Department of Computer Science and Engineering, Rajalakshmi Engineering College, Chennai, India
e-mail: prithi.s@rajalakshmi.edu.in

D. Sumathi
SCOPE, VIT-AP University, Amaravati, India

T. Poongodi
School of Computing Science and Engineering, Galgotias University, Greater Noida, India

P. Suresh
School of Mechanical Engineering, Galgotias University, Greater Noida, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_16

are available to provide security including data integrity techniques, access control techniques, encryption techniques, etc.

16.1.1 Basic Access Control Models

Access control mechanisms provide security from various threats such as data loss, insecure API (Application Programming Interfaces), misuse of cloud services, DoS, malicious attacker, etc. The traditional approaches fail in monitoring the operation, i.e., user behavior, but the access is guaranteed in the cloud environment [1]. Access control mechanism helps in restricting the user access and monitors the logging status in the cloud environment. Access control systems are constructed using models, algorithms, and administrative capabilities. Moreover, every system is designed with its own attributes, methods, and capabilities used to restrict the user accessibility. This model supports to secure data and computation process occurring in the cloud environment. Access control decides the user privilege for accessing specific set of resources. Access control in cloud performs various actions such as identification and authentication before granting resources. There are five primitive access control mechanisms that can be applied in the cloud environment that include DAC (Discretionary Access Control), MAC (Mandatory Access Control), ABAC (Attribute-Based Access Control), RBAC (Role-Based Access Control), and CAAC (Context-Aware Access Control). An access control mechanism is designed using fine-grained CAAC framework for MongoDB which enhances the data protection of NoSQL data store [2]. Several advanced CAAC models are introduced that can be employed in cloud computing and IoT environment to manage the sensitive information [3].

16.1.2 Trust-Based Access Control Models

Cloud technology is preferably used in several Internet applications by exploiting different services that include SaaS, PaaS, and IaaS. Security is a major concern since a massive number of users are using cloud and the probability for malicious activities is high. Trust-based access model functions based on the behavioral parameters of users. The trusted value of a user or cloud service provider is evaluated using different parameters in every trust model. Some of the trust-based access models are discussed below:

Agreement-based access model: The trusted value is formed based on the agreements that are exploited by the providers to deliver different services to the users. In agreement-based approach, the trust model is implemented to prepare and exchange the agreement based on user specifications. The requirements of users from the trust module are security, Quality of Service (QoS), etc. However, the agreement could be in the form of practice statement (SLA-Service Level Agreement) or service [4].

Certificate-based access model: In this trust model, the trust value is computed based on Trust Tickets (TTs), endorsement keys, and certificates obtained from the concerned certificate authority. With the security certificates procured for platform, infrastructure, and software services, the trust value can be evaluated. TTs ensure the integrity and confidentiality of data that are handled in the cloud computing environment [5].

Feedback-based access model: The user's feedback and opinions are considered for calculating the trust value; it can be decided either to trust the service provider or not [6]. The feedback can be gathered about QoS and various security parameters. Hence, the cloud users can identify the trust value of a particular service provider by transmitting the request to a desired provider.

Domain-based access model: The trust value is computed by partitioning the cloud into intra-domain and inter-domain trust relationship. In intra-domain trust relationship, the trust values are computed based on the transaction occurred among entities. The trust value of any entity can be identified either using direct trust table or recommended trust value obtained from other entities [7].

Subjective access model: The trust value is calculated by categorizing the trust into subgroups such as trust based on execution, code, and authority. The probabilistic theory with assigned weights can be used to compute the trust value [8]. The final value will be taken by considering the combination of value that is computed in each subgroup.

16.2 SLA and Trust Management

An agreement that exists between the Cloud Service Providers (CSP) and the cloud users in the form of certain purposes and goals is considered as SLA. In this agreement, the terms and conditions would have been framed on which the services would be provided. Through this, transparency could be achieved in terms of deployment of services, providing security, resolving issues, and other services. From various inferences, it has been observed that SLA could be categorized into five types, and they are based on an entity like customer and attributes such as service.

- Customer-based SLA: This agreement works for the usage of services employed by the users.
- Customer level SLA: It covers the issues associated with the group of specific users.
- Multilevel SLA: This is framed to focus on the circumstances of several customers for the same service. It covers SLA of diversified levels.
- Service-based SLA: An agreement that exists between the provider and the user based on the service.
- Service-level SLA: Issues that are pertained to a particular service could be focused in this SLA.

16.2.1 SLA in a Multi-Cloud Environment

When it comes to a multi-cloud environment, then the complexity of maintaining SLA is a great challenging task. Traditional deployments of cloud computing includes various features such as operating system, network connectivity, performance, security, and business continuity administration. The main objective of the cloud user is to get the service immediately based on the request. The service provider's computation includes the failures during the availability of the services that arise during the provision of virtual machines, networking, storage, and other services. The end user requires to have a complete SLA that covers all infrastructure availability between them and the service provider. Running an application in a multi-cloud environment obscures the environment, and the cloud SLAs were not that complex enough to handle it since there are no common protocols to be followed in framing the SLA. The choice of framing the SLA is based on the selection of parameters, limitations, and exemptions that are associated with the availability of the service and standards.

Nowadays, the cloud service provider deploys the services on multi-cloud models like cloud service brokerage, hybrid cloud, inter-cloud, coalesced cloud, distributed cloud, and multi-cloud management framework [9].

16.2.2 Deployment of SLA in Multi-cloud Models

Traditional SLA exists between the CSP and consumer. Usually, the control would be with the CSP to ensure that the SLA functions properly for the consumers who have availed of the service. The same relationship could be applied to inter-cloud. Inter-cloud refers to the availability of more than two clouds. Therefore, the association of SLA could be established between the several CSPs as similar to the SLA in a single cloud.

16.2.2.1 SLA in the Coalesced Cloud

This works in two cases like peer-to-peer interaction and centralized way. In P2P interaction, irrespective of the existence of several CSPs, forming a group must agree to certain conditions related to the outsourcing. There might be "n" number of CSPs. All must agree on SLA to participate in this group. To provide a service, an intermediator has to interact with the cloud user. Hence, there is a need to maintain an SLA between the intermediator and cloud user. A centralized cloud federation employs a representative or centralized controller who takes the responsibility of interacting with the cloud user. In this case, the representative collaborated with all CSPs must have SLA. Service level in SLA among the CSPs and the representative

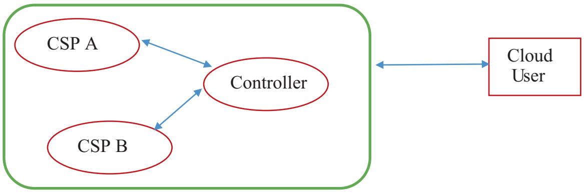


Fig. 16.1 SLA in the distributed cloud

are found to be public in the group and it is equivalent. If a user is in demand for a service from the coalesced cloud, then there is a need for the establishment of the SLA between the user and the coalesced cloud.

16.2.2.2 SLA in the Distributed Cloud

Distributed cloud could be categorized based on the two configurations as per the possession of multiple clouds. The distributed cloud model comprises several CSPs that are operated by a single CSP as shown in Fig. 16.1. In this case, an SLA has to be maintained among the providers as in the centralized cloud coalesced model. The distributed cloud differs from the centralized coalesced cloud in its nature as the distributed cloud model emphasizes the physically disseminated resources.

16.2.2.3 SLA in the Hybrid Cloud

Hybrid cloud refers to the amalgamation of both public and private cloud. When the private CSP needs service, then a request has to be made to the public CSP. In this context, an SLA has to function among the private and public CSP. Additionally, private cloud users of private cloud could be served based on their needs. To do the proper functioning, an implicit formation of an SLA would be done.

16.2.2.4 SLA in Multi-cloud Management Platform (M-CMP)

Cloud resources could be collected from various providers of public clouds, or it could be on cloud hosted by the admin in the multi-cloud management. Hosted cloud and the multi-cloud organization are managed by the admin employed by the multi-cloud management. Therefore, there is no necessity of an SLA among them. When the multi-cloud management admin takes the responsibility of managing the resources of the public CSP, then SLA is needed. In this scenario, there is no direct association among the cloud consumers and the multi-cloud management. However, when the M-CMP behaves as the CSP in terms of providing services on the top of already existing stack, then there is a need of the establishment of SLA among them.

16.2.2.5 SLA in Cloud Service Brokerage (CSB)

CSB takes the responsibility of managing the services among the cloud consumers and several providers. Initially, CSB must be aware of the services owned by the CSP. At this juncture, an SLA is needed to be maintained between the CSB and the CSP. At the next level, when the services are provided on demand to the cloud users, then an SLA must be established between the CSB and CSP. SLA (CSB, CSP_x) could be equivalent with the SLA (CSB, CU_y). Here CSP_x denotes any service provider and CU_y represents any cloud consumer. When looking into the services provided by the several providers, trustworthiness is a great factor that has attracted many researchers and academicians.

16.2.3 Trust Management Methodology

The trustworthiness of the services provided by the provider might vary since they adopt diversified network protocols and mechanisms [10]. Determining trustworthiness of multi-cloud service provider is a challenging issue, and it varies from single cloud. To implement multi-cloud in a successful way, a strong trust association is needed between the entities such as Cloud Providers, CSP, and Cloud Service Users. Here, the trustworthiness is examined, retained, and modified and it is disseminated. Computation of trustworthiness is done in terms of objective and subjective. Objective trustworthiness is linked with security, privacy protection, and Quality of Service (QoS). In [11], the trustworthiness is measured based on the QoS metrics that are retrieved from an SLA. Certain features like priority, needs, and profile of a CSP contribute to the computation of subjective trustworthiness. The next section describes the trust management framework that computes the subjective and objective trust of a CSP. The designed frameworks depend on the Trust Service Provider (TSP), spread over the clouds that could generate proof for trust from diversified sources. Basically, it gathers the knowledge about the amount of fidelity that CSP possesses to the SLA, and in addition, it also collects the feedback provided by the various users. Suppose, when there is a need for the CSP to believe in another CSP for providing the service in a combined manner, then in that case the selection of CSP is based on the services they provide. Hence, certain suggestions or recommendations or reviews could be provided with custom-made needs based on the services provided by the CSP. To offer services, TSP must have agreements with CSPs for examining their services. There must be cooperation between the TSP and CSP through which a reputation could be gained and lead to better recommendations. Building the trust could be done in two layers in the form of the construction of subjective trust model and objective trust model. The objective of the both models is to determine the trustworthiness, untrustworthiness, and uncertainty values of an entity. In [12], trust computations are based on the interaction of one consumer locally and also based on the communications done by all consumers globally.

Therefore, local trust and global trust is computed. Three scalars such as $\theta = (\{‘S’\}, \{’–S^1’\}, \{‘S–‘S’\})$ are used to represent the trustworthiness, untrustworthiness, and uncertainty. The objective trust comprises local and global objective trust. Similarly, the subjective trust comprises the computation of both local and global subjective trust. The significance of SLA is seen in evaluating the objective trustworthiness of services such as transparency, reliability, availability, customer support, security, and privacy. All these attributes contribute toward the evaluation when it is functioning based on the SLA between the CSP and cloud user.

Evaluation of local objective trust:

To find out, three factors like satisfied (S), not satisfied (S^1), and uncertain (U) are used. Considering a session between the provider and the consumer, observations are done to record the communication between them, and it is stored in "R." Assume $\alpha_{i,j,k}$ (t) denotes the number of satisfied communications among the provider and consumer for each of "k" services. Let $\beta_{i,j,k}$ (t) refers to the number of unsatisfied communications that are obtained, and $\gamma_{i,j,k}$ (t) denotes the communications that are found to be uncertain. The metrics are used to evaluate for a complete session between the provider and the consumer.

$\alpha_{i,j,k}(t) = \beta_{i,j,k}(t) = \gamma_{i,j,k}(t) = 0$ denotes that there is no interaction.

The local objective trust ($L.T_{i,j,k}$) is computed by summing the probabilities of all events that occurred during the interactions.

$$L.T_{i,j,k} = \left\{ \text{lom}_{i,j,k}(\alpha),\ \text{lom}_{i,j,k}(\beta),\ \text{lom}_{i,j,k}(\gamma) \right\}$$

The global objective trust is computed by aggregating the trust values from the provider. For example, to compute the global objective trust ($G_{j)}$ = {gom$_j$ (S), gom$_j$ (S^1), gom$_j$ (U)}. This is computed for all users who interact with the service providers for obtaining a service.

Computation of subjective trust based on the feedback: A threshold value is fixed for computing the local subjective trust. Feedbacks are obtained from the consumers and based on their rating; it is computed.

$$S.T = \left\{ S.T_{i,j}(S), S.T_{i,j}(S^1), S.T_{i,j}(U) \right\}$$

This is used to compute the local subjective trust.

Global subjective trust: It is computed by evaluating trusts computed by the users for a specific provider.

$G.S.T_j$ = {gsm$_i$ (S), gsm$_i$ (S^1), gsm$_i$ (U)} denotes the summation of the global subjective probabilities for trustworthiness, untrustworthiness, and uncertainty. It is computed for all users' feedback.

To handle the multi-cloud applications, security SLA is taken into consideration since multi-cloud is more prone to security issues [13, 14].

16.2.4 Security SLA in Multi-cloud

Designing of security SLA is based on the deployment of components and relationships among the components and resources that are in usage. During the functioning of the multi-cloud, a detailed evaluation of diversified monitoring tools [15] associated with the security for collecting the measurements of particular parameters that are related with the group of security Service Level Objectives (SLOs) is done. All these metrics have been mentioned in the Security SLA as mentioned in SPECS [16, 17]. This section describes about a framework that has been intended to improve multi-cloud applications with the help of security-by-design methodology. This is made possible by considering the account security problems at the initial phase. The MUSA framework [18] worked with the help of group of components that execute individually. Components communicate while the deployment of multi-cloud application is executed. Components thus deployed make use of SaaS (Software as a Service) cloud services and/or are hosted by IaaS (Infrastructure as a Service) cloud services. Service Level Objectives (SLOs) would be in the security SLA. SLO refers to the contributions that could be assessable by the customers. The MUSA development framework is intended to aid in terms of security SLAs that are pertained to each component that exists in the multi-cloud. All those Security SLAs are then stored in a location for performing the verification of the SLOs that are related to the multi-cloud application.

Security SLA: To denote security in SLA, certain SLA format has been suggested as per the WS-agreement standard. A few concepts oriented with security have been embedded in SLA. They are as follows:

Security metrics: These are used to evaluate the security levels of services that are offered by the providers.

Security capabilities: Security mechanism that has been implemented on the target service.

SLOs: This refers to the constraints that are communicated based on the security metrics. This denotes the phases of security that must be valued based on the SLA.

The MUSA framework performs the tasks as described below:

1. Application Modelling: This is the first phase as it is used to specify the Cloud Provider Independent Model (CPIM), an activity aided by the Modeller.
2. Continuous Risk Assessment: The choice of security controls and metrics is done through this activity. This action is specified in the Security SLA and managed at the execution time.

   It supports in choosing security controls and metrics which are granted and controlled in the Security SLA at execution time.
3. Choosing the cloud services: A tool, namely, MUSA Decision Support Tool (DST), is used to identify the proper cloud service that suits with the corresponding security needs of the application components.
4. SLA template construction: An automatic creation of the security SLA template is done.

5. Planning for the implementation: All the templates are stored in the repository for further process. These Security SLAs are extracted and used during the construction of deploying the multi-cloud.
6. Collected Security SLA generation: The Security SLAs thus stored are combined and collected for the complete application.
7. Final execution: The plan that has been already devised is executed. During the execution, it checks for the facility, configuration of resources, its deployment, and as well as the MUSA agents that are needed in the plan.
8. Frequent monitoring: As soon as the deployment is completed, the metrics on the multi-cloud are regularly monitored. Through this regular inspection, there might be a chance of raising an alarm during any mishaps.
9. Implementation of security controls: Responsive measures are raised in effect to the mishaps, and moreover, the dynamic administration of MUSA security application agents is also supported through this activity.

Throughout this framework, it has been designed in such a way that activities such as application modelling, frequently assessing the risk, and selecting the cloud service perform in the iteration basis. It would be easy to check the security needs in the corresponding application that are not probably addressed with the security controls provided by the services.

16.3 Security Framework for Multi-cloud

In the early days of the public cloud, businesses used one cloud to meet all of their digital infrastructure needs. Today, many companies rely on multiple cloud strategies. These methods use multiple cloud service providers to provide the society with a variety of options and capabilities to choose from. Another major consideration is the need to protect various cloud providers, in other words. The right security framework will protect your business and allow it to maximize the total value of a multi-cloud environment. Multiple cloud security is a comprehensive cloud security solution that protects and blocks business and customer data, goods, and applications from advanced security threats and cyber-attacks on multi-cloud infrastructure and environments.

16.3.1 Shift from Single Cloud to Multi-cloud

The cloud computing usage has been quietly increased in several enterprises. Cloud computing affords a lot of advantages in terms of low cost and data access. The key factor in cloud computing environment is considered in the perspective of security, because users frequently cache sensible information from cloud storage providers; however the providers may be unreliable. Handling with "single cloud" providers are expected to be less attractive to customers due to the risk of failure to access services and the potential for intruders within a single cloud. Bowers et al. [19] have shown that more than 80% of company executives "fear security threats and loss of control over data and systems." A step towards "multi-clouds," or "inter-clouds" or "cloud-of-clouds," came into existence in recent years. The assumptions given by Vukolic [20] to move into multi-cloud are to strengthen the features of single cloud by spreading trust, reliability, and security among many cloud providers.

According to the predictions of Gartner, by 2021, around 75% of medium-sized enterprises and large organizations would adopt multi-cloud or hybrid cloud strategy. In another survey of public cloud users done by Gartner, 81% of interviewees use two or more providers. The reason why most organizations establish a multi-cloud strategy is to evade vendor lock-in or to benefit from best-of-breed solutions. Gartner further predicts that by 2023 half of the public cloud users will be governed by 10 biggest public cloud providers.

16.3.1.1 Single Cloud: Security Limitations

The security issues related to data storage in the cloud are still the first obstacle to acquire cloud computing from state-owned companies and organizations. Moreover, security concerns cannot be avoided on a cloud computing to achieve the required level of maturity, for the forthcoming generation of IT. It is the main function of cloud services to guarantee data integrity and to deliver privacy for the storage of data in the cloud; nevertheless the providers don't have control over the stored data in the data centers. In this section, about few limitations of security of single cloud are discussed.

Integrity of Data

Data integrity is a significant issue related to cloud security issues. Data transferred among the client and cloud providers may be corrupted or lost. For an instance, in October 2009, the Sidekick user data such as contact references, calendars, and photos were lost due to malfunctioning of server in Danger (Microsoft) database; within one-year Microsoft has been perceived that most of the missing information cannot be found [21]. As another example, in January 2009, Magnolia servers suffered entirely from data loss due to complete failure; the loss of half a terabyte of data has made the recovery process impossible, ended up the site dead [22].

Confidentiality and Privacy of Data

Protecting sensitive data such as banking details or documents related to health care should be among the most important things for cloud providers such as either internal (malicious administrators working for Cloud Service) or external attacks.

Various cloud providers adopt a variety of technologies to solve data privacy issues, but a customized cloud environment makes traditional methods unstable in dealing with privacy risks, and the exploitation of different encryption methods is still limited. Garfinkel [23] provides an example of privacy loss such as an Amazon Cloud service, and it [24] shows that by simply identifying the Amazon account password, access to the total account settings and resources is possible.

Data Availability

Cloud computing is a ubiquitous technology that motivates customers to migrate to cloud service due to high availability of services, data, and applications. If data is placed on a single cloud provider and not considering a solution for storing data in the same platform or location, this may increase the risk of downtime and affect customers who may be trapped for several hours without any access to the data. Amazon [25] insists in its agreement that the service can be reduced at any time.

16.3.1.2 Single Cloud Homomorphic Encryption

Cloud providers use conventional methods to protect the customer's data by encrypting the data before sending it to cloud. So whenever a client needs to process its data, the cloud provider wants a decryption key. Indeed, it is not possible to accomplish any processing of encrypted data without decryption. As a result, the cloud provider handles all the specific details.

Homomorphic encryption systems are utilized to accomplish operations on encrypted data without using a secret key, i.e., without encryption, and the client is the unique owner of the secret key [25]. Encryption is a function, if: from Enc (a) and Enc (b) it is possible to calculate Enc ($f\,(a, b)$), where f can be +, $\times$, $\bigoplus$ without using the private key. The homomorphic encryption can be grouped into two types, namely, additive homomorphic encryption and multiplicative homomorphic encryption. Pailler and Goldwasser-Micalli cryptosystem are known as additive homomorphic encryption where addition is applied on raw data. RSA and E1 Gamal cryptosystems are the multiplicative homomorphic encryption where products are applied on raw data. Some of the limitations of homomorphic encryption system are slow in moving, large encrypted text, limited bandwidth, and the size of the ciphertext depends on the ISP. Because it has provided its IT management to the Cloud Provider, the secret key used to encrypt the data held by the cloud provider is lost; only the encrypted data can be retrieved, so it cannot be used again.

16.3.1.3 Security Mechanism in Multi-cloud

With the limitations discussed earlier with single cloud security issues, cloud computing should not be limited to a single cloud as well as the sensitive data should not be assigned to a single cloud and should ignore relying on single cloud provider. So, cloud computing has moved from single cloud to multi-clouds, inter-clouds, or cloud of clouds that has the responsibility to achieve the data security.

A reliable shared storage using a Byzantine Fault Tolerance (BFT) strategic set was proposed by Vukolic for use on multi-clouds. Many recent studies in this particular domain have created inter-cloud protocols. RACS (Redundant Array of Cloud Storage), for example, uses RAID-like techniques that are widely used for disks and file systems, but for multi-cloud storage. Abu-Libdeh et al. [21] consider that to prevent "vender lock-in," distributing user data between multiple clouds is a useful solution. Moreover, the repetition reduces the cost of switching providers and provides better tolerance for errors. Hence, storage load will be distributed among multiple providers thanks to the RACS representative. HAIL (High Availability and Integrity Layer) is a protocol that controls and manages multiple clouds. HAIL is a distributed cryptographic system that robustly ensures the client's stored data is integral and retrievable. HAIL offers a software layer which addresses availability and integrity of the stored data in an inter-cloud.

Bessani et al. [26] proposed DepSky a virtual storage system for the integration of several clouds to construct a cloud of clouds. And, DepSky enhances the availability, integrity, and confidentiality of information that are maintained in cloud through encryption, encoding, and data retrieval. This allows to reduce the restrictions of individual clouds using a number of reliability and security measures. The DepSky system confirms the availability and confidentiality of data stored on various cloud providers through multiple clouds architecture and integration of the byzantine algorithm by fault tolerance, private sharing, and eradication cryptographic codes. DepSky architecture comprises four clouds, and each cloud maintains its own interface.

16.3.2 Cyber Security Framework

The Cyber Security Framework (CSF) contains regulations, policies, and best practices for managing cyber security-related risks. Irrespective of the nature of the organization or its purpose, functions, enforcement measures, obligations, and purposes related with ensuring a strong security stand can be made and discussed using CSF. This empowers security management more convenient and enables the sharing of better information. This framework has a strategic, malleable, and affordable approach for promoting the protection and flexibility of critical infrastructure and other important sectors of the economy and national security.

16.3.2.1 Cyber Security Challenges: Multi-cloud Environment

Organizations face high-risk or critical business plans that are held in a data center and transfer the low risk to the cloud to become enlightened with the cloud. With all the requirements for compliance and safety and control for all cloud providers, it has led to many other security concerns. The following challenges that have to be taken care of when designing cloud or hybrid environmental security are depicted in Fig. 16.2.

1. **Lack of visibility in security posture:** As the load (s) of your work (s) are distributed across the cloud, the security status as an organization will not be reflected. Security will depend on CSP tools and technologies that can be exploited and integrated with it.
2. **Agreement on organization policies and regulatory requirements**: It becomes more and more tough as the number of players in the security area increases. Translation of organization's security policies into multiple and diverse retailer technical areas can be an overwhelming assignment for security professionals. Compliance testing in accordance with legal requirements will be another area where the evaluation and visibility of reports depends on the CSP.
3. **Shadow:** The significant advantages of cloud services are the rapid deployment of virtual machines, storage, and applications. This advantage is also the reason

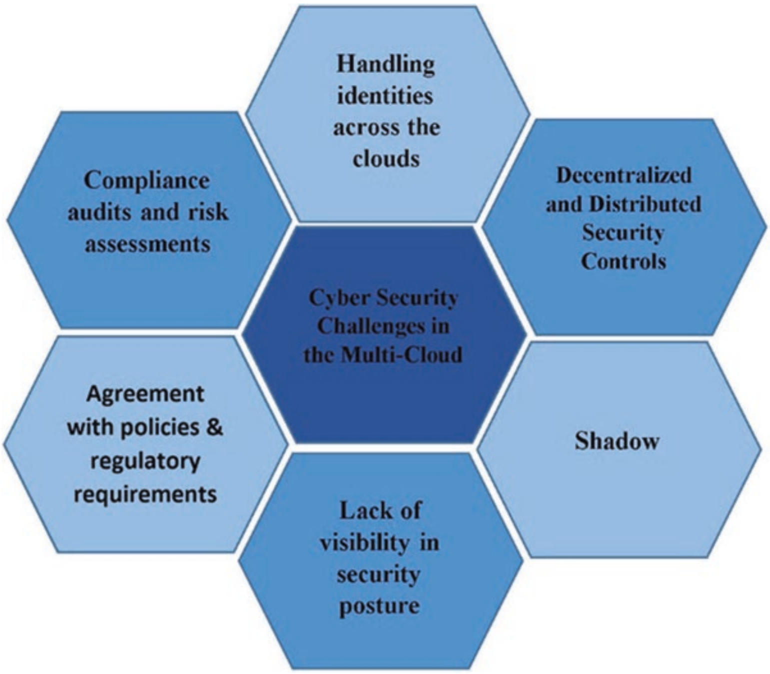


Fig. 16.2 Cyber security challenges in multi-cloud

for the rapid growth of IT shadow. To be more productive, business units and individuals can make their own purchasing decisions without IT involvement, which has led to the existence of data assets outside of IT control. This creates the problem of protecting data assets, applying policies and procedures to maintain the security status of organizations.

4. **Decentralized information and Distributed Security Controls**: For every CSP there are own security tools/technologies available for efficient and effective use of their platform. This turns out a challenge for security groups as combining all these tools and gaining a balanced view of risk, implementation of policy, and threat to security becomes a challenge. This also contributes to the efficiency and effectiveness of security professionals as they are unable to perform multiple tasks and require to obtain the ability to observe and accomplish different tools for the same task. Security technologies deployed in many cloud environments required to allocate threat intelligence to stop the onset of murderous chains, as security logs will be produced in many cloud locations in a variety of ways. To find these logs in the central point of describing the merger, to perform the analysis is not possible but difficult if the holding structures of the CSPs are geographically different.
5. **Handling identities among clouds**: The challenge includes timely management and security of provision and reduction of users. The challenge could be an in-house IDAM solution that cannot be expanded or needs to be significantly improved to provide a federated IAM. Security experts often have the challenge of empowering users to navigate around the environment while maintaining appropriate safety standards – or more importantly, ensuring a consistent and secure management policy between multiple cloud services and pre-built applications.
6. **Compliance audits and risk assessments**: From a business perspective "Safety" is about risk management. In the event of a cloudy environment, the integration of tools such as Identity Access Management and encryption could present additional risks. Managing and maintaining inventory details and defining business priorities becomes more complicated, and this can interfere with risk management. Moreover, the GRC team needs to spend a lot of time and effort in gaining an integrated view of the risks throughout the distributed area.

16.3.2.2 Ways to Handle Challenges in Multi-cloud Environment

The cloud security architecture is designed by considering to protect users, maintain data, and better communicate with compliance requirements. As well as consistent security should be there for all the components in a multi-cloud environment. Other cloud security considerations such as data encryption, secure communication links and handling encryption keys and certificates are the entities that have to be taken into consideration.

1. **Legitimate Involvement**: To start any cloud migration project, it is necessary to install a legal team to verify CSP contracts. In the event of a multi-cloud environment, the result of comparisons among several CSP contracts can be used to determine which workload can be held in place. These contracts make way to an important input into how the event would be handled and how data breaches in a multi-agreement environment. What could be shared, when, and how required incident management data and pre-investigation would be provided.
2. **Visibility:** The most important aspect is being able to access and visualize the resources and services for cloud service providers. However, security professionals need a single dashboard framework that could provide them an understanding of what resources are being used and how services are related. To avoid the problem of shadow IT, it needs to create business rules; these rules can define which users can provide services and what types of resources can be provided. Role-Based Access Control (RBAC) is critical to provide the right level of access to the right users to perform their tasks and at the same time maintain the need to know the principle of access provision. Determining the appearance of an attack is not targeted at the location, and how that will be dealt with is another important point that needs to be considered, e.g., DDoS attacks or viruses on shared infrastructure.
3. **Availability of Data**: These days most cloud service providers offer carriers so you can transfer workloads to another cloud service provider. Though it is very easy to transfer virtual machines, when it comes to transferring data sets, it is not so easy so make sure that data backup is automatic and data recovery is checked so that in the event of a disaster or if there is a need to switch to another service provider, information will be available.
4. **Centralized Monitoring, Management, and Orchestration**: Obtaining a single dashboard view with a safety profile is essential. Safety logs from all areas should be collected and processed in one place so that the same process can be used to analyze and give priority to events. This improvises the efficiency of the sacred security professionals. Safety tools and technologies used in the right place will be able to manage from the centralized management console. This will help to manage the same rules and policies that apply to tools in all areas. All security items including tools, technology, policies, and configurations should be handled and handled in the middle. This will ensure that even though different CSPs may have different vendors, you will still be able to have consistent safety and security policies.
5. **Identity and Access Management**: Security architect should not create different identity silos by offering cloud deployment. And, each cloud deployment may have different accounts within each provider and application. To solve this problem, organization should look at a private Identity and Access Management (IDAM) service that has the ability to manage the access of all users and authenticate before they can access any applications in the multi-cloud environment. This may include using a trade union guide and applying an integrated solution (SAML-based) to customers and other third-party users such as partners. This

will help the organization to easily and securely access data and applications installed on multiple CSPs with a single login.

16.3.3 Security Controls in Multi-cloud

Cloud security is a completely new ballgame, with a completely new risk. In order to secure the data, and relevant information to the new environment, there are two steps to be taken. The first step is to ensure that promising solutions in the cloud environment have appropriate security capabilities, and next to ensure that the cloud solutions are companionable with security deployed on a distributed network. The truth is, mishandling the cloud security controls can leave the type of space cyber criminals are waiting for. In reality, Gartner predicts that 99% of all cloud security incidents will be because of the device misconfigurations done by customers. The five indispensable features to address when deploying cloud security controls are discussed below.

1. **Centralized visibility of the cloud infrastructure:** The essentials of cloud security management include moderate visibility in security policies, configuration settings, and user activity, and in addition, potential risks hidden in online data stores. It highly reduces the security risk for cloud security due to poor maintenance, or by losing an amazing job that could indicate an attack. The main challenge falls on the various clouds who offer variations in configuration settings and the choice of these settings by developers who don't have knowledge about security. Getting visibility in all conditions and clouds is no easy task. To reduce these types of risks, security teams need moderate visibility in cloud infrastructure. Cloud workload protection tools (CWP) can be used which strongly integrates cloud management and security systems. These tools provide security team the facility to monitor and assess the configuration status of current services, along with the overall security posture of the cloud environment. Automated configuration monitoring allows IT teams to identify and quickly respond to security misconfigurations, thus bolstering security while shortening the time it takes to implement fixes.
2. **Cloud management and security system integration:** Cloud computing depends on a shared sharing model, where the customer and public cloud vendor control the security settings. Visibility in cloud-safe environment requires a close connection between CWP solution and the basic cloud environment. This means the integration of the API level with tools such as Amazon Inspector and VPC Flow logs and GuardDuty for AWS; Stack Event and Flow Drivers for Google Cloud Platform; and the Azure Safety Center. SaaS customers may also need a cloud access security broker (CASB) solution that integrates seamlessly with the SaaS service to identify risks and set up problems specific to SaaS deployment.
3. **Protection of web application layer in integration with machine learning and AI:** Cloud providers only take responsibility for infrastructure. To better

fulfill their role in the shared commitment model, the organizations requisite to strengthen the security of web applications with web application firewalls. Apps threat detection is different when applications run in the cloud rather than on the premises. Here the discovery of the threat requests to happen within the app's content, and not on the traffic. The only way to use the power of a computer with the speed of artificial intelligence is to protect modern cloud-based systems. Machine learning can help to identify the type of user behavior and application behavior that indicates a problem and may apply security measures in ways that no human-enabled method can match quickly or accurately.

4. **Security Automation:** Provided the gap in cyber security capabilities, the present condition of the cybersecurity sector is not enough to meet all the business needs of the twenty-first century. Cybersecurity professionals are in high demand, and DevOps and security teams have a gap in skills that end businesses at risk of various threats. Until the organization is able to meet the needs of the business and the demands of greater and more talented business, security designers are motivated to support organizations to carry out their security functions where possible. Another method currently in use includes plugins that bring forth administrators more visibility in multi-vendor ecosystems, allowing for automation and simplified management. When app changes occur, IT and DevOps teams retain without updating their security policies every time app attributes change.
5. **Threat Intelligence Feeds:** When the cloud environment becomes more complex – the result of a merchant spread happens when multiple providers with various security platforms are used and become more vulnerable to threat. Cloud security comes with a complete solution that puts under one umbrella where all the cloud service employs the organization. However, a good solution should include a powerful intelligence feed that incorporates in-depth expertise on global and domestic security issues. When selecting cloud security controls, look for suppliers whose solutions provide data gathered from the sensors.

16.4 Research Challenges

Maintaining an SLA is a complex task since many applications move into the cloud. The environment is dynamic since applications might possess their own needs and it is also not possible to plan in advance and accurately in delivering the services needed in each environment. Monitoring QoS and application-level monitoring is also trivial. There are more opportunities for the multi-cloud application to inherit the vulnerabilities from the place where it is combined. This is found to be a more common issue that has to be resolved in the architectural level. Maintaining a standard security methodology in the cross-environment is too complex, and it seems to be more complex when the provider is not able to assist you technically during the issue of the service.

16.4.1 Authentication

Companies and organizations must impose on multiple layers of authentication to each cloud service which stores sensitive data along with access control and identity management. Multi-factor authentication (MFA) tools have to be utilized which need secondary verification either through email, SMS, or key generator. This is a big research challenge for the companies, and furthermore risk-based solutions yield various risk factors such as position, behavior, or authorizations into account.

16.4.2 Compliance

There seem to be multi-cloud environments that interact with business data which becomes difficult to guarantee whether sensitive data is labelled and secured properly. There are multiple metropolises that maintain precise guidelines which concern security compliance and data privacy. It is the responsibility of the organization to ensure that all sensitive information is compliant in each and every place they organize business.

16.4.3 Vulnerability Management

There needs to be frequent scanning and testing to check vulnerability in the applications and networks. Besides the customary vulnerability scanners, the organization must exploit threat intelligence software to ascertain the security in cloud resources from the utmost progressive threats on the Internet.

16.5 Conclusion

This chapter provides an overview of the access control mechanisms that exist in the multi-cloud environment. Apart from these models, it also highlights the significance of SLA in multi-cloud. Later, the multi-cloud that has been formed through the interaction of service providers is prone to security issues during the service provisioning. Hence, this chapter also addresses the importance of trust that has to be integrated with SLA for providing an uninterruptable service to the consumers. The deployment and need of security in SLA have been described in the perception of a framework. Finally, the issues that could arise due to the deployment of multi-cloud have also been discussed.

References

1. Behera, P. K., & Khilar, P. M. (2017). A novel trust-based access control model for cloud environment. In *Proceedings of the international conference on signal, networks, computing, and systems* (pp. 285–295). Springer.
2. Colombo, P., & Ferrari, E. (2017). Enhancing Nosql datastores with fine-grained contextaware access control: A preliminary study on mongodb. *The International Journal of Cloud Computing, 6*, 292–305.
3. Kayes, A., Rahayu, W., Dillon, T., Chang, E., & Han, J. (2019). Context-aware access control with imprecise context characterization for cloud-based data resources. *Future Generation Computer Systems, 93*, 237–255.
4. Alhamad, M., Dillon, T., & Chang, E. (2010). Sla-based trust model for cloud computing. In 13th international conference on network-based information systems (pp. 321–324). IEEE.
5. Krautheim, F. J., Phatak, D. S., & Sherman, A. T. (2010). Introducing the trusted virtual environment module: A new mechanism for rooting trust in cloud computing. In *International conference on trust and trustworthy computing* (pp. 211–227). Springer.
6. Habib, S. M., Ries, S., & Muhlhauser, M. (2011). Towards a trust management system for cloud computing. In IEEE 10th international conference on trust, security and privacy in computing and communications (pp. 933–939). IEEE.
7. Jamshidi, P., Ahmad, A., & Pahl, C. (2013). Cloud migration research: A systematic review. *IEEE Transactions on Cloud Computing, 1*, 142–157.
8. Li, W., Ping, L., Qiu, Q., & Zhang, Q. (2012). Research on trust management strategies in cloud computing environment. *Journal of Computer Information Systems, 8*, 1757–1763.
9. Grozev, N., & Buyya, R. (2012). Inter-cloud architectures and application brokering: Taxonomy and survey. *Software Practice and Experience*. https://doi.org/10.1002/spe.2168
10. Abawajy (2011). Establishing trust in hybrid cloud computing environments. In Proceedings of 2011 IEEE 10th international conference on trust, security and 1041 privacy in computing and communications (TrustCom) (pp. 118–125).
11. Chakraborty, S., & Roy, K. (2012). An SLA-based framework for estimating trustworthiness of a cloud. In Proceedings of 2012 IEEE 11th international conference on trust, security and privacy in computing and communications (TrustCom) (pp. 937–942).
12. Fan, W., & Perros, H. (2014). A novel trust management framework for multi-cloud environments based on trust service providers. *Knowledge-Based Systems, 70*, 392–406. https://doi.org/10.1016/j.knosys.2014.07.018
13. Bohli, J. M., Gruschka, N., Jensen, M., Iacono, L. L., & Marnau, N. (2013). Security and privacy-enhancing multicloud architectures. *IEEE Transactions on Dependable and Secure Computing, 10*(4), 212–224.
14. Singhal, M., Chandrasekhar, S., Ge, T., Sandhu, R., Krishnan, R., Ahn, G., & Bertino, E. (2013). Collaboration in multicloud computing environments: Framework and security issues. IEEE Computer (pp. 76–84).
15. Casola, V., Benedictis, A. D., & Rak, M. (2015). Security monitoring in the cloud: An sla-based approach. 10th international conference on availability, reliability and security, ARES 2015, Toulouse, France (p. 7).
16. Rak, M., Suri, N., Luna, J., Petcu, D., Casola, V., & Villano, U. (2013). Security as a service using an sla-based approach via specs. IEEE 5th international conference on cloud computing technology and science (CloudCom), (Vol. 2, pp. 1–6).
17. De Benedictis, A., Rak, M., Turtur, M., & Villano, U. (2015). Rest-based SLA management for cloud applications. Enabling technologies: Infrastructure for collaborative enterprises (WETICE), 2015 IEEE 24th international conference (pp. 93–98).
18. Rios, E., Mallouli, W., Rak, M., Casola, V., & Ortiz, A. M. (2016). SLA-driven monitoring of multi-cloud application components using the MUSA framework. 2016 IEEE 36th international conference on distributed computing systems workshops (ICDCSW), Nara (pp. 55–60).

19. Bowers, K. D., Juels, A., & Oprea, A. (2009). HAIL: A high-availability and integrity layer for cloud storage. CCS'09: Proceeding of 16th ACM conference on computer and communications security (pp. 187–198).
20. Vukolic, M. (2010). The byzantine empire in the intercloud. *ACM SIGACT News, 41*, 105–111.
21. Abu-Libdeh, H., Princehouse, L., & Weatherspoon, H. (2010). RACS: A case for cloud storage diversity. SoCC'10: Proceeding of the 1st ACM symposium on cloud computing (pp. 229–240).
22. Sarno, D. (2009). Microsoft says lost sidekick data will be restored to users. Los Angeles Times.
23. Naone, E. (2009). Are we safeguarding social data? *Technology Review published by MIT Review*. http://www.technologyreview.com/blog/editors/22924/.
24. Garfinkel, S. L. (2003). Email-based identification and authentication: An alternative to PKI. *IEEE Security and Privacy, 1*(6), 20–26.
25. Tebaa, M., & El Hajji, S. (2013). Secure cloud computing through homomorphic encryption. *International Journal of Advancements in Computing Technology (IJACT), 5*, 16.
26. Bessani. (2011). DEPSKY: Dependable and secure storage in a cloud-of-clouds, EuroSys'11, Salzburg, Austria.

Part V
Intelligent Broker Design for Multi-cloud Environment

Chapter 17
Intelligent Workflow Adaptation in Cognitive Enterprise: Design and Techniques

Arunkumar Panneerselvam

17.1 Evolution of Cognitive Computing and Cognitive Enterprise

17.1.1 Introduction

Digital revolutions have changed the way of doing business, and the operations and processes of the enterprises adapt to the advent of the technologies. Artificial Intelligence-supported supply chain management systems, customer relationship management, and marketing management are transforming the business organization into cognitive enterprises. Cognitive enterprise adapts a set of emerging technologies in all its applications. The technologies include Artificial Intelligence, machine learning approaches, natural language processing, big data analytics, and much more. Cognitive skills are learnt and adapted by the systems and applications of the cognitive enterprise. When speaking about automation of business systems, it is inevitable to discuss about the workflows.

Workflows are automated set of tasks or business operations that happen in sequence to attain a business need. Workflows in general do not have the properties of learning, but they can be made to learn, adapt, and make decisions by incorporating the machine learning techniques, and such workflows termed as intelligent workflows are the key building block in building cognitive enterprise.

Intelligent workflows transform the business process to new heights, and they build value to the business. Bringing better customer experiences by personalization, reducing the operational time, and leveraging the enterprise-wide information

A. Panneerselvam (✉)
Department of Computer Applications, E.G.S Pillay Engineering College (Autonomous), Nagapattinam, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_17

and empowering end-to-end business operations are the key features of intelligent workflows.

As the business organizations handle more data from different distributed systems such as customer relationship management, enterprise resource planning, supply chain management, and many more, it requires suitable platforms such as cloud to process, store, and make analytics over the huge data. Cognitive enterprise applications are more cloud centric. Workflow management systems and applications collect data from different vendors, customers, and other business units which host the data in multi-cloud environment.

Designing of intelligent workflows, tools, and processes for intelligent workflow management is the key concern for cognitive enterprises. This chapter will give the understanding of the cognitive enterprise and its structure, intelligent workflows overview, components and building techniques, and challenges in implementing intelligent workflows in multi-cloud environment.

17.1.2 Cognitive Transformation

The business enterprises worldwide during the past decades are constantly undergoing reformation in its way of doing businesses due to advancements in the technologies. While the technological adaptations are inherited by the business organizations, it paves way for the incredible transformations in customer experiences and employee-centric activities. As a result, the business owners gain high returns and are able to lead their organization to greater heights.

The term “cognitive” relates to “cognition” which means gathering information and gaining knowledge with the help of thought, sense, and decisively the experiences. The cognitive action of the human brain can be imitated in machines such as computers for better decision-making process, and thus cognitive computing evolves.

Though there is no established standard definition for cognitive computing, the cognitive computing consortium which aims at developing cognitive computing community defines cognitive computing as “Cognitive computing makes a new class of problems computable. It addresses complex situations that are characterised by ambiguity and uncertainty; in other words it handles human kinds of problems” [1]. Building computing models that exactly represent the human thought process to solve complex problems that are ambiguous and uncertain in nature is the key of cognitive computing.

Convergence of technologies is inevitable in the growing intelligent era. Business processes are changing rapidly by adapting new technologies. Digital enterprises are now transforming as cognitive enterprises. The technologies such as Artificial Intelligence, machine learning, big data, and cloud work together to build a cognitive enterprise.

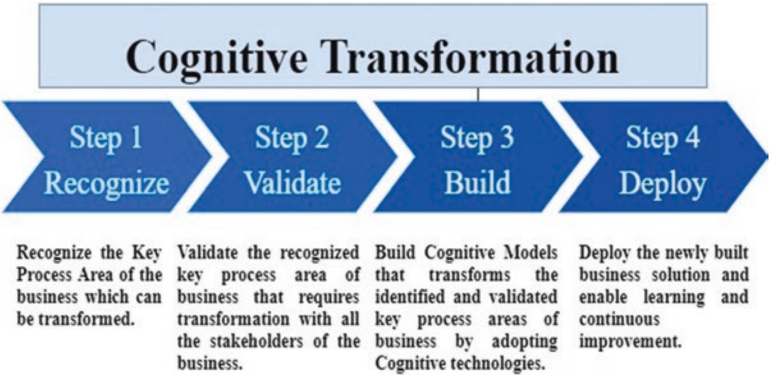


Fig. 17.1 Steps for transforming the existing business process to cognitive model

Cognitive enterprises can be termed as reinvented business enterprises with the adoption of cognitive computing. The adoption of cognitive technologies at all levels of business leads to the rise of cognitive enterprise. Technologies such as Artificial Intelligence (AI), machine learning, big data, natural language processing, Internet of Things (IoT), block chain, and cloud converge to form a cognitive enterprise. Cognitive business can evolutionarily scale by learning and adaption without much human interaction.

A business deals with lot of data and information that is gathered throughout all the business process at enterprise level. This gathered data is an asset to an organization from which valuable insights and decisions can be made that raise the business to higher levels. Cognitive technology applications in business change the business process to a smarter way with rich customer and employee experiences. The steps for transformation of digital business to cognitive business are depicted in Fig. 17.1.

17.1.3 Cognitive Skills Necessary for Cognitive Enterprise Applications

Enterprise cognitive computing applications refer to leveraging the cognitive computing capabilities into software systems that drive the business processes. According to the cognitive computing consortium the cognitive systems should contain the following properties [1]:

- Adaptive, which refers to the capability to learn continuously
- Interactive, the ability to interact with external systems and people
- Iterative and stateful, capability to get complete input from the user and remembering previous interactions
- Contextual, ability to recognize the contextual factors such as time, profiles, preferences, location, tasks, and processes

The three important aspects of cognitive applications are the cognitive modelling which provides best solutions for the unanticipated complex business problems by continuous learning and adaption. Second, decision support with better confidence score and third is the invention of unknown insights from data which cannot be even discovered by the human experts.

17.1.4 Value Additions for Stakeholders from Cognitive Enterprise

The cognitive computing paradigm attracted most of the world business organizations, and reinvention of digital enterprise to cognitive enterprise surprises the stakeholders with remarkable benefits. The value additions such as Operational Excellence, Personalized Customer Experiences by recommending best products and services, and enhancing the level of employees are the key benefits that a cognitive enterprise achieves [2].

Case Study: IBM Watson
IBM Watson is a question-answering computing system built with cognitive capabilities. IBM's DeepQA project was developed by the research team of IBM led by David Ferrucci. It gained importance when the Watson won the famous quiz show jeopardy champions. From then IBM's Watson has been adapted by several companies for the services in healthcare, technical assistance, weather forecasting, fashion, and advertising.

17.2 Workflow Automation and Current Approaches

17.2.1 Workflows an Overview

A workflow is an orchestration of set of activities that follows a particular pattern. The workflows take input, process it, and produce the output. The workflows employed in business organization systematically map the resources into processes and provide services. Business workflows automate the business process without much human interventions. Workflows serve as an important tool in managing business process, its supervision, and control. Fig. 17.2 represents a sample workflow of the healthcare management system.

Automating the workflows of a business provides tangible and intangible benefits of various functional areas of business such as sales and marketing, customer and supplier management, project management and operations, etc.

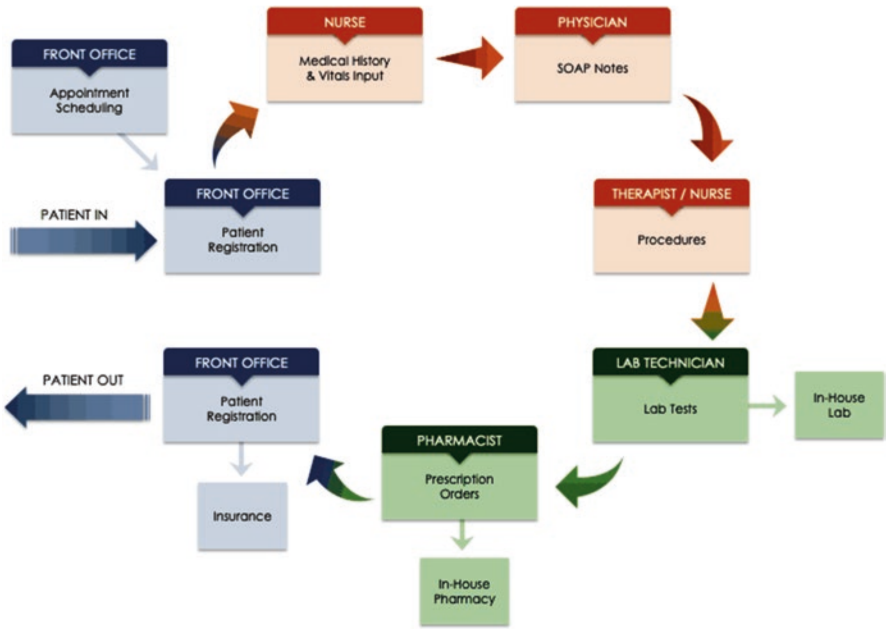


Fig. 17.2 Sample workflow representation of healthcare management system

17.2.2 Current Context of Business Workflows

Workflow automations have reached a matured state in the recent past. The enterprises today with less human interventions are able to design, automate tasks and processes, and can complete the business activities. Workflow management tools automate the business process by providing interfaces, business rule applications, and monitoring capabilities.

The workflow, whether it is simple or complex, can be automated to perform the sequence of business task. The automated workflows increase the productivity and efficiency of an enterprise. It gives accuracy in performing the various business processes with accountability.

Workflow automation has been adapted extensively in sectors such as Information Technology and Information Technology Enabled services, Banking and Financial, Production, Marketing, Sales, and Human Resource Organizations.

Even though the workflows management was automated, still there is a gap in providing intelligent capabilities to the workflows employed in business. Cognitive workflows are essential for cognitive enterprises.

17.2.3 Tools and Processes for Automated Workflow Management System

Workflow management system is a collection of tools that helps in creating, designing, managing, and executing workflows in a business. It allows the users to automate the tasks to be executed in sequence. It streamlines the activities of business process by automating the tasks, processes, and services. There are several automated workflow management software available in the market. A good workflow automation application will contain the following features:

1. A simple to use workflow designer that can even be used by users with less program knowledge.
2. GUI-based workflow designer which facilitates exact representation in designing is the one which is processed and executed.
3. Easy to integrate with other applications and distributed networks such as cloud.
4. Sophisticated reports and data protection facility.
5. Able to procure with less cost and yet containing powerful features.
6. Supports cross platforms.
7. Mature enough to learn adapt and scale.

The workflow management system should support any kind of workflow categories. The workflows used in organizations may be ad hoc which need not contain sequence of tasks and are often executed with a human intervention. On the other hand, collaborative workflows build with high business value, and sophisticated orchestration is used in business. Production workflows are workflows which automate the core activities of a company. Administrative workflow tasks are often repetitive and pre-determined. An automated workflow management system should be flexible and rich enough to handle all kinds of workflows in business sectors.

The workflow management coalition which defines standards for workflow management systems contributed the workflow reference model which clearly depicts the essential components and interfaces required for a workflow system as represented in Fig. 17.3 [3].

The workflow reference architecture depicted in Fig. 17.3 is a general architecture, and when it comes to cognitive computing and cloud-based working environment, several reinventions are necessary for the workflow architecture so that the workflows are more adaptive and intelligent.

17.2.4 Workflows and Multi-cloud Environment

Most of the enterprise applications are migrated to cloud platforms. Business enterprises operations are now transformed to service-based distributed environment such as cloud. Cloud service offerings are often attracting the business owners because of reduction in investment on Information Technology infrastructure.

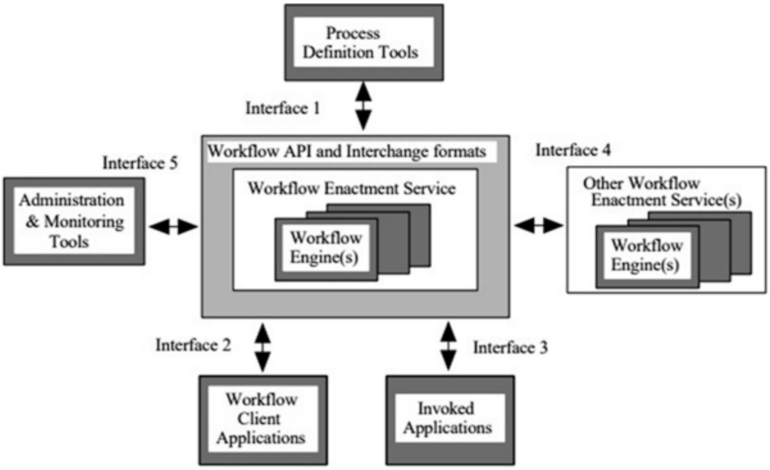


Fig. 17.3 Workflow reference architecture components and interfaces [3]

Cloud-based offerings range from simple data retrieval process to leveraging of Artificial Intelligence (AI) capabilities for the high-end businesses.

Cloud-based workflow management systems are preferred by modern businesses for the following features:

1. Reduction in operational cost of developing or procuring legacy workflow management system
2. Scalability of the resources required to execute the workflows
3. Cost-effective pricing models
4. Efficient and anywhere use and access model
5. Options to customize workflows as per the changing needs
6. Collaborative and reliable access
7. Availability of intelligent capability which can be integrated in the workflows

It is worth to note most of the workflow management services offering companies are only cloud based.

17.2.4.1 Multi-cloud: An Overview

Cloud computing model refers to ubiquitous distributed computing environment which is elastic in nature and provides on demand self-services to the cloud users. The paradigm shift happened two decades ago when cloud gained its importance in business sectors. Cloud providers emerged with all sort of business services as cloud offerings which attracted the business people. Basically, cloud provides three important service models such as Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS). Advancement in cloud gives rise to lot more service models such as Analytics-as-a-Service and

Workflows-as-a-Service. As more and more business offerings are available in cloud, a dramatic shift in way of doing business evolved. The information technology infrastructure required for the business organizations is provided by the cloud service providers with nominal subscriptions.

Cloud deployment models are categorized into Public Cloud, Private Cloud, and Hybrid Cloud. Business enterprises depending on their business needs selected the appropriate cloud model for their operations. Most of the business owners utilized public cloud for common business process and private cloud for sensitive business process as a hybrid model. As more cloud providers emerged with variety of business service offerings, the business naturally had a choice to choose the best services from different cloud service providers, and thus multi-cloud usage got started with the business community.

The use of multiple cloud services in a single heterogeneous build is termed as multi-cloud. Fig. 17.4 illustrates the concept of multi-cloud. A business enterprise leverages different services from three popular cloud service providers.

Although the multi-cloud strategy benefits organizations in choosing the best available services from cloud service providers, when it comes to implementation of intelligent workflows, there are several challenges to be addressed.

Case Study: AWS – Simple Workflow Service

Amazon is the pioneer in commercial cloud service offerings. AWS – Simple Workflow Service (SWF) is the cloud workflow service for amazons cloud applications such as Amazon Cloud Search, Amazon Simple Queue Services (SQS), Amazon Simple Notification Services (SNS), Amazon Simple Email services (SES), and Amazon SWF. Amazon provides the SWF services for its diverse customers ranging from small to large business users who use the Amazon service offerings.

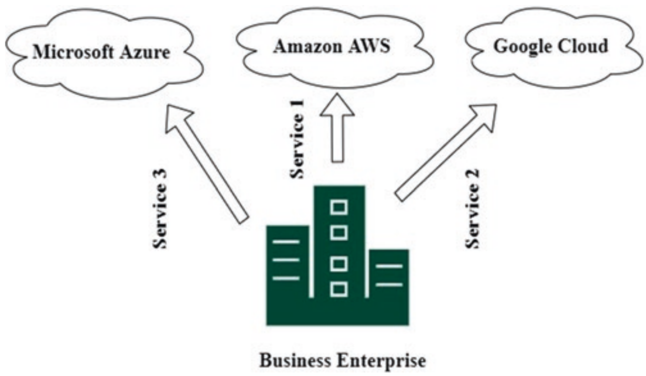


Fig. 17.4 Multi-cloud illustration

17.3 Intelligent Workflow Approaches for Cognitive Enterprise

17.3.1 Intelligent Workflows with Intelligent Agents

The rise of Artificial Intelligence and machine learning techniques transformed the digital business to cognitive business. The cognitive skills represented by the human brains are now modelled within the software and hardware applications. Traditional workflows employed by the business community are not adaptable and have no learning capabilities. Workflows when reinvented with cognitive skills become intelligent enough to make better decisions in automated way while executing the business process in cognitive enterprises.

The important difference between a conventional workflow and the intelligent workflows is that in the conventional workflows the workflow tasks to be executed in sequence or pre-built and the workflows will be executed in that sequence only, whereas in the intelligent workflows the next task to be executed is determined the intelligent agents [4]. The intelligent agents collaborate with each other and share the cognitive information they gather to guide the workflow process. Here the workflow process is actually cognitive flow process.

The conventional workflows and cognitive workflows operations are different. In the cognitive workflow the next task to be executed is determined by the intelligent agents based on the cognitive information sensed by learning the current environment. Fig. 17.5 represents the difference between traditional workflow approach and the cognitive approach [5].

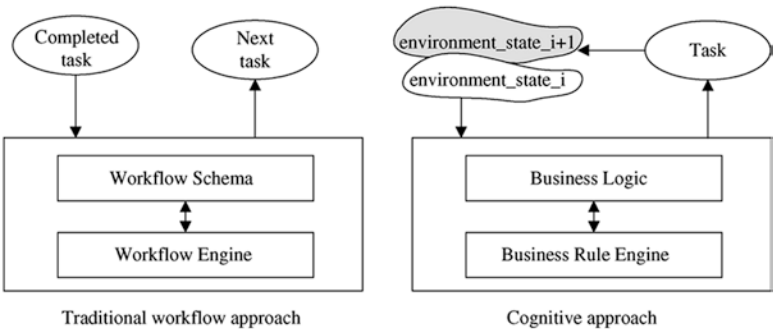


Fig. 17.5 Difference between traditional and cognitive workflow approach [5]

17.3.2 Designing of Cognitive Workflows

A cognitive workflow design requires convergence of technologies revolving around the Artificial Intelligence (AI). Several design principles which need to be considered while designing the cognitive workflows for cognitive enterprise are listed below [6]:

1. Cognitive support – Design of workflows should provide cognitive support that reduces the cognitive load of the human.
2. Collaborative work – This facilitates transfer of data, documents to variety of users depending on the context.
3. Communication – The cognitive business includes diverse applications integrated with the system. The workflow must be designed to enable communication among the integrated applications.
4. Knowledge management – Knowledge gathering from different stakeholders of the business, managing the gathered knowledge, and disseminating it in intelligent ways are essential for cognitive workflows.
5. Awareness and transparency – Awareness refers to understanding others' activities in the system, and transparency means awareness of the current status of the activities and work items.
6. Coordination – As several users work parallel in the business environment coordination, structuring and synchronization is required in collaborative business activities.

17.3.3 Design Strategies for Intelligent Workflow Adaptation for Multi-cloud

As more and more businesses are getting reengineered to operate in cloud environment, the design challenges of the workflows and intelligent workflow adaptation in multi-cloud require more attention. Multi-cloud environment does not have volunteered cloud providers to share their services and resources as in federated cloud environment, and thus the responsibility of executing of workflows and choosing of appropriate resources for the applications lie with the cloud business users. Current techniques adapted for workflow adaptation in cloud are broadly categorized as listed below:

1. Rule-based approach
2. Brokering-based approach
3. Orchestration of services approach
4. Agent-based approach

In rule-based approach, the workflow engine which is responsible for executing workflow will communicate with the adaptation engine. The adaptation engine with

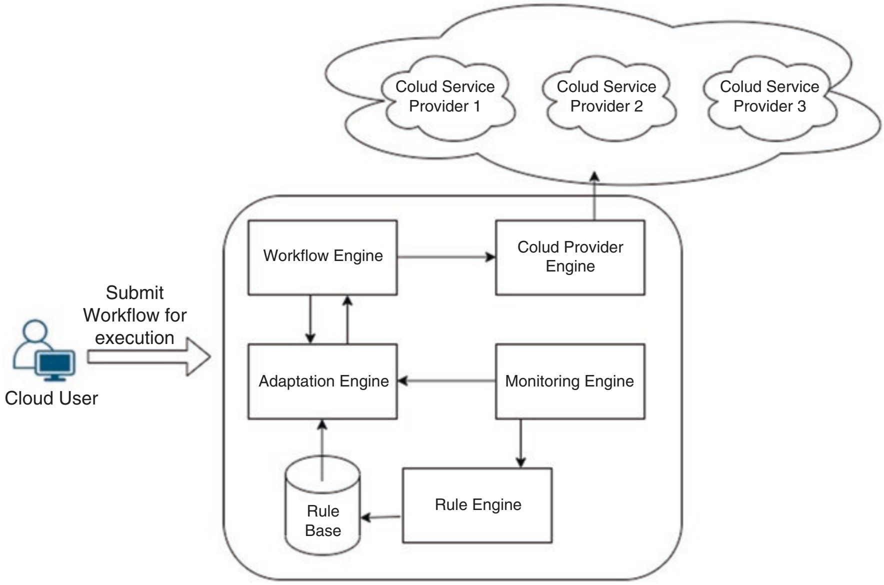


Fig. 17.6 Generic design of rule-based workflow execution in multi-cloud

the help of a monitoring engine continuously monitors the events of task execution of workflows in multi-cloud environment, and the rules are updated regularly by the adaptation engine. The rule engine contains the rule base for workflow execution path. New rules are inferred and transformed into adaptive workflows [7]. Fig. 17.6. depicts the general design of rule-based workflow execution in multi-cloud environment.

In brokering-based approach cloud service broker selects the best in breed services as required by the workflows application need. The cloud service selection methods include Multiple-Criteria Decision-Making, Analytic Hierarchy Process, Artificial Intelligence, Ontological Construct, Context-Based, Fuzzy Logic-Based, and Opinion Mining. Fig. 17.7. shows the generic design of brokering-based approach.

The different cloud providers register with the cloud service broker, and the cloud service broker maintains a services repository. The workflow engine generates the workflow request to the service broker, and the service broker matches the services and select the cloud service provider as required by the workflows.

The orchestration of services approach is not a separate model but follows the cloud brokering-based approach with sophisticated involvement of orchestration tools to facilitate Iaas, SaaS, PaaS, and cross level service orchestration [8].

In agent-based approach several intelligent agents such as Discovery agent, Matchmaking agent, and Authentication agent cooperate and coordinate together to execute the workflows task in multi-cloud environment by discovering the services,

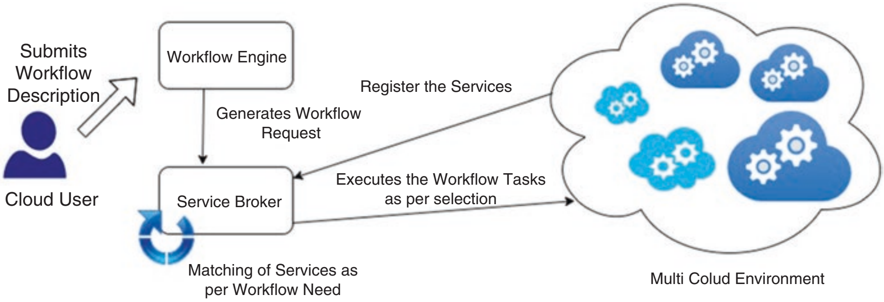


Fig. 17.7 Generic design of brokering-based workflow execution in multi-cloud

matching the services with workflow requirements, and providing authentication among several cloud providers [9].

In all the above approaches discussed Service Level Agreements (SLAs) and Service Level Objectives (SLOs) are considered as an important parameter that should not be violated in adapting the workflows in multi-cloud environment. The Cloud Application Modelling and Execution Language (CAMEL) is being used recently for deploying and executing applications in multi-cloud environment.

17.3.4 Reengineering of Workflows

The conventional workflow process should be reengineered to transform to intelligent workflows. Redesign of the company workflows is necessary with AI adoption. Reengineering workflows of a company is a complex process, and it should be carefully done with several considerations.

While the reinvention of workflow process is carried out in an enterprise to adapt intelligent workflows, the following considerations are crucial:

1. Identify the tasks of the workflow process that need intelligent automations.
2. Determine the level of automation for the identified tasks.
3. Access the impact of intelligent automation in the business process.
4. Transformation or improvements in customer experience and employee work styles gained as the result of intelligent workflow conversion.
5. Timeline required for the end-to-end transformation.

All the considerations above should be evidently discussed with all the stakeholders of the business, and how decisions are arrived should be assessed with established procedures.

Case Study: Adoption of Intelligent Workflow Automation

A European-based insurance company requires reducing the processing time on medical investigation reports. The company employees would receive a hundreds of medical investigation reports for claim processing and settlements. Reading the medical investigation reports which contain more than 10 pages manually increases the claim processing time. The company adopted the intelligent workflow automation. As a result medical investigation reports received are fed to OCR (Optical Character Reader), and analysis is done with seconds and the results are fed into the systems so that the employees dealing with claim processing take a faster decision for approving the claims. The key areas involved in intelligent workflow automations are insurance claim form processing, medical reports processing, cognitive capture of information, and digital transformation. As the result of adoption of intelligent workflow automation, the claim processing time reduced to 80%, and thus the company would engage the employees in other value added service in the saved time by automation.

17.4 Issues and Challenges in Employing Intelligent Workflows in Multi-cloud

17.4.1 Challenges in Adapting Intelligent Workflows in Multi-cloud

The adaptation of intelligent workflows in a single cloud itself poses various challenges, and when it comes to adaptation of intelligent workflows in multi-cloud environment, there are several challenges that need to be addressed. The below-listed challenges are to be addressed in adapting intelligent workflows in multi-cloud:

1. Novel research issues related to automated service selection, dynamic service composition, comprehensive SLA model, service migration and upgradation, and privacy issues are major concerns.
2. Dynamicity in services and scalability of resources.
3. Data flow between different cloud vendors.
4. Security and fault tolerance issues.

The intelligent agent broker-based approach has to be improved by addressing the shortfalls of cloud ontology and context-based service selection, improving collaboration of agents with reduced waiting time.

17.4.2 Issues and Challenges in Multi-cloud Business Operations

Basically, the cloud environment should ensure self-provisioning of Information Technology Resources to various business operations and facilitate workflow management of a business organization. These two are the key features for a cloud-based business. The multi-cloud architecture face challenges in providing the above features to the users.

The multi-cloud management involves the following activities:

1. Infrastructure provisioning
2. Algorithmic load distribution
3. Ensuring connectivity
4. Managing data transfer
5. Cross-cloud service discovery
6. Managing security

Different cloud vendors follow different cloud strategies and policies which hinder the multi-cloud usage by the business organizations.

The issues in multi-cloud are identified in the area of data governance and compliance, different skill sets from multiple vendors, security issues, compliance and security standards, portal differences, cost management, and application management differences by various service providers. In addition challenges have been faced in scaling and monitoring of resources in multi-cloud environment.

17.4.3 Benefits of Multi-cloud for Business

Though there are challenges and issues in multi-cloud business operations, there are several benefits gained by the business through multi-cloud presence as listed below:

1. The business users can find the state-of-the-art business cloud business solutions for their business process.
2. Greater agility and flexibility in performing business tasks.
3. Avoids vendor lock-in of business data.
4. Competitive price advantages.
5. Data management by priority basis.

17.4.4 Use of Intelligent Workflows in Multi-cloud

Intelligent workflows empower the cognitive enterprise. Cognitive enterprises take advantages of the multi-cloud platforms and thus emerge as a new model of business. How to implement intelligent workflows in multi-cloud environment is a highly challenging question for the business corporates.

To address the issues in implementing intelligent workflows often the workflow segmentation is suggested. Enterprise workflows can be categorized into transactional and analytical. Transactional workflows which handle the standard business operations can be managed with a single cloud service. Analytical workflows which

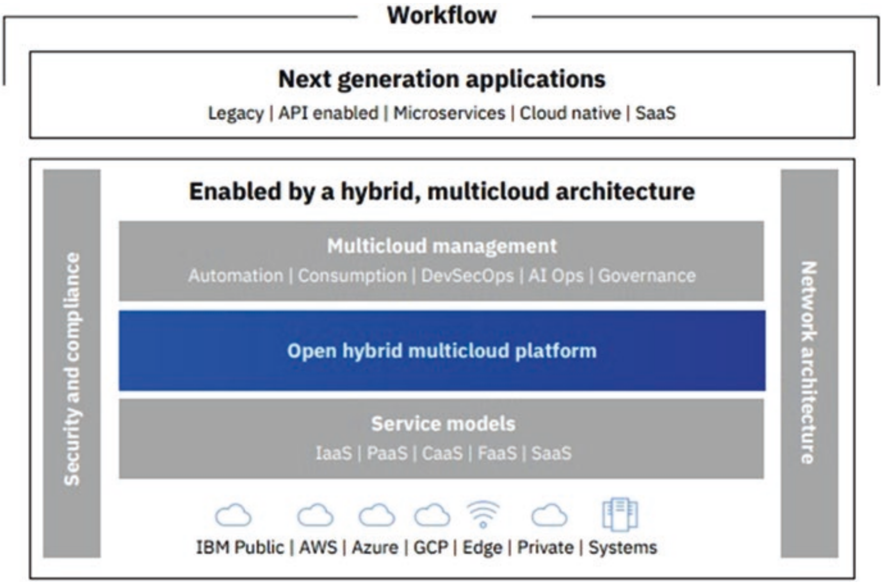


Fig. 17.8 Dynamic workflow orchestration in hybrid multi-cloud [7]

are used for building insights and decision-making can be hosted in another capable cloud service.

Another popular approach is to categorize the applications the workflows deal with and then accordingly implement the workflows segments in multi-cloud. Intelligent workflows which comprise front-end applications can be executed in public cloud offering, and the workflow which deals with back-end application can be hosted in private cloud offerings.

Dynamic orchestration of workflows is essential in multi-cloud environment. The IBM Institute for Business Value provides solutions of using workflows in hybrid multi-cloud. Fig. 17.8 represents the dynamic workflow orchestration in hybrid multi-cloud as devised by the IBM [10].

The multi-cloud service choosing is often now carried out by intelligent workflows themselves. The intelligent workflow without human intervention chooses the right service from the right vendors.

17.5 Conclusion

This chapter titled “Intelligent Workflow Adaptation in Cognitive Enterprise: Design and Techniques” explores the evolution of cognitive computing and cognitive enterprises and transformation of existing business process to cognitive model. It describes the cognitive skills necessary for cognitive enterprise applications. It summarizes the workflow automation and tools and processes for automated workflow management. It explains in detail the workflow execution in multi-cloud and in specific concentrates on intelligent workflow adaptation in cognitive enterprise. It describes the various workflow design strategies for multi-cloud environment and the techniques for designing cognitive workflows. Reengineering of current workflows and case studies are given for understanding. Also the chapter throws light on challenges and issues to be addressed in adapting intelligent workflows in multi-cloud environment and challenges in multi-cloud business operations. The benefits and uses of multi-cloud and intelligent workflows are pointed out.

References

1. https://cognitivecomputingconsortium.com/resources/cognitive-computing-defined/.
2. Experts Insights. The cognitive Enterprise: Reinventing your company with AI. *IBM Institute for Business Value*.
3. The Workflow Management Coalition Specification. The Workflow Reference Model. (1995). Document Number TC00–1003.
4. Zhuge, H. (2003). Workflow- and agent-based cognitive flow management for distributed team cooperation. *Information & Management, 40*(5), 419–429. https://doi.org/10.1016/s0378-7206(02)00061-7
5. Wang, M., & Wang, H. (2006). From process logic to business logic - a cognitive approach to business process management. *Information & Management, 43*(2), 179–193. https://doi.org/10.1016/j.im.2005.06.001
6. Hyysalo, J., Oivo, M., & Kuvaja, P. (2017). A design theory for cognitive workflow systems. *International Journal of Software Engineering and Knowledge Engineering, 27*(01), 125–151. https://doi.org/10.1142/s0218194017500061
7. Kritikos, K., Zeginis, C., Politaki, E., & Plexousakis, D. (2019). Towards the modelling of adaptation rules and histories for multi-cloud applications. Proceedings of the 9th international conference on cloud computing and services science. https://doi.org/10.5220/0007706503000307.
8. Kritikos, K., Zeginis, C., Iranzo, J., Gonzalez, R. S., Seybold, D., Griesinger, F., & Domaschka, J. (2019). Multi-cloud provisioning of business processes. *Journal of Cloud Computing, 8*(1). https://doi.org/10.1186/s13677-019-0143-x.
9. Sim, K. M. (2019). Agent-based approaches for intelligent Intercloud resource allocation. *IEEE Transactions on Cloud Computing, 7*(2), 442–455. https://doi.org/10.1109/tcc.2016.2628375
10. Research Insights. Building the cognitive Enterprise: Nine action areas, IBM Institute for Business Value.

Chapter 18
Broker-Based Collaborative Auction Method for Resource Scheduling in Cloud Computing

R. Anantha kumar and K. Kartheeban

18.1 Introduction

The computing system is quickly changing to creating programming for millions to utility as a model, instead of to execute on their separate PCs. In distributed computing, the vast majority of the models depend on pay per utility [1]. The cloud virtualization innovation is commonly utilized in cloud systems, which is utilized to isolate a solitary physical system into numerous virtual systems in a practical manner [2]. Virtual systems ought to be progressively allocated to coordinate authentic application requests, instead of the peak requests.

Cloud suppliers need to guarantee that they could be adaptable in their provision deliverance to meet different purchaser necessities, at the same time as keeping the clients detached from the fundamental framework [3]. Utilizing server virtualization system, a physical machine can execute different virtual systems (VMs). The Virtual Machines will get fine-grained enhancement of processing resources which progressively moved the exist VM activities from framework, for example, create, relocation and cancellation [4].

To acquire huge execution for application heterogeneous frameworks, allocation methods assume a significant responsibility. Instances about such applications incorporate airplane run frameworks, transport frameworks, and clinical hardware [5]. It comprises of a great deal of trusted cloud suppliers with private data of their

R. Anantha kumar (✉)
Department of Computer Science and Engineering, Kalasalingam Institute of Technology, Krishnankoil, India

K. Kartheeban
Department of Computer Science and Engineering, Kalasalingam Academic Research Education, Krishnankoil, India
e-mail: k.kartheeban@klu.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_18

resources in business multi-cloud scenario. The issue of allocation of control flows in heterogeneous PC frameworks is called in NP-Complete [6].

Cloud Service Broker goes about as a mediator among Cloud Service Client and Cloud Service Supplier [7] as depicted through Fig. 18.1. Broker chooses the great supplier for the client as indicated by the buyer's prerequisites and supplier's implementation between others; the advantages of utilizing a mediator for a buyer are expanded adaptability, simplicity of model, and diminished expense.

Regarding adaptability, by not reaching various cloud suppliers legitimately, the shopper should just make a record in broker, and the separate buyer will be given permission to numerous suppliers. By utilizing broker, shopper can stay away from to be protected a solitary supplier and obtain the opportunity to migration among cloud suppliers.

Cloud suppliers hold their individual conditions also classifications for highlights in their items that might be complicated for new clients [8]. To aggravate it even, the determinations of the equipment, programming, and organizations are difficult to be thought about.

Broker attempts to give the simple utilization of the model via rearranging the resource choice toward clients. Broker could assist clear the expense of leasing physical machines in cloud suppliers via helping buyer in choosing the majority reasonable price of initial stage for shopper's product.

The referenced advantages that are given by broker to shopper are generally appreciated simply during the utilization of the product. Especially, it's a static resource allocation. But the distributed computing industry is evolving insistently. Another element, another cloud supplier, another datacenter, or another pricing plan is being presented quickly by the market. The previously executed software should get the most recent proposals from the market into concern as they likely can enhance execution and diminish cost. For instance, the ongoing dispatch of Singapore cloud datacenter through Google Cloud Platform (GCP) [9] ought to be concerned by the nearest clients around there. Accordingly, it is required for a component in broker to effectively search for the best accessible resources for the clients. Such element is called "dynamic resource allocation."

It's not generally an obvious cycle on programming or application movement between cloud suppliers. Every supplier has their individual characteristics and designs on how the product can execute on their IaaS (Infrastructure as a Service) and PaaS (Platform as a Service). This is a significant detriment for purchasers since

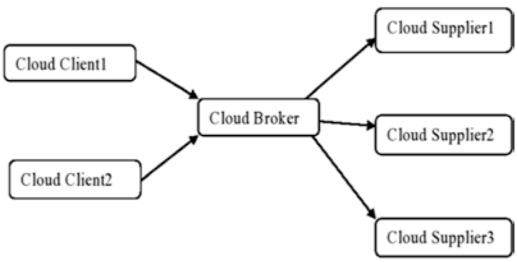


Fig. 18.1 Cloud service broker system

they are assured merchants [10]. Subsequently, the underlying advantage, adaptability, is invalidated by supplier's inflexibility. For relocation of a product or an application between cloud suppliers easily, certain necessities should be incorporated into broker [11].

The broker-based innovation manages job allotment problem in cloud frameworks which got from distributed computational intelligence (DCI) [12]. As indicated by Wooldridge and Jennings, a broker has fit for controlling its individual authoritative in the cloud framework which gives an approach to assign jobs. In broker-based scheduling procedure, it has numerous individual brokers which are equipped for dealing with multi-broker framework. The brokers are associating with one another in a framework which can collaborate, organize, and arrange [13].

18.2 Related Works

Resource allocation methodologies are utilized in numerous purposes in different territories. Allocating methods could be either fixed or dynamic proposed by Qin et al [5]. The (QoS) quality of service prerequisites should be tended to in different tough and also delicate real-time frameworks. In fault tolerant system presents essential reinforcement form which implements the reinforcement allocator at the same time while implementing essential allocator in fault tolerant framework [14]. It accomplishes large adaptability and accessibility with schedulability about resources.

VM absorbs numerous provisions which decrease resource utilization; this problem is understood via Genetic Algorithm (GA) [4]. The sort I with sort-II fuzzy frameworks forecasts the provision accessibility and workloads to enhance the framework accessibility with execution [15]. Resource allocation methodology for application work process on IaaS cloud, in which the swarm optimization strategy limits the general work process implementation in time restriction recommended by Buyya et al [1].

An energy-productive allocation strategyfor Virtual Machine distribution it examines the greatest supplier's benefit is presented by Berral et al. [16]. In energy-proficient distributing method, to accomplish energy absorbing in cloud framework, it executes live movements in VMs, proposed by Armbrust et al. [2]. However, the aforementioned method does not answer problem about adaptability with inefficient allocation productively. Then, allocation about jobs with resources ought to be executed powerfully in the cloud system.

In broker-based allocation, the jobs are assigned progressively in an effective way. The broker-based distributed manufacturing allocation depends essentially on the mix of market-based methods utilized by the multi-breaker frameworks [17]. Further, allocation is done in distributed framework which permits parallel computing and enhances the presentation of the framework [18].

In analysis, we researched novel breaker based allocation to enhance the allocation dependent on CNP system. The new system embraces the collaborative auction method where the two jobs with resources are reporters and auctioneers.

18.3 System Design

In this part, we manage framework method, documentation with proposed form design utilized in our effort.

18.3.1 System Model

Our framework design is described in Fig. 18.2. Our broker-dependent collaborative auction method is planned with three principle brokers specifically VM broker, job broker, and manager broker. The essential collaboration between brokers is implemented via the accompanying steps: primarily the cycle begins with VM set. Beginning VM set, another VM is created which also refreshed in the VM broker. At that point, the VM data is forwarded to manager broker. From the job set, another

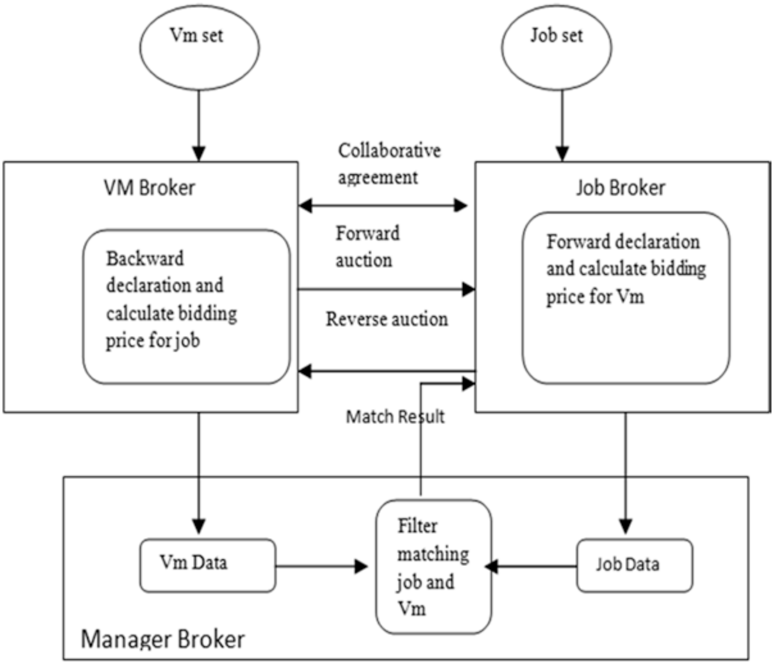


Fig. 18.2 System architecture

job is created in job broker. Job broker forwards job data to manager broker. Afterward, manager broker penetrates coordinating outcome and forwards to job broker. At that point, job broker delivers forward declaration to VM broker. VM broker computes forward auction and delivers to job broker. Then, job broker granted VMs to VM broker. Though VM broker delivers in reverse declaration to job broker, it computes in reverse auction and delivers to VM broker. In the end, collaborative agreement is granted among job with VM broker. In the present cycle, Virtual Machine could be included powerfully as another job is reached.

18.3.2 Broker-Related Form

Our broker-related form comprises of three brokers, i.e., job broker, VM broker, and manager broker. The broker form is considered as follows:

$J^B = \{J_M{}^B > M = 1,2,\ldots J|\}$ is a job broker set where

$J_M{}^B$ represents the m^th^ job broker in J^B.

$V^B = \{V^B{}_{na} > n = 1,2\ldots|V_a| > A = 1,2,\ldots|PH_a|\}$is a VM broker set,

where $V^B{}_{nk}$ denotes the $\underline{n}^{\text{th}}$ V Mon h_K.

m^B represents a manager broker.

18.4 Broker-Based Collaborative Auction Method (BCA)

The collaborative auction method comprises of three stages, i.e., fundamental coordinating stage, forward declaration auction stage, and in reverse declaration auction stage.

18.4.1 Fundamental Coordinating Stage

In this stage, VM needs to fulfill the fundamental prerequisite of jobs. In the event that it fulfills, the manager broker will penetrate the job and VM. The cycle can be clarified in detail as follows:

1. At the point when another job visits, the new job broker is created. The job broker forwards the job data to manager broker.
2. After another job visits, progressively another VM is created via VM broker. The VM broker forwards VM data toward manager broker.
3. Manager broker gets job with VM data, and at that point coordinates the data and penetrates the outcomes.
4. Then the manager broker delivers the chosen VM data to job broker.

18.4.2 Forward Declaration Auction Stage

In this stage, the job broker chooses the VM to implement job in it. VM is chosen depending on auction price. The cycle could be clarified as follows:

1. Initially, job broker gets VM data from manager broker.
2. Next every job broker forwards job data including its boundary, for example, priority, arrival time, length, and due date, to VM broker.
3. Subsequent to creating forward declaration, VM broker computes auction price.
4. Job broker gets the auction price of VM broker and makes forward agreement. The job brokers discuss with VM broker to choose at least one VMs for a job on circumstance and the goal of the VMs can ensure the timing restriction.

18.4.3 Reverse Declaration Auction Stage

In this stage, the VM broker chooses the job to execute on it. It chooses the job dependent on the auction price. The cycle might be clarified as follows:

1. VM broker forwards in reverse declaration to job broker.
2. At that point the job broker gets the VM broker data and computes the auction price.
3. After computation, job broker forwards the auction price to VM broker.
4. Next, the VM broker gets the auction price and makes in reverse agreement.
5. Afterward, collaborative agreement is performed among job and VM broker.

18.4.4 Auction Computation

The allotment is primarily related on the auction price. The forward-declaration stage manages forward auction price, and the backward declaration stage manages in reverse auction price.

18.4.4.1 Forward Auction Calculation

In forward-declaration stage, the job broker chooses the VM dependent on the computation of forward auction price. The forward auction computation scenario is as follows:

$$Fa_{ijk} = dt_i - et_{ijk} - f_{pjk}$$

The f_{pjk} signifies the previous job completion time on a similar Virtual Machine. If $Fa_{ijk} > 0$, it can be ready to finish the job prior to due date. Else it may not complete the job prior to due date.

18.4.4.2 Backward Auction Calculation

In this backward-declaration stage, VM broker chooses job related on the computation about backward auction price. The backward auction computation scenario is follows:

$$ba_{ijk} = P_i^Q \cdot e_{ijk} / \left(d_i - f_{ij}\right) \cdot \sum_{K=Ki}^{K} \sum_{J=Ji}^{J} fb_{ijk}$$

here ba_{ijk} is applied to mention auction price about task ti to V_{jk} with factor Q mentions weight of precedence, for allotting the task. The highest precedence is taken.

18.5 Broker-Based Collaborative Auction Scheduling Method (BCAS)

We present our novel broker-based allocation method in distributed computing – BCAS based on our broker-based scheduling form for autonomous, periodic, real-time jobs.

18.5.1 Responsibility About Manager Broker

Manager gets the novel jobs and also selects VM which fulfills the essential necessity about job brokers. At that point it delivers the chosen VM to job broker to execute the job.

18.5.2 Responsibility of Job Broker

Job broker delivers the declaration data to VM broker. As per forward auction price which is determined by VM broker, job broker chooses a VM dependent on determination technique and makes forward agreement for VM.

18.5.3 Responsibility of VM Broker

VM broker forwards the declaration data to job broker. As indicated by in reverse auction price which is determined by job broker, VM broker chooses a job dependent on determination technique and makes collaborative agreement among job and VM.

Broker-Based Collaborative Auction Scheduling Algorithm (BCAS)
Step1: *newVm* Creation with processing power *Pnew*;
Step2: Assign $v_{new=NULL,}$ *discover= FALSE;*
Step3: **foreach**ph_k *in* ph_a*do*
Step4: **if**ph_k *can*contain *newVM***then**
Step5: Assign ph= *newVM;*
Step6: Assign $v_{new=}$*newVM*,and *discover = TRUE;*
Step7: ***if*** *discover = FALSE* ***next***
Step8: physicalHost←Transfer VM between physical hosts towards create place for new incomingVM;
Step9: **if***physicalHost≠NULL****next***
Step10: Assign *physicalHost= newVM;*
Step11: Assign $v_{new=}$*newVM*,; *discover = TRUE;*
Step12: **if** *discover = FALSE* next
Step13: Switch on physical hostph_{new} *in* $PH-PH_a$;
Step14: **if** ability about ph_{new} *fulfils* P_{new}**next**
Step15: Assign *newVM* = ph_{new}&&v_{new}=*newVM*;

In this method, it makes another VM while a new job is reached. Row 2–6 speaks with that physical host may ready to contain the requested VM. If it contains, then allot recent VM to physical host. Row 8 speaks to VM about that movement while physical host does not contain requested VM. Row 13 speaks if that relocation obtains not succeed, then it transforms inactive physical host to dynamic host.

18.6 Simulation Results

To calculate our proposed system, we utilized CloudSim for simulation. The execution of BCAS allotment is experimented via various types of methods, for example, Random Scheduling, Priority Scheduling, and Auction Scheduling.

The implementation factors applied for estimated proposed method comprise:

1. Total Execution Time (TET) defines the total time needed to execute all jobs.
2. Total Waiting Time (TWT) defines the sum of Waiting time of all jobs.

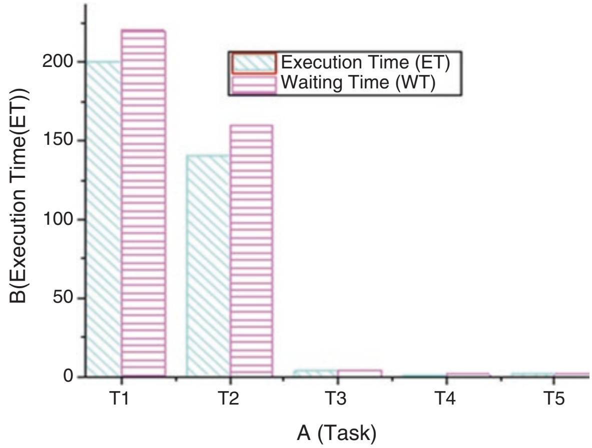


Fig. 18.3 Random scheduling

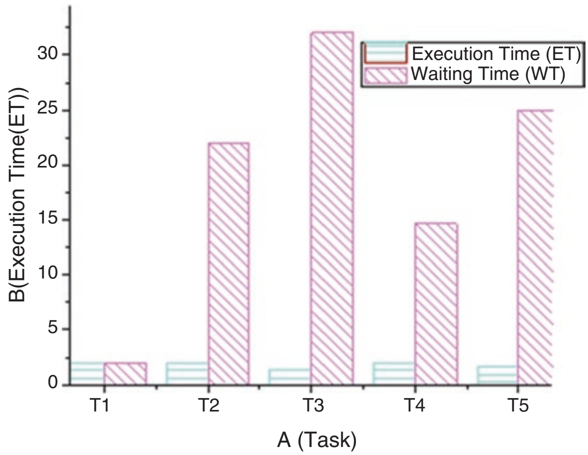


Fig. 18.4 Priority scheduling

Fig. 18.3 displays execution time with waiting time for all tasks in random scheduling.

Fig. 18.4 displays execution time and waiting time for all tasks priority scheduling.

Fig. 18.5 displays execution time and waiting time for all tasks auction scheduling.

Fig. 18.6 shows the comparison between random scheduling, priority scheduling, and auction scheduling in terms of TET and TWT. It evaluates various types of BCAS method which provides the best outcome for broker-based Collaborative Auction Scheduling method. While job count rises, framework fulfills request demand via making new VM toward complete job prior to the due date.

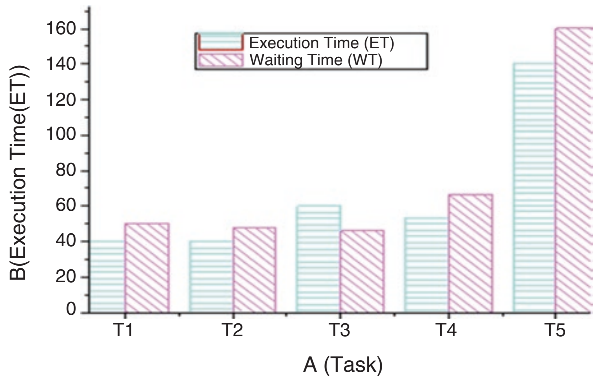


Fig. 18.5 Auction scheduling

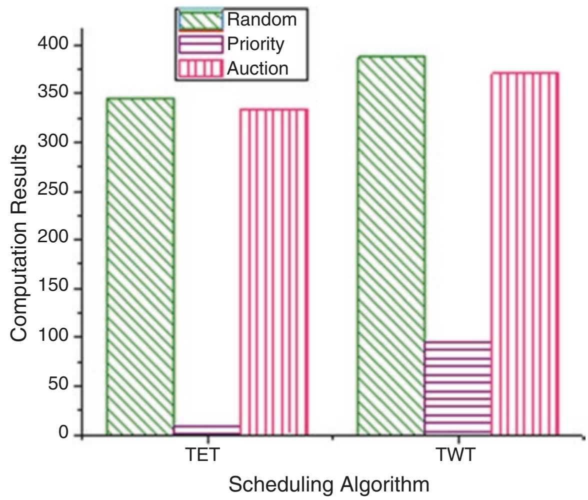


Fig. 18.6 Comparison results

18.7 Conclusion and Future Work

In our manuscript, we presented a broker-related collaborative auction method for job allocation for cloud systems. We planned various brokers to allot the jobs; it incorporates forward with reverse declaration auction system. The computation scenario and chosen techniques are utilized to grant agreement among job and VM for allocation. In our future work, rather than broker-based collaborative auction method for job allocation for cloud, we are going to execute broker-based collaborative auction method for job allocation for multi-cloud. In broker-based collaborative auction method for job allocation for multi-cloud environment, the winner of job and resource is chosen from multiple service providers.

References

1. 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 System, 57*(3), 599–616.
2. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., & Stoica, I. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50–58.
3. Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy- aware resource allocation heuristics for efficient Management of Data Centers for cloud computing. *Future Generation Computer System, 28*(5), 755–768.
4. L. He, D. Zou, Z. Zhang, C. Chen, H. Jin, and S. A. Jarvis (2014) Developing resource consolidation frameworks for moldable virtual machines in clouds, Future Generation Computer System, vol. 32, pp. 69–81.
5. Qin, X., & Jiang, H. (2006). A novel fault-tolerant scheduling algorithm for precedence constrained jobs in real- time heterogeneous systems. *Parallel Computing, 32*(5), 331–356.
6. Bittencourt, L. F., Madeira, E. R. M., & Da Fonseca, N. L. S. (2012). Scheduling in hybrid clouds. IEEE Communications Magazine. (pp. 42–47).
7. Liu, F., Tong, J., Mao, J., Bohn, R., Messina, J., Badger, L., & Leaf, D. (2011). NIST cloud computing reference architecture. *NIST Special Publication, 500*, 292.
8. Jeya, S. K., Angela, J., & Revathi, T. (2016). Broker based collaborative bidding mechanism for efficient scheduling of real time jobs in cloud computing. International conference on electrical, electronics, and optimization techniques (ICEEOT).
9. Lardinois, F. (2017). Google launches its cloud platform region in Tokyo. *TechCrunch*.
10. Malawski, M., Juve, G., Deelman, E., & Nabrzyski, J. (2012). Cost –and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds. Proceedings of the international conference for high performance computing, networking, storage and anal (SC '12) (pp. 1–11).
11. Carlini, E., Coppola, M., Dazzi, P., Ricci, L., & Righetti, G. (2011). Cloud federations in contrail. In *European conference on parallel processing* (pp. 159–168). Springer.
12. Goldingay, H., & Mourik, J. (2013). The effect of load on broker-based algorithms for distributed job allocation. *Information Science, 222*, 66C80.
13. Sim, K. M. (2012). Broker-based cloud computing. *IEEE Transactions on Services Computing, 5*(4), 564–577.
14. Zhu, X., Qin, X., & Qiu, M. (2011). QoS-aware fault-tolerant scheduling for real-time jobs on heterogeneous clusters. *IEEE Transaction of Computers, 60*(6), 800–812.
15. Google Cluster Data V2. http://code.google.com/p/google/clusterdata/wiki/ClusterData2011_1.
16. Goiri, I., Berral, J. L., Fitó, J. O., Julià, F., Nou, R., Guitart, J., Gavaldà, R., & Torres, J. (2012). Energy-efficient and multifaceted resource Management for Profit-Driven Virtualized Data Centers. *Future Generation Computer System, 28*, 718–731.
17. Yangui, S., Marshall, I.-J., Laisne, J.-P., & Tata, S. (2014). Compatible one: The open-source cloud broker. *Journal of Grid Computing, 12*(1), 93–109.
18. Ganish, Young-Woo, Seong-Woo, Emanuele (2017) Dynamic resource selection in cloud service broker", International Conference on High Performance Computing & Simulation.

Chapter 19
An Effective Cloud Broker Framework for Knowledge Discovery in Multi-Cloud Environment

M. G. Kavitha and G. Kalaiselvi

19.1 Introduction

A lot of tools like online platforms, wikis and information management, or online learning portals exist for supporting the reason for organizing, capturing, and distribution of information. Recommender Systems (RS), Natural Language Processing (NLP) or Named-Entity Recognition (NER), and Information Retrieval (IR) are the various areas that distribute the captured information. Natural language processing provides search functionality for a heterogeneous structure of the content. It deals with a variety of content like text, audio, video, source code, etc. Considering the keyword-based search, the users provide the word of identification to find the refined results by a multi-dimensional categorization scheme. The way in which the human beings understand and use language so that appropriate tools and techniques can be developed to make computer systems understand and manipulate natural languages to perform the desired tasks.

Cloud computing is an interface necessary to offer universal and smooth end user access. This technology benefits both the cloud providers and their customers based on its properties such as pliability, setting, and pooling resources. Cloud possessions may hold in reserve with respect to the amount of work done, valuing only on the temporarily utilized resources and the services provided by cloud applications, their components to be standardized. Multi-cloud environment is an approach that makes use of more than one cloud service provider for deploying the applications. This

M. G. Kavitha (✉)
Department of Computer Science and Engineering, University College of Engineering Pattukkottai, Thanjavur, India

G. Kalaiselvi
Department of Computer Science and Engineering, Anjalai Ammal Mahalingam Engineering College, Thiruvarur, India

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_19

approach may involve the use of either two private cloud services or public cloud services. This environment provides high availability and avoids application failover where the scalability to other cloud service is the base of the multi-cloud computing. Multiple cloud services are put to use to create a single underlying infrastructure, which are becoming increasingly popular among larger enterprises. Multi-cloud infrastructure can provide a greater speed due to its flexibility, and it includes small amount of storage costs.

The web is a vast resource of valuable information from which the needed information can alone be filtered based on the need of the user by spending much more time to identify related concepts avoiding the irrelevant information from a large resource of knowledge. The search may be confined to a specific set of relevant keywords provided by the user to analyze and fetch the necessary findings. A lot of knowledge management techniques are used to structure the scattered, distributed information that cannot be done manually, easily, and allow to extract the information from the search engines by combining the keywords.

Ontology defines the depiction of information by analyzing the notion of field and the associations between them. It is an unambiguous arrangement of a collective approach and provides a joint terminology, which is applied to model an idea and to exhibit the things and their mutual associations. Artificial Intelligence, the Semantic Web, Systems Engineering, Software Engineering, Biomedical Informatics, Library Science, Enterprise Bookmarking, and Information Architecture for knowledge representation are some broad areas where ontology plays a major role. It takes a range of structures although essentially it comprises the definitions of some expressions and its meaning. The uses of ontology can be specified basically to communication, interoperability, and system engineering.

The process of identifying and retrieving unstructured documents with detailed information stored in it is known as Information Retrieval. Documents written in natural language are called unstructured documents. Information Retrieval mainly focuses on the extraction of documents from an ordered definite massive group of documents on the web. Conventional text search engines are not most advantageous for finding the significant documents, and they are produced by various approaches of ontologies and semantic data. An architectural approach for Ontology-based information retrieval that uses data mining technique to classify the documents is presented in this paper.

19.2 Related Work

A model for compliance management known as Compliant Cloud Computing (C3) for insisting compliance necessities about protection and confidentiality in addition to dependence compliance level agreements is proposed [1]. An optimistic way of computing allows the operational capabilities of a computer like basic amenities of people. Service availability and throughput are some of the quantifiable non-functional performance metrics of the cloud providers.

There are a wide range of methodologies available in natural language processing; a model that automatically ensures correspondence to templates emerged [2]. Templates are efficient tools for raising the accuracy of natural language necessities that ignore the confusions that arise on the application of unrestricted natural language. Verifying conformance to templates is difficult, and it is a challenge when the requirements change frequently. The proposed method for representing templates into NLP pattern is exhibited, and it also reflects the realistic practice of implementing programmed checkers for templates in Requirements Engineering community.

A technique for dynamically structuring an ontology of famous people related searching suggestions of a search engine that are formed in response to partial user search queries is proposed and is united with data from DBpedia [3]. Ontology-Supported Web Search system is supported with dynamic versions of search suggestions since the search engines are responsible for providing users with recommended query completions as search suggestions are often uncertain; also this may refer to many homonyms. An approach that allows to construct ontology using the set of rules generated by rule-based learning system [4] is proposed that utilizes the extracted and representative rules created from the novel dataset in developing ontology elements. This idea confirms a criterion that ontologies can play a major role in large extent searching options.

In the partial computerization of creating knowledge chains, ontologies and data mining form the core of a learning process for creating personal knowledge [5] through the exchange of knowledge chains is proposed [6]. A chain toward the learner is created and recommended where all the agents observe every media used by the learner and categorize its content [7] via ontology. The semantics of AL-log collated with conventional web log files, that has more communicative semantic information and it is a dominant mixture of information representation is explained. Therefore, employing AL-log to create the knowledge base of the log ontology can determine more effective and useful access patterns [8].

Well-defined Datalog atom is adopted in the ILP framework to express the part-whole relation between atomic events and complex events. The identified relationship uses the ALC propagation rules also they are constrained SLD-reputation to study the recurrent association rules on log ontology [9]. The excellence of outcome that usual full-text search engines [10] give is still not most favorable for a lot of user queries. Ontologies and semantic [11] metadata can offer a resolution for the abovementioned problems. Development of an optimal information retrieval process and a broad framework that is based on ontology-supported semantic metadata generation and ontology-based query expansion is proposed [12].

A multi-agent system-informed ontology is used in modern environments to facilitate information storage, inference, and retrieval for the case of distributed resources. A cloud manufacturing OWL-S ontology representation is proposed based on service-oriented thinking that performs ontology mapping as initialization process and then the service request is satisfied by using an inference engine following the publishing and binding processes [13]. An extensive learning of domain ontology's evolution principle, gist, method and model [14] reconstructed predictable ontology data as specialized vocabulary, proficient phrase book, textbook that manipulate knowledge collection in order to construct an automation of domain ontology efficiently [15]. A wide range of modified services are provided to users with the advancement of cloud computing and cloud service providers. An intelligent service discovery platform is needed in search [16] of appropriate services correctly and promptly. An intelligent agent that integrates ontology for service discovery in cloud environment framework is proposed by the authors [17]. Agents are used to assist users in discovering proper service according to user need. A cloud service discovery environment that demonstrates the concept and its application is implemented in a way to estimate the accuracy of the system.

19.3 System Architecture

The preprocessed information content retrieved from the browser history is used for the classification purpose based on the user interest [18]. The knowledge is updated in the IaaS repository. Information as a Service repository maintains permanent and impermanent interests of the user and plays a major part in integrating number of solutions. Every time when a demand arises from cloud user, the broker checks for the repository after constructing the cloud ontology. The query provided by the user through the graphical user interface is reformulated by using the information present in the cloud search engine. Service relationship identified for the query by using the IaaS repository is re-ranked [19], and the user's interest is published to the cloud broker system.

The cloud broker is capable of providing a range of services that is also called as Software-as-a-Service (SaaS) on behalf of users and contributors and considered as cloud broker [20]. The broker is an entity placed in between the consumer and provider to act as an intermediary that performs functionalities such as collecting requirements, service discovery, and service composition for providing better results to the user query in a well-formulated manner by considering the search of a particular user's interest. The overview of the proposed architecture is shown Fig. 19.1.

19.3.1 Document Retrieval

The access log information is filtered from the captured web search such that the visited URL of type "text/html" is extracted as in Fig. 19.2. ODP extracts and breaks down every URL provided in the user's search. Content retrieval of the URLs is done during the training and testing phases of indexing. Internet Explorer's History View is used during the testing phase in which the URLs are extracted from the user's browsing history for classification purpose.

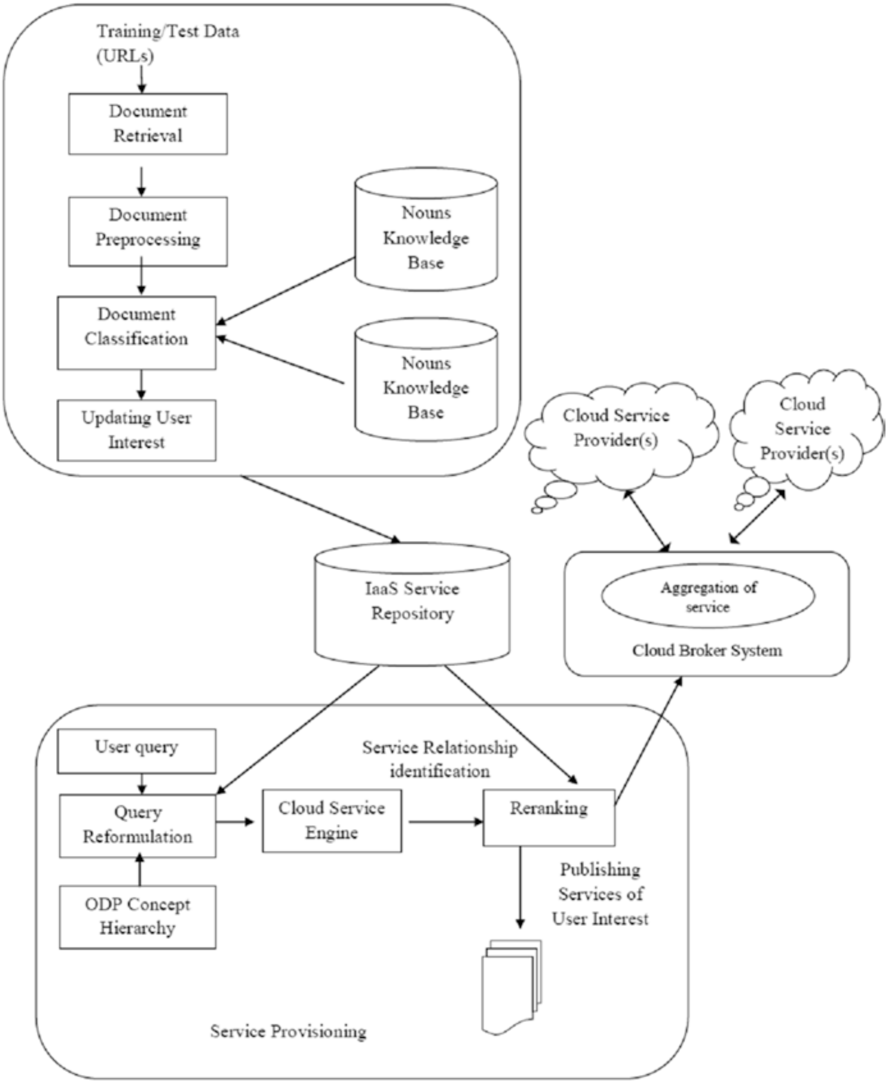


Fig. 19.1 Architecture diagram

19.3.2 Document Preprocessing

Preprocessing is an important activity to be carried out prior to document categorization in order to obtain the majority of key terms. It includes different operations such as stop words removal, stemming, nouns identification, and identifying generic form of nouns.

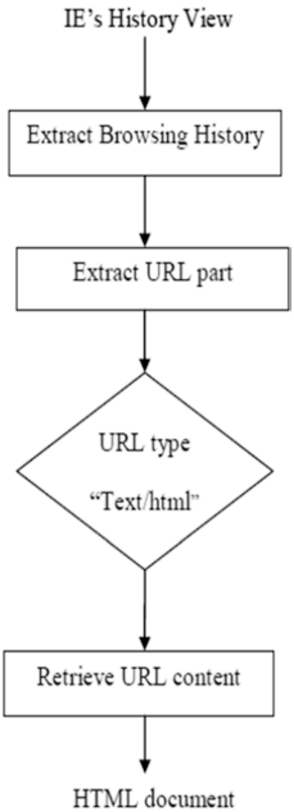


Fig. 19.2 Document retrieval

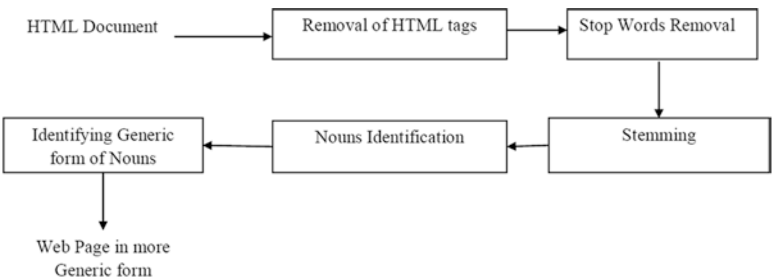


Fig. 19.3 Document preprocessing

At the end of preprocessing phase, the web documents are represented as the documents containing only the nouns in generic form, by analyzing which the category of the web document can be easily identified as represented in Fig. 19.3.

19.3.2.1 Lexical Analysis

The extraction of plain words and terms from a document, stripping out administrative metadata and structural or formatting element by eliminating HTML tags for a web page, is known as tokenization process [21] that takes place during the lexical analysis. This action desires to be completed proceeding to categorization of documents.

19.3.2.2 Stop Words Removal

Stop words removal has to be done to remove high-frequency function words that are necessarily not needed in the document. Filtering out of such ineffective vocabulary like prepositions and articles is done in the document preprocessing step.

19.3.2.3 Stemming

Stemming algorithms are deployed for removing the suffixes from verses; thus similar text like "oceaneering," "oceanic," "oceanics," "oceanization," and "oceans" can be mapped to "ocean." Lovins, Porter, and Paice/Husk are some variations of stemmers; each of these works in its own ways, such that they all have diverse things and stemming rules where efficiency diverges subject to given input. Porter's stemming algorithm is used to decrease each expression to its origin to decrease the term disparities on the classification.

19.3.2.4 Nouns Identification

To identify additional helpful patterns, in particular point nouns identification is done as they are primary featured subject of a sentence.

19.3.2.5 Identifying Generic Form of Nouns

In the direction of moderating the number of traits in a document, it is essential to identify the most common form of words instead of the already used terms. This paves a way to capture a large extent of terms of same knowledge. As an instance, if a web page has terms like vegetables, meat, juice, starter, rice, and curry, then the connection of all these terms together can be stated as food which would minimize the number of representations from many to one.

19.3.3 Document Classification

Document classification is done using Ant Miner Algorithm [22]. Document classification process is done in two phases, ODP category extraction and categorization, as depicted in Fig. 19.4.

19.3.3.1 ODP Category Extraction

The Open Directory Project is a wide-ranging index [23] that extends numerous issues but mainly focuses on Internet site categorization. A human created directory of the Internet is established to provide the information needed to the user from the web pages.

Categories are warehoused in the ODP as a mixture of verses that states a complete route to a focus like Entertainment/Videogames/Super Mario. **Words** remain as separate topics which group to generate a category. In the example, the category Entertainment/Videogames/Super Mario is separated to three words: Entertainment, Videogames, and Super Mario. ODP contains a resource description framework on the respective site that provides the arrangement of the information based on the subject category, category identifier, label, explanation, latest update, subdivisions of the topic, and figurative mentions.

19.3.3.2 Categorization

An Ant Colony Algorithm called Ant Miner Algorithm is applied in categorization phase to categorize the web search of user to ideas.

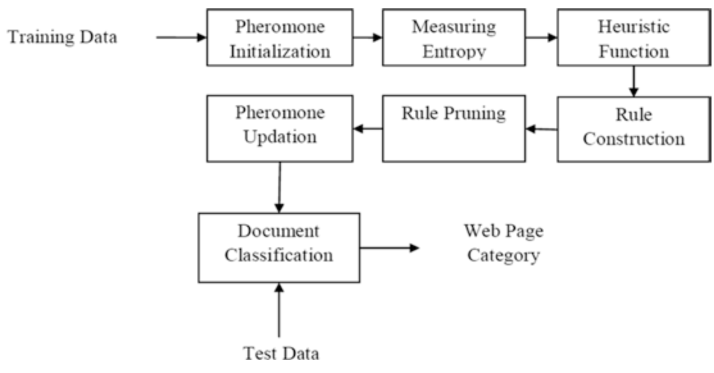


Fig. 19.4 Document classification

Ant Colony Algorithms

This class of algorithms works similar to the natural behavior of ants searching for food. A chemical pheromone present in the ants helps them to form a path one after the other toward the food which is an intelligent action arising from a group of unintelligent units known as "swarm intelligence." A strong trace of the pheromone is released in an amount by every ant to which other ants are invited to come together, which simplifies pattern they use while searching for food. In this principle, ants do not move on an unplanned path; instead they have a tendency to unite to the shortest path, such that if many ants proceed to the same minimal path, they can have supplementary tours to the food stuff and back to the colony.

When the ants make extra trips on the shorter path, they tend to accumulate large quantity of pheromone over a given distance compared to the extended path as represented in Fig. 19.5.

Ant Miner Algorithm

A similar concept taken from ant colony model deployed to data mining is known as ant miner algorithm. Initially an ant starts with empty rule and then it adds the value pairs in an iterative manner to the rules by applying probability functions with respect to the virtual pheromone and a heuristic function that measures the information gain of that particular attribute value pair. Unlike natural ant's food forage, the virtual ants forage for rules and the path. The Ant Miner Algorithm consists of different major modules like:

- Pheromone Initialization
- Heuristic Function
- Rule Construction
- Rule Pruning
- Pheromone Updation

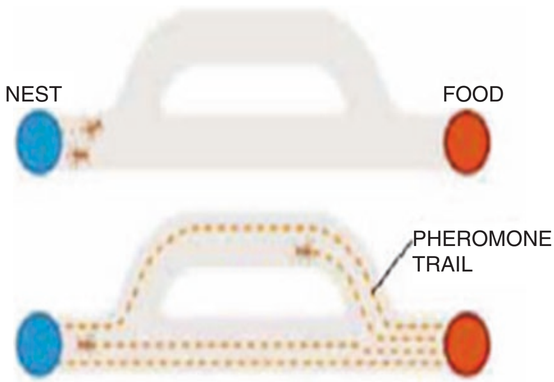


Fig. 19.5 Representation of natural pheromone trails

Pheromone Initialization

Expressions in the web page are initialized with the same amount of pheromone, given by (Eq. 19.1) where u is the number of attributes in total and v is the number of probable values in the field of attribute I.

$$\tau_{mn}\left(t=0\right)=\left(\frac{1}{\sum_{m}^{u} v_n}\right) \tag{19.1}$$

Heuristic Function

ACO algorithm uses a heuristic value along with pheromone value to compute the probability of each term being picked. The information gain measured from the entropy associated with an attribute value pair is the heuristic function (Eq. 19.2).

$$\text{info}C_{mn} = -\sum_{h}^{y=1}\left(\frac{\text{freq}T_{mn}^{\ y}}{\left|T_{mn}\right|}\right) * \log_2\left(\frac{\text{freq}T_{mn}^{\ y}}{\left|T_{mn}\right|}\right) \tag{19.2}$$

Here "*e*" is the number of classes, $|C_{ij}|$ is the total number of cases which have attribute m equal to value *n*, and FreqC_{ij}^{w} is the number of cases that *i* is equal to *j* for class *y*.

If the value *j* does not appear for the attribute *i* in any of the cases, the entropy is set to infoC_{ij} = $\log_2$ (number of classes) which corresponds to the lowest power of prediction. The term$_{ij}$ appears in only one class, and then it is set to infoT_{ij} = 0, which is the highest predictive power.

$$\eta_{mn} = \frac{\log_2(h) - \text{ info Cmn}}{\sum_{u}^{m}\sum_{vi}^{n}\log_2(h) - \text{ info Cmn}} \tag{19.3}$$

Rule Construction

IF <attribute> = <value> AND < attribute> = <value> AND … THEN <class>.

The class is predicted by the rule for the records where the predictor attributes hold as in (Eq. 19.4). Here "*a*" icand b_i is the number of values on *I* domain.

$$P_{mn} = \frac{\tau_{mn}\left(t\right)mn}{\sum_{m}^{a}\sum_{n}^{b_i}\tau_{mn}\left(t\right)\eta_{mn}.\forall \in \text{I}} \tag{19.4}$$

Rule Pruning

The predictive accuracy and simplicity of the rule can be improved using rule pruning as in (Eq. 19.5). It removes every word from the rule which can increase the quality of the rule after that action. True positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) are the different cases covered by the following expression. The rule quality is evaluated using the expression

$$Q = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \bullet \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \tag{19.5}$$

Pheromone Updation

Each ant later than constructing the rule increases the pheromone of used terms and reduces the pheromone of unused terms. Increasing the pheromone is carried out by the expression given in (Eq. 19.6).

$$\tau_{mn}\left(t+1\right) = \tau_{mn}\left(t\right) + \tau_{mn}\left(t\right) * \underset{v}{\mathrm{O}} \forall m \mid n \in \text{ to the rule} \tag{19.6}$$

Normalization is achieved by separating the value of each τ_{mn} by the summation of all τ_{mn}, $\forall$ m, n. In this way the algorithm categorizes each visited URL.

In this proposed system, during the training phase of indexing, the classification rules are generated using web documents (training data). During the testing phase of indexing, these rules are applied on the web documents visited by the user (test data), which have been collected from the Internet Explorer's History View by executing the command "iehv/sxml filename.xml" which brings the IE history view in an xml file with a name "filename" and the ODP concept of each visited web document is identified. Using these identified concepts, the interest of the user is determined [24] using which search operation is performed. Ant Miner algorithm is used for finding the category of web documents.

The browsing history is tracked and classified into different categories using classification procedure, and the results are updated in the user interest profile with their weight after getting user query.

This module functions as follows:

- Get the Internet Explorer's History View by executing the command "iehv/sxml filename.xml" which brings the IE history view in an xml file with a name "filename."
- Extract the browsing history of the user from the result of the above command.
- Extract the URL part of the current session from the xml file.
- Call the procedure for document categorization, and building of user interest profile.
- Update the identified categories in the user interest profile after eliminating the duplication.

Categorization phase creates, maintains, and uses an index called "category index" that contains the already classified web page URLs and their identified concepts. The category index is being created and updated dynamically during the testing phase. The main purpose of this category index is getting higher efficiency during classification.

The document categorization module functions as follows:

- During testing phase read the URLs visited by the user.
- Checks for the concept of the URL in the category index.
- If a concept exists, then top-level category count is incremented else the following steps are performed.
- The document-preprocessing phase for the URL is called that finally gives out the generic form of nouns.
- Classification rules are applied for each ODP category on these nouns.
- If the nouns follow any one classification rule, then the corresponding ODP category indicates the concept for the web page. Increment the corresponding category count that finally gives out the user interest.
- The URL of the web page is updated and its concept in the category index.

19.3.3.3 User Model Construction

The construction of user model comprises of the below-mentioned steps as illustrated in Fig. 19.6

- Building of User interest profile
- Current Session Tracking
- Building of User Model

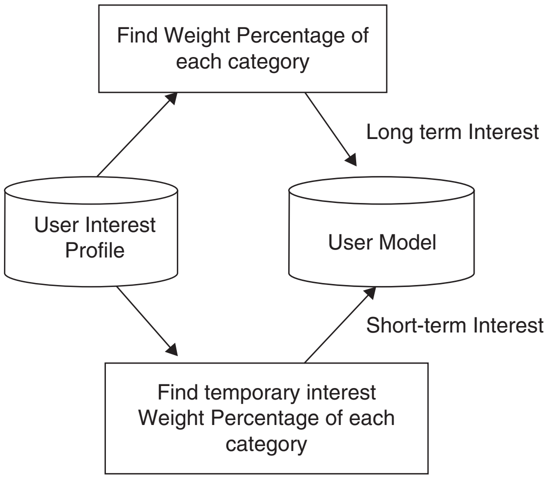


Fig. 19.6 User model construction

Building of User Interest Profile

User interest profile maintains the various concepts visited by the user along with the count for each concept. This maintains the history of user interest. The construction of user interest profile as in Fig. 19.7 is as follows:

- Find the category or concept in which the user is interested from user browsing history. If the user browsing history is empty, get the category option from the user and perform searching operation for the entered query.
- Find the weight for each class.
- Update the user interested concepts and the count in the user interest profile.
- Store concept, weight in which the log information is created during the testing phase in user interest profile.
- Identify user interest during retrieval phase by using user-interested categories.

Current Session Tracking

The browsing history of the current session is tracked, and the resulting web pages are categorized into different classes and they are updated in the prototype along with the weight. These operations are performed by the system when the search query is entered by the user:

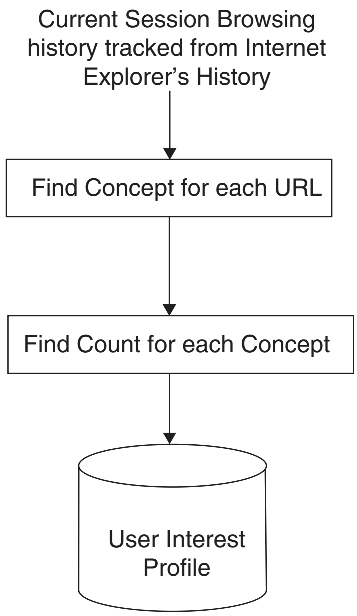


Fig. 19.7 User interest profile

- Get the Internet Explorer's History View by executing the command "iehv/sxml filename.xml" which brings the IE history view in an xml file with a name "filename."
- Extract the URL part of the current session from the above xml file.
- Call the procedure for document categorization, and building of user interest profile.
- Update the identified current session categories in the user interest profile after eliminating duplication.

Building of User Model

Construction of prototype is the core part of this work which is performed by analyzing the user browsing history. User interest profile is created and updated by making use of browsing history. The prototype acts as a reference ontology which contains user interested concepts with their corresponding weight. Hence the prototype represents permanent and impermanent interests of the user.

19.3.3.4 Information Retrieval

The indexing phase of the project comprises of the major modules document retrieval, categorization, and the user model construction. The retrieval phase comprises of three major parts, namely, as depicted in Fig. 19.8.

1. Query Reformulation
2. Searching
3. Re-ranking

Query Reformulation

Query Reformulation, which is part of Information Retrieval phase, comprises of the following steps:

- Read the query specified by the user.
- Identify the concept of entered query using ODP concept hierarchy.
- Identify the interest of user using prototype.
- Transform the query into more specific form.

Identification of Query Concept

Concept of the query given by the user has to be identified initially with the help of Open Directory Project Concept hierarchy which gives out the various categories for a given query string.

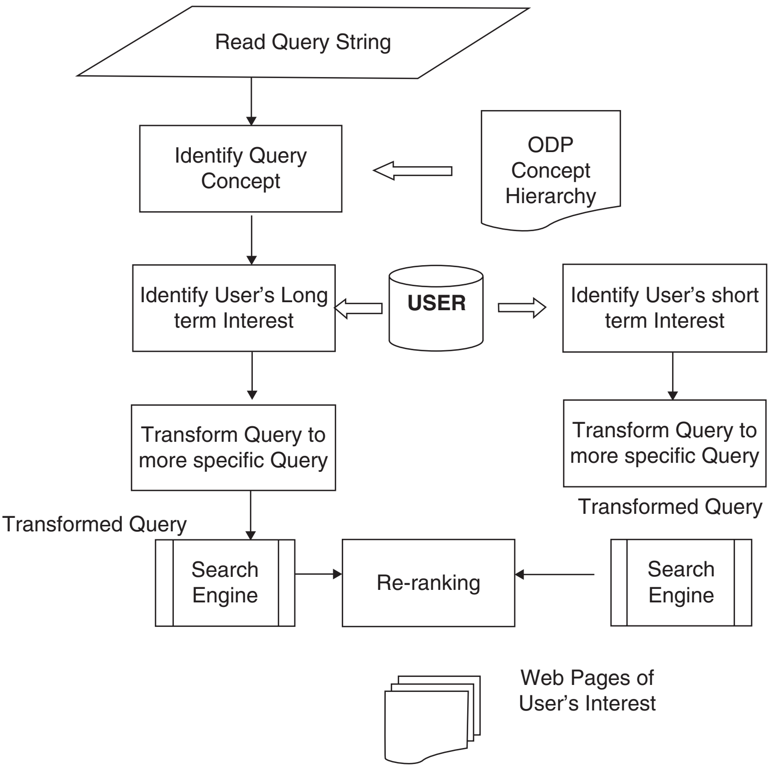


Fig. 19.8 Information retrieval

- Art: Design (20 matches)
- Commerce: Production (219)

Reference: Comprehension (63) Games: Football: Structural design (18)

The above list of categories shows that the given query is having ambiguous meaning and falls under the top-level categories [arts, business, reference, sports], among which the user may be interested in any of them that can be found in the next stage of query reformulation.

Identification of User Interest

User interest is identified from the prototype maintained by the user, and this prototype has information about the weight percentage of both permanent interest and impermanent interest for each category. Whenever user gives query through this personalized system, the content is temporally updated. ODP concept hierarchy option is obtained from user as input through this system if the user model is empty. Sample for user model content is given as follows.

Computer	14	Arts	12
Sports	10	Fruit	7
Real estate	5	Recipe	13

The first column is the various categories used by the system. Second column depicts the weight percentage of user interest. The various steps involved in identification of user interest are as follows:

- Identify the top-level categories for the query given by the user as query concepts.
- Identify the user's long-term interested concept among the query concepts.
- Identify the user's short-term interested concept among the query concepts.
- Use the obtained user interested concept to transform the query string for both permanent and impermanent interests.

Query Transformation

Query transformation can be done in many ways as depicted in Table 19.1. The proposed system transforms the query by "adding" some terms that make the query more specific. This way of doing query transformation can be called as "Query Expansion." There are many ways of providing terms for query expansion; they are:

1. Relevance feedback: User query is resubmitted by adding some keywords, which are chosen from the relevant documents.
2. Accompany the user query with related words from the thesauri.
3. Analyze each word with the perspective of words surrounding it. This contextual analysis [25] is performed with the help of a sample relevant document.
4. A probabilistic grammar is trained using a set of queries and it is helpful for recognizing new queries. Adjectives, nouns, or variations of proper names can be used for expanding the query.
5. A query log containing all queries with relevant terms is maintained which will be helpful for the expansion of new query.

The proposed system uses the combination of method 2 and method 3, that is similar to method 2, this system uses the thesauri called Open Directory Project for

Table 19.1 Query transformation types

Key	Transformation	Examples
SPL	Term splitting or joining	Rockclimb→rock climb
DEL	Term deletion—to make query less specific	Malaysia electricity→Malaysia
ADD	Term addition—to make query more specific	Windows95→windows95 help
SUB	Term substitution—semantically related terms	Electronic commerce→electronic contract
DER	Derived form of words	Tourism→tour
ABR	Abbreviation expansion or contraction	JPL→jet propulsion laboratories

Table 19.2 Example for query expansion

Query	User interest	Reformulated query
Architectural models	Computer	Architectural models computer hardware software internet
Architectural models	Real estate	Architectural models real estate property builder residential

finding the words related to the query and like method 3, words associated to the query are chosen carefully from the context of temporary and permanent interests of user separately. Table 19.2 shows the query expansion for the query "Architectural Models" using the directory ODP and the users' context of visiting web pages.

Searching

User query is given to any of the existing search engine after reformulating it separately for permanent and impermanent interest of the user. The reformulated queries are then re-ranked to retrieve web pages of users' interest that satisfy both permanent and temporary interests. The snapshots for the search result for the query "salsa" without reformulation and with reformulation are given in Figs. 19.9 and 19.10. The obtained search results from the search for the query "salsa" have ambiguous meanings dance music and recipes.

The query before reformulation retrieves the web pages of both "dance & music" and of "recipes" with the existing search engine "Alta Vista." But when the search engine works with the user model for finding the user's interest, say "Arts," then the developed system reformulates the query "salsa" with "dance & music" related terms and retrieves the web pages of the user's interest.

Re-ranking

The obtained search results contain both permanent and temporary interests of users. The final retrieved web pages for the query "salsa" given by the user, the following output will be displayed when the user is interested as depicted in Figs. 19.11 and 19.12.

- Dance & Music—permanent interest
- Recipes—impermanent interest

19.4 Results and Evaluation

Generally, Google or any other search engine gives preference to some concepts when the query is related to many concepts. For example, Google always ranks the computer-related pages higher for the query "apple" irrespective of the users' interest. If the user is interested in fruits, he/she gives the query "apple," but Google

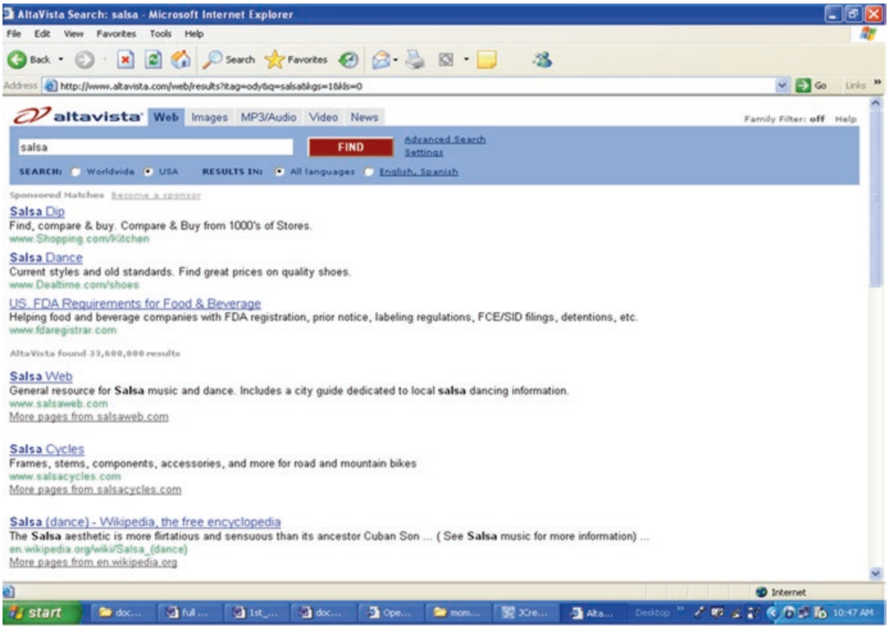


Fig. 19.9 Search result before reformulation

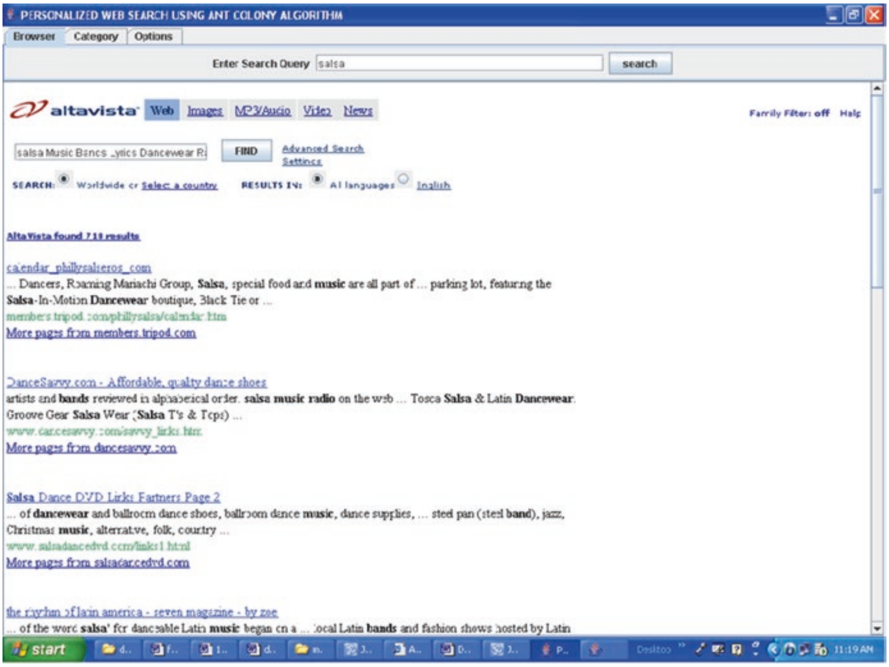
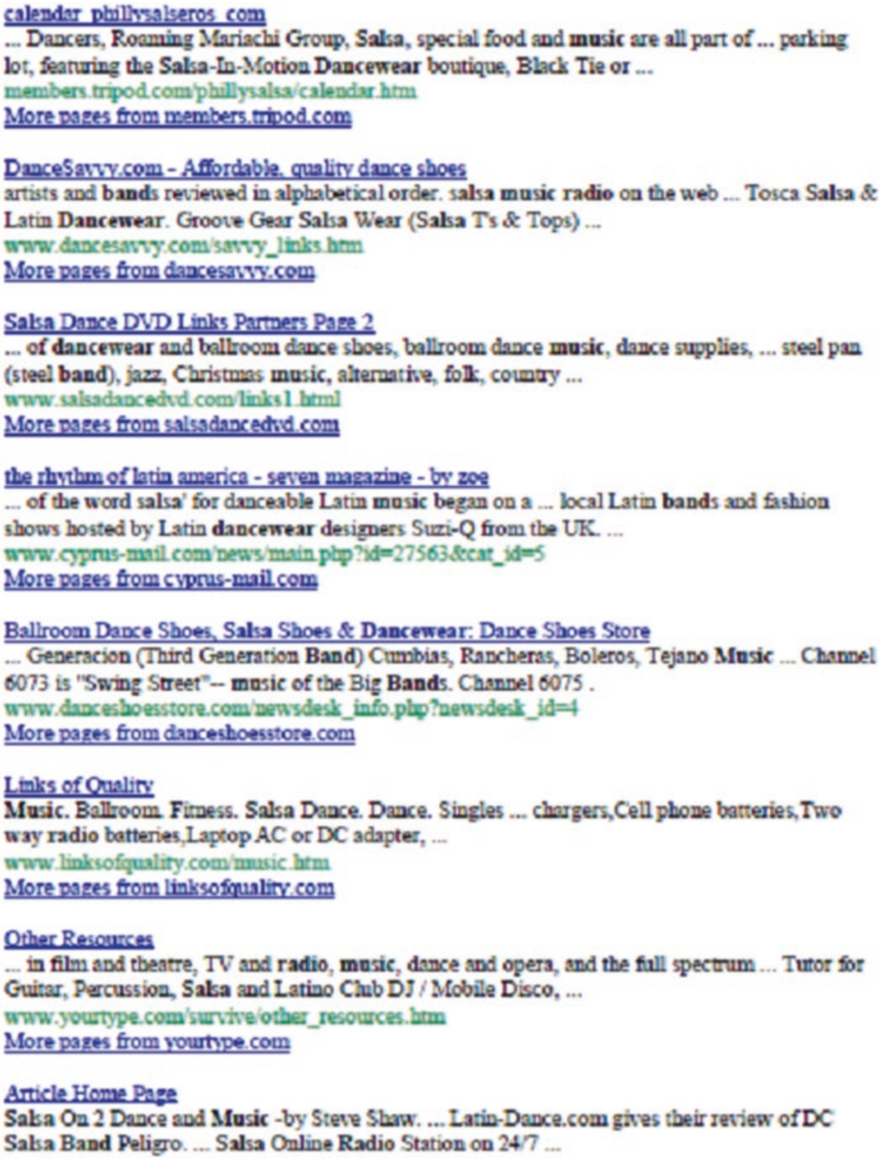


Fig. 19.10 Search result after reformulation


calendar phillysalseros com
... Dancers, Roaming Mariachi Group, Salsa, special food and **music** are all part of ... parking lot, featuring the Salsa-In-Motion **Dancewear** boutique, Black Tie or ...
members.tripod.com/phillysalsa/calendar.htm
More pages from members.tripod.com

DanceSavvy.com - Affordable, quality dance shoes
artists and **bands** reviewed in alphabetical order. salsa **music** radio on the web ... Tosca Salsa & Latin **Dancewear**. Groove Gear Salsa Wear (Salsa T's & Tops) ...
www.dancesavvy.com/savvy_links.htm
More pages from dancesavvy.com

Salsa Dance DVD Links Partners Page 2
... of **dancewear** and ballroom dance shoes, ballroom dance **music**, dance supplies, ... steel pan (steel **band**), jazz, Christmas music, alternative, folk, country ...
www.salsadancedvd.com/links1.html
More pages from salsadancedvd.com

the rhythm of latin america - seven magazine - by zoe
... of the word salsa' for danceable Latin **music** began on a ... local Latin **bands** and fashion shows hosted by Latin **dancewear** designers Suzi-Q from the UK. ...
www.cyprus-mail.com/news/main.php?id=27563&cat_id=5
More pages from cyprus-mail.com

Ballroom Dance Shoes, Salsa Shoes & Dancewear: Dance Shoes Store
... Generacion (Third Generation **Band**) Cumbias, Rancheras, Boleros, Tejano **Music** ... Channel 6073 is "Swing Street"-- **music** of the Big **Bands**. Channel 6075 .
www.danceshoesstore.com/newsdesk_info.php?newsdesk_id=4
More pages from danceshoesstore.com

Links of Quality
Music. Ballroom. Fitness. Salsa Dance. Dance. Singles ... chargers,Cell phone batteries,Two way radio batteries,Laptop AC or DC adapter, ...
www.linksofquality.com/music.htm
More pages from linksofquality.com

Other Resources
... in film and theatre, TV and radio, **music**, dance and opera, and the full spectrum ... Tutor for Guitar, Percussion, Salsa and Latino Club DJ / Mobile Disco, ...
www.yourtype.com/survive/other_resources.htm
More pages from yourtype.com

Article Home Page
Salsa On 2 Dance and **Music** -by Steve Shaw. ... Latin-Dance.com gives their review of DC Salsa **Band** Peligro. ... Salsa Online Radio Station on 24/7 ...

Fig. 19.11 Permanent interest – dance & music

shows most of the pages related to apple computers that are most irrelevant to the users. The developed personalized system reformulates the query given by the user who is interested in fruits with the terms related to fruits and gives preference to fruit-related web pages as in Table 19.3. The performance comparison of Google search result for the query with and without reformulation is shown in Tables 19.4

calendar_phillysalseros_com
... Dancers, Roaming Mariachi Group, Salsa, special food and music are all part of ... parking lot, featuring the Salsa-In-Motion Dancewear boutique, Black Tie or ...
members.tripod.com/phillysalsa/calendar.htm
More pages from members.tripod.com

DanceSavvy.com - Affordable, quality dance shoes
artists and bands reviewed in alphabetical order. salsa music radio on the web ... Tosca Salsa & Latin Dancewear. Groove Gear Salsa Wear (Salsa T's & Tops) ...
www.dancesavvy.com/savvy_links.htm
More pages from dancesavvy.com

Salsa Dance DVD Links Partners Page 2
... of dancewear and ballroom dance shoes, ballroom dance music, dance supplies, ... steel pan (steel band), jazz, Christmas music, alternative, folk, country ...
www.salsadancedvd.com/links1.html
More pages from salsadancedvd.com

the rhythm of latin america - seven magazine - by zoe
... of the word salsa' for danceable Latin music began on a ... local Latin bands and fashion shows hosted by Latin dancewear designers Suzi-Q from the UK ...
www.cyprus-mail.com/news/main.php?id=27563&cat_id=5
More pages from cyprus-mail.com

Ballroom Dance Shoes, Salsa Shoes & Dancewear: Dance Shoes Store
... Generacion (Third Generation Band) Cumbias, Rancheras, Boleros, Tejano Music ... Channel 6073 is "Swing Street"-- music of the Big Bands. Channel 6075 .
www.danceshoesstore.com/newsdesk_info.php?newsdesk_id=4
More pages from danceshoesstore.com

Links of Quality
Music. Ballroom. Fitness. Salsa Dance. Dance. Singles ... chargers,Cell phone batteries,Two way radio batteries,Laptop AC or DC adapter, ...
www.linksofquality.com/music.htm
More pages from linksofquality.com

Other Resources
... in film and theatre, TV and radio, music, dance and opera, and the full spectrum ... Tutor for Guitar, Percussion, Salsa and Latino Club DJ / Mobile Disco, ...
www.yourtype.com/survive/other_resources.htm
More pages from yourtype.com

Article Home Page
Salsa On 2 Dance and Music -by Steve Shaw. ... Latin-Dance.com gives their review of DC Salsa Band Peligro. ... Salsa Online Radio Station on 24/7 ...

Fig. 19.12 Impermanent interest – recipes

and 19.5. The pages with top 10 ranks after reformulating a query were ranked much lower before reformulation. This shows that the system that reformulates the query having ambiguous meaning with the help of user model ranks the search result of Google according to users' interest. Since the users are always referring to the highly ranked web pages and there is possibility of neglecting to visit the low

Table 19.3 Search result comparison between the developed system and Google

User's interest	Query	Ambiguous meanings	Google's relevance percentage	System's relevance percentage
Fruits	Apple	Computers Home/fruits	Computers – 80% Home/fruits – 10% Others – 10%	Home/fruits – 100%
Computers	Architecture	Real estate Computers	Real estate – 55% Computers – 35% Others – 10%	Computers – 100%

Table 19.4 Search result comparison for the query "architectural models real estate"

Displayed URLs for the query "architectural models real estate"	Page rank after reformulation	Page rank without reformulation
www.architectural-models.com/gallery.html	1	8
profiles.themonitor.com/kb-home	2	4
activerain.com/action/navy/show_featured/Builder-Contractor	3	11
www.southwestcoastrealty.com/VERANDAH	4	9
www.vtbuilders.com	5	327
bellagiorealestate.com	6	102
www.raiseyrealestate.com/newconstruction.htm	7	14

Table 19.5 Search result comparison for the query "architectural models computer"

Displayed URLs for the query "architectural models computer"	Page rank after reformulation	Page rank without reformulation
dmoz.org/Computers/Software/Graphics/3D/Models	1	20
en.wikipedia.org/wiki/Open_software	2	9
www-scf.usc.edu/~rsoma/papers/ViSAC.pdf	3	11
www.informationweek.com/story/IWK20020212S0002	4	22
www.miami.edu/umbulletin/pdf/CL0304en.pdf	5	90
csdl2.computer.org/persagen/DLAbsToc.jsp?resourcePath=/dl/...l&DOI=10.1109/TSE.2006.66	6	>1000
www.rspa.com/reflib/ArchitecturalDesign.html	7	15

ranked pages, it is necessary to rank the relevant pages high. This is being done with the developed system. The obtained search result of this system is highly related to permanent and impermanent interests of the user.

Table 19.6 shows URLs obtained after re-ranking that combines the search result of the permanent interested concept related query and temporary interested concept related query for the user. It is necessary to make the query to be more specific with the user's interest and make the search engine to retrieve a small number of web

Table 19.6 Re-ranked search results

Displayed URLs after re-ranking	Page rank	User interest
www.realestatejournal.com/propertyreport/residential	1	Short-term interest "real estate"
www.usarchitecture.com/By_the_State/Florida/Real_Estate.htm	2	
www.lancodevelopment.com	3	
www.usarchitecture.com/Real_Estate.htm	4	
www.nwbuildnet.com/stores/calc/real	5	
www.realestatejournal.com/commercial	6	
www.christyrealestate.com	7	
www.saxerealestate.com/map/architecture.htm	8	
www.apple.com/software	1	Long-term interest "computers"
www.apple.com/de	2	
www.compusa.com	3	
developer.apple.com	4	
www.nextag.com/Apple-Xsan-version-1-73791913/prices-html	5	
store.apple.com/Apple/WebObjects/germanstore	6	
www.apple.com/education/powerschool	7	
en.wikipedia.org/wiki/ApplleComputer	8	

Table 19.7 Retrieved page count by google

Query	No. of pages before reformulation	No. of pages after reformulation
Apple	212,000,000	104,000,000-computers 1,710,000-fruits
Gaming	155,000,000	90,100,000– computers 56,100,000 – sports

pages related to interest of the user as shown in Table 19.7. While a query is given by the user, Google returns many pages as search result. But it has been minimized when the query is reformulated by the user-interested concepts that are available in the user model.

19.5 Conclusion

Analyzing, designing, developing, and testing a system which performs conceptual and personalized search is the core part of this system. A prototype maintaining the user interested perceptions is built from this proposed system. The user model is used to perform the personalization of the search engines. This project mainly concentrates on four issues, how to implicitly learn the user interest, how to track the user's interest, how to update the user model dynamically with the interest (permanent/temporary) of the user, and how to re-index the search engines result.

References

1. Brandic, I., Dustdar, S., Anstett, T., Schumm, D., Leymann, F., & Konrad, R. (2010). Compliant cloud computing (C3): Architecture and language support for user-driven compliance management in Clouds. IEEE 3rd international conference on cloud computing, Miami, FL, USA, 2010 (pp. 244–251). https://doi.org/10.1109/CLOUD.2010.42.
2. Ochs, C., Tian, T., Geller, J., & Chun, S. A. (2011). Google knows who is famous today – building an ontology from search engine knowledge and DBpedia. IEEE Fifth international conference on semantic computing, Palo Alto, CA, USA (pp. 320–327). https://doi.org/10.1109/ICSC.2011.50.
3. Arora, C., Sabetzadeh, M., Briand, L., & Zimmer, F. (2015). Automated checking of conformance to requirements templates using natural language processing. *IEEE Transactions on Software Engineering, 41*(10), 944–968. https://doi.org/10.1109/TSE.2015.2428709
4. Kharbat, F., & Ghalayini, H. (2009). New algorithm for building ontology from existing rules: A case study. International conference on information management and engineering, Kuala Lumpur, Malaysia (pp. 12–16). https://doi.org/10.1109/ICIME.2009.16.
5. Leake, D., Birnbaum, L., Hammond, K., Marlow, C., & Yang, H. (1999). Task-based knowledge management. In *Proceedings of the AAAI-99 workshop on exploring synergies of knowledge management and case-based reasoning* (pp. 35–39). AAAI Press.
6. Sun, M., Chen, B., & Zhou, M. (2008). An ILP approach to mine the association rules on log ontology, International conference on apperceiving computing and intelligence analysis, Chengdu, China (pp. 274–278). https://doi.org/10.1109/ICACIA.2008.4770022.
7. Labrou, Y., & Finin, T. (1999). Yahoo! As ontology: using Yahoo! Categories to describe documents. In Proceedings of the eighth international conference on information and knowledge management (pp. 180–187). ISBN: 978-1-58113-146-8. https://doi.org/10.1145/319950.319976.
8. de Rezendel, J. L., Pereira, V. B., Xexeo, G, & de Souza, J. M. (2006). Building a personal knowledge recommendation system using agents, learning ontologies and web mining. IEEE international conference on apperceiving computing and intelligence analysis.
9. Chaffee, J., & Gauch, S. (2000). Personal ontologies for web navigation. In Proceedings of the ninth international conference on Information and knowledge management (pp. 227–234). https://doi.org/10.1145/354756.354823.
10. Liu, F., Yu, C., Meng, W., & Pitkow, J. (2002). Personalized web search by mapping user queries to categories. In Proceedings of the 11th international conference on information and knowledge management (pp. 558–565). https://doi.org/10.1145/584792.584884.
11. Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. In R. Meersman, Z. Tari, & S. Stevens (Eds.), *Database semantics. IFIP — The international federation for information processing* (Vol. 11). Springer.
12. Nagypál, G. (2005). Improving information retrieval effectiveness by using domain knowledge stored in ontologies. In *OTM confederated international conferences on the move to meaningful internet systems* (pp. 780–789). Springer.
13. Saeidlou, S., Saadat, M, & Jules, G (2014). Cloud manufacturing analysis based on ontology mapping and multi agent systems. IEEE international conference on systems, man, and cybernetics.
14. Thenmozhi, D., & Aravindan, C. (2005). User model for conceptual and personalized search. In Proceedings of the 4th international conference on computer recognition systems CORES'05 (pp. 295–302).
15. Liu, Y., Chen, X., & Sui, Z. (2007). Study on evolution of domain ontology. IEEE. https://doi.org/10.1109/ICICIC.2007.360.
16. Dumais, S., Joachims, T., Bharat, K., & Weigend, A. (2003). SIGIR workshop report: Implicit measures of user interests and preferences. *ACM SIGIR Forum, 37*, 50–54.
17. Chang, Y., Juang, T., Chang, C., & Yen, J. (2012). Integrating intelligent agent and ontology for services discovery on cloud environment. IEEE international conference on systems, man,

and cybernetics (SMC), Seoul, Korea (South), 2012 (pp. 3215–3220). https://doi.org/10.1109/ICSMC.2012.6378286.
18. Holden, A., & Freitas, A. A. (2004). *Web page classification with an ant colony algorithm, lecture notes in computer science* (Vol. 3242, pp. 1092–1102). Springer.
19. Trajkova, J., & Gauch, S. (2004). Improving ontology-based user profiles. In RIAO (Vol. 2004, pp. 380–390).
20. Rajganesh, N., & Ramkumar, T. (2016). A review on broker based cloud service model. *Journal of Computing and Information Technology, 24*(3), 283–292. https://doi.org/10.20532/cit.2016.1002778
21. Guarino, N., Masolo, C., & Vetere, G. (1999). OntoSeek: content-based access to the Web. *IEEE Intelligent Systems and their Applications, 14*(3), 70–80. https://doi.org/10.1109/5254.769887
22. Liu B., Abbas, H. A., & McKay, B. (2005). Classification rule discovery with ant colony optimization. IEEE/WIC international conference on intelligent agent technology (IAT'03) (p. 83). https://doi.org/10.1016/j.procs.2018.05.100.
23. The Open Directory Project (ODP). http://dmoz.org.
24. Ravindran, D. R. (2000). KeyConcept: Exploiting hierarchical relationships for conceptually indexed data. Doctoral dissertation, University of Kansas.
25. Pitkow. (2002). Personalized search. *Communications of the ACM, 45*(9), 50–55.

Chapter 20
Effective Deployment of Multi-cloud Customizable Chatbot Application for COVID-19 Datasets

T. Chellatamilan, N. Senthil Kumar, and B. Valarmathi

20.1 Introduction

With the prolific advancement in cloud-based systems and artificial intelligence, the world is witnessing a widespread utilization of messaging services and improve the implicit connection between human and machine. Nowadays, the Chatbots are gaining huge momentum in the fields of healthcare, travel, tourism, retail, etc. and made a remarkable impact on society. The messaging services at Chatbots have enabled people to improve their communication at ease, and it has been easily made possible through the appropriate inclusion of Natural Language Processing. As per the survey released in the year 2019 by Gartner [1], that on average, approximately around five billion users have proactively used the Cloud-Based Chatbot either directly or indirectly. Besides, many leading business entities have leveraged the hassle-free platforms to ensure cutting-edge advantage to its customers to augment their messaging services and thereby largely cutting out their cost by around 30–35%. In the recent World Economic Forum 2020, it was promulgated that the utilization of Artificial Intelligence in research would gain huge momentum and would be very crucial in the coming decades. Therefore, appropriate inclusion of Artificial Intelligence in Message Services (Chatbots) would largely replicate human-like conversations and would connect the customers through interoperable devices such as mobile phones, tablets, laptops, and other wireless connectivity handhelds. In the recent survey released by Gartner, 2019 [1], the appropriate blend of Artificial Intelligence with Chatbot has largely decreased their business operational costs and thereby witnessed a huge surge in their cost-saving to their business. There have

T. Chellatamilan (✉) · N. S. Kumar · B. Valarmathi
School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India
e-mail: chellatamilan.t@vit.ac.in; senthilkumar.n@vit.ac.in; valarmathi.b@vit.ac.in

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1_20

been many leading players of Cloud-Based Chatbot in the market such as IBM, Amazon, Microsoft, Google, and others. They have even provided the stakeholders with appropriate APIs for easy deployment and development of Chatbot to effectively serve their business needs [2]. As stated in the Garnet Report, after the emergence of Cloud-Based Chatbot, there has been a huge surge for Chatbot services in the market and boost the business growth in all verticals. But there has been a tremendous challenge in selecting the suitable Cloud-Based Chatbot which fits into the business model exactly. The comparative analysis and empirical evaluation of finding suitable Cloud-Based Chatbot would be the toughest task in determining the modern business parameters such as end-user satisfaction, affordability, on-the-fly customization, and efficiency. Generally, the Chatbot is trained based on the application domain such as healthcare, travel, education, online shopping, etc. But those Chatbots were relied on human intervention to evaluate the produced outcomes, and most of the time, the results would be contrary and inaccurate which further leads to time-consuming and labor-intensive tasks for the business partners. Although the prevalent utilization of Cloud-Based Chatbot is soaring high on the one end of the markets, there has been a huge threat lingering on the network for a long time. There have been around 3000 Chatbots released in the service markets since 2016, the risk of security is the lowest priority in their models, and it has given the room for high breach of trust on their networks. For example, the most popular Ticketmaster website has faced the biggest problem of a data breach in the year 2017, and it has released the official report that the malicious bot infiltrated their Chatbot services and providing the important information of the customers to the third party. Likewise, Delta Airlines reported that a malicious bot had penetrated the messaging software and hacked their customer's payment data as well as personal information.

20.2 Related Works

The researchers have used various metrics and evaluation strategies to gauge the performance of the Cloud-Based Chatbot. Earlier, the researchers have used the four comparative testings such as atomic match of the words or phrases or sentences, first-word match of the phrase or sentences, significant match of everything, and finally, no match relevantly identified. The ultimate objective of that comparative analysis was to find the suitable measures to gain the possible insights from the user values and tend to find the ways to increase the general capability of the system to obtain the suitable answers for the general query issued by the end users. In this connection, many Chatbot systems were developed and deployed at various applications and paved the way for easier communication between the business partners and potential customers at an affordable cost. Earlier implementations of Chatbot used in the domain areas such as healthcare and travel were initially considered as baseline systems for the newly designed Chatbot and further paved the way for implementing the robust cloud-based Chatbot in the future. In this section, we have listed and provided a detailed overview of the most prominent cloud-based Chatbot and its potential impacts on the decades of new development in the cloud-based messaging services.

ChatEval [3] is a web-based messaging service which provides a user-friendly dialog system for its end users to communicate easily and share the information mutually with the system without any serious glitches. Moreover, the ChatEval dialog system widely utilized the Natural Language Processing metrics such as semantic similarity, Jaccard similarity, and perplexity score for the ambiguous words to improve the performance of the interaction between the end user and dialog system.

Likewise, ECHO [4] web-based interface has been developed to provide the users the ability to access, compare, and evaluate the questions raised to the designed Chatbot. The ECHO Chatbot has certain additional advantages, that is, it provides the end users an option to choose the level of their interactions such as basic, medium, and hard level. Based on their choices and the type of conversation, the sequence of scenario-based questions will be raised to users and store the results in the cloud for further evaluation and comparison. Besides, it provides the potential users an API wrapper that connects the interface to the Cloud-Based Chatbot through an API call and acts as an interface to store the responses given to the users for evaluation. Upon storing the responses into the cloud, the answers were evaluated with stored results and display to the user with a precise answer. ECHO web-based interface achieved the accuracy rate at a considerable level initially.

Later in the year 2015, Bogdan Ionescu et al. [5] developed a novel chat-driven architecture to coordinate the information scattered in different group activities and collaborate the much-required text for further editing and consolidation. The fundamental aim of this architecture is to provide a hassle-free platform for consolidating the information scattered through different groups and garner the unified approach to disseminate the information synchronously. The core theme of this concept is operational transformation, i.e., coordinating and collaborating the potential information needed for the end users. This architecture was developed completely on a web-based solution to embed the information as well as the interactive functions to help the potential users to coordinate their task strenuously.

Subsequently, Haoliang Wang et al. [6] developed a most interactive web-based design for a social TV real-time chatting communication called Touch Talk. The sole objective of this application was to give the viewers a better understanding of the comments and opinions shared in real-time when many viewers had been viewing the programs telecasted on social TV. The authors had done some investigation on the user capabilities such as the viewer's behavior, attention rate, channel change rate, etc. Based on these grounding facts, they proposed a suitable model for the implementation of Touch Talk and bring a reasonable design to the application.

Besides, some of the widely used public cloud-based architecture for Chatbot have been given for understanding. The most popular and majorly used technologies, such as Ejabberd server, AWS lambda, API Gateway, and Chatbot, are discussed in Table 20.1.

Later the year 2017, many leading business companies have seen a huge spike in the utilization of Chatbot with their customers globally and locally. In particular, areas such as healthcare, hospitality, travel, telemarketing, customer service agents,

Table 20.1 Review of intelligent Chatbot and its potential applications

Public cloud-based architecture	Description	Language	Advantages
Ejabberd server	• Open source instant messaging server • Highly cross-platform, supports distributed environment, low fault-tolerant, and enables open standards to perform real-time interactions • Widely utilizes XMPP protocol on the back end	Erlang	• The efficient mobile reliability layer actively assists the mobile network's disconnection, rightly identifies the message deliveries, and a flawless conversation on any online device, and enables push notification and mobile interfaces • Ensures the message delivery with correct acknowledgment
AWS lambda	• Mostly event-driven, server less computing platform, guarantee that user can code their own application supported by AWS lambda	Node.Js, Java, and Python	• Use the AWS resources provided by lambda without provisioning or managing servers • One request per day to many requests per second; it affords the scalability and runs the code if only it is needed • Besides, user can run the code virtually on AWS lambda with any type of application
Amazon API gateway	• Affords the developers to create new applications, manage, and maintain it with high security	Java, Python	• API gateway enables the developers to provide the secure transactions to its end users and develop an affordable mobile and web application back ends at ease • Besides it offers the developers to connect mobile and web applications securely to business logic hosted on AWS lambda
Chatbot	• The Chatbot normally stands for chatter robot • It's a conversational program mostly deployed for chatting environment so that the user will never feel that they are interacting with the software or any computer	Java, Python. C++	Mostly it has been widely used in the areas like customer service, telemarketing, call centers, travel operations, and Internet gaming
ML lambda	• The machine learning lambda allows the programmers to experience the appropriate utilization of packages and technologies used in machine learning domain	Java, Python, R, etc.	• Through MLL, the coders can develop new machine learning frameworks and models and generates the appropriate predicates for any new design or applications • Research outcomes like find pattern generation, sequence modelling, similarity measures, and many more can be developed very easily

and finance services industries have witnessed a global reach and stand forefront with some of the promising companies such as General Motors (GM), Accenture, American Express, Salesforce, and the Dutch Airlines KLM. For instance, a Swedish bank has meticulously deployed an intelligent Chatbot in its local operation for effectively taking around 4000 conversation with 700 employees to understand the pertaining issues in the bank and resolved the emerging challenges related to Chatbot to provide a flawless service to its retail customers [7]. Examples like this witnessing the tremendous reception in the global arena and increasing adaptation of these Cloud-Based Chatbot in the future. As per the Juniper report released recently [8], an effective adaptation of intelligent Chatbot will gradually save around $11 billion to the companies in the year 2023.

20.3 Cloud Framework for Chatbot Design Tools and Use Cases

Build once and deploy everywhere is the predominant characteristic of cloud-based systems, whereas in this proposed system, we build the application once but that application could be accessed through the conversational interfaces enabled with windows, desktop, web, mobile, and Internet of Things (IoT) applications with natural and rich interactions. The general purpose Chatbots are designed and built using the frameworks listed below:

- Google's DialogFlow end-to-end Enterprise Editions
- Amazon's Amazon Lex
- Microsoft's Azure Bot Service
- IBM Watson Assistant
- RASA Tool kit

Chatbot or conversation bot is a computer program which will enable the human being to interact with a machine with a sequence of conversation called dialogs. The bot understands the user's request through the terms present in the query using natural language understanding (NLU) and natural language processing (NLP), and it generates the response to clarify or gratify those intents instantly. The back end system is also integrated with the Chatbot to make such an intelligent and instant decision-making system with the help of the internal or external API callouts, and it is shown in Fig. 20.1.

The intents or the user's request consists of different entities within it and further retrieves the intended response as action or dialog. In the age of information delivery, Chatbots are more preferable for timely delivery and reduced market cost. The serverless architecture framework provides the platform to upload the developed personalized code without considering the difficulty of the entire process. All over the world, 57% of industries have started to use the Chatbot for automating the interaction between the users and service. Traditionally, the Chatbots have been

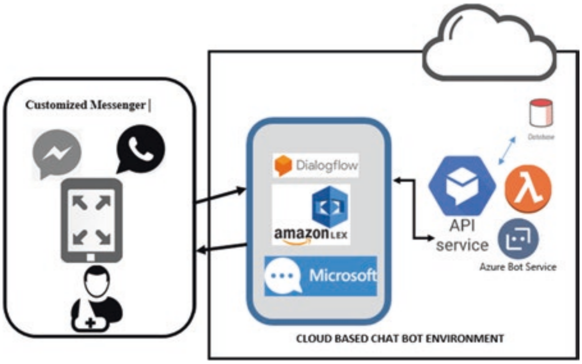


Fig. 20.1 The proposed cloud-based Chatbot environment

developed with the help of asynchronous API where the applications interacted through the message using the following sequence of steps:

(a) Develop and train the Chatbot using machine learning approach.
(b) Deploy the Chatbot on top of the API layer.
(c) Integrate a user interface with the help of mobile or web application or social media instance messaging applications.

It is very cumbersome to maintain the server if the Chatbots are launched with cyclic updates. The problem of maintenance and scalability could be simplified with the help of a cloud-based system that supports auto-scaling, reduced cost, and flexibility in integration.

AMAZON LEX: The voice and text-based Chatbots are being built and integrated into any conversational applications through the help of Amazon Lex. The natural language understanding (NLU) and the natural language processing (NLP) recognize the conversational text or speech and then extract the hidden content behind the conversation. The Q&A bots and information bots are very famous for consumer interest since these are capable of running on mobile devices, chat devices, and IoT devices. Lex is also powered like Amazon Alexa. The user input is passed to derive the potential content, and the output is returned for further processing. The Lambda function processes these requests and then triggers the cloud function as a fulfillment integration. High-quality machine learning, seamless deployment of the scale-up application, and built-in integration of the AWS platform are the fundamental benefits of Amazon Lex.

DIALOGFLOW: It is previously known as API.AI governed by the Google development framework helping us in building a text- or voice-based Chatbot as an end-to-end development suite. The conversational analytics are embedded over the bot powered by Auto machine learning and deep learning in recognizing the intent, named entities, and context-specific information about the conversation. The Dialog flow provides a webhook that will be fired by the Google cloud function when an instantaneous event occurs in the Chatbot application. It is also capable of connecting with any real-time database management system such as firebase for gathering the required information relevant to the intents of the user text.

IBM Watson Assistant: When any doubts or misunderstanding arises on the human side, most Chatbots help the human being to search for an answer for the questions and then direct the human being in the right way upon exchanging the conversational dialogs. IBM Watson Assistant [9] is an application that can run on any cloud-based environment and supports the users by bringing the auto artificial intelligence (AutoAI) to their data and applications where they are currently working with the system. The major components of the assistants are shown in Figs. 20.2 and 20.3, whereas Fig. 20.3 shows the sample set of intents, entity, and dialogs of a typical Chatbot application.

The IBM Watson Assistant's architectural components are given below:

SKILL: It is a main architectural component of the Watson built with atomic reusable computer program capable of representing a specific domain knowledge. The users will do the conversation with the skill to automate the task of making decisions.

INTENT: Users at first aim that the Chatbot anticipate from their requests during the conversation with certain skills. For example, a user's goal when conversing with a COVID-19 Chatbot is to get the details about the spreading ways of the virus, for example, #spread_ways, #new_cases, #new_death, and #symptoms.

ENTITIES: The set of terms or objects found in the intent utterances in turn provides the context for an intent. For example, an entity might be a city name which helps the routing core to determine which city is highly affected by COVID-19 in India for the spread of new cases.

CONTEXT: The information about the session and dialog conversation are maintained within the assistant to add more intelligence to our skill. All such information carried out throughout the session are represented by various types of context variables, namely, session context variable, built-in context variable or user defined shared context, or a type of skill context to track the frequent changes during the session.

DIALOG: The dialog in Watson defines how the Watson Chatbot is going to respond to the user's request upon recognizing the user's intents, entities, and contexts present in the query. The dialog is processed by the Watson Service on the basis of tree like structure (i.e., node by node access in the tree) and each

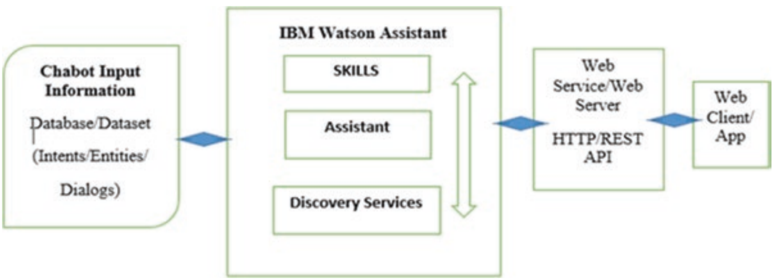


Fig. 20.2 IBM Watson Assistant architecture

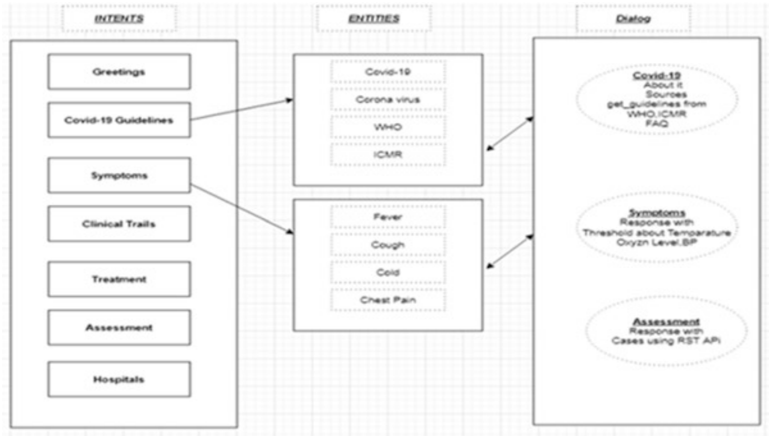


Fig. 20.3 Interaction among the components of the bot application

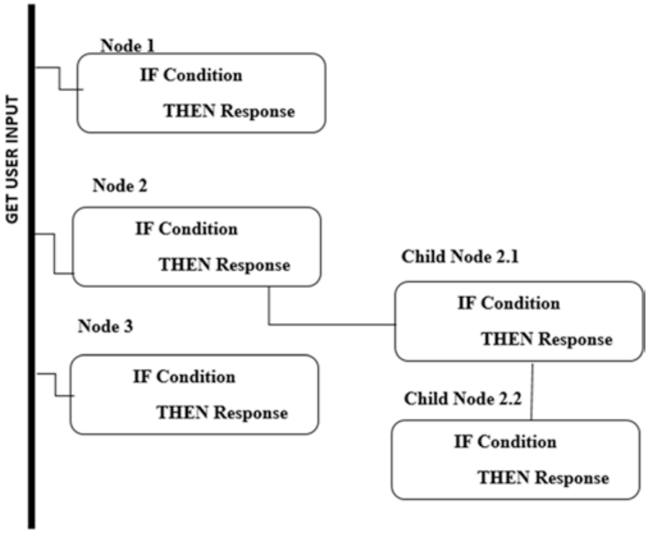


Fig. 20.4 Sample set of dialog nodes and its sequence of Fire

node in the tree represents the appropriate responce to be given for the respective occurrences of the intent, entity or the context.

As soon as the user enters the intents, the assistant triggers to process the intents with the help of sequences of nodes present in the dialog component of the Watson Assistant. As shown in Fig. 20.4, the assistant goes for recognizing the user input in node 1 initially. The node checks the condition using the current instantaneous contextual information and generates the response accordingly. If node1's condition is not met, the control will migrate to the next node in the sequence, and the process continues till it finds the suitable matching node for the given intents. The nodes have also been divided into sub-nodes according to the chain of sub-conditions that the assistants will apply to the child node.

SLOTS: To track the multiple pieces of information from users within the dialog nodes, the assistant adds the slots into the dialog nodes. The slots will be used to keep storing the frequent information within it, and the service asks for the details which are not available in the slots.

CALLING EXTERNAL WEB SERVICES/FUNCTIONS THROUGH API: The service defines the actions in terms of web functions that can programmatically call the external applications or services which pose the details of the intended intents and get back a result as a part of the processing that occurs within a dialog run. The schematic view of how such external functions were called by the dialog node has been visualized in Fig. 20.5 with serials of messages exchanged between all.

A Watson assistant can create any number of skill component to make conversation and to get several features during the conversation flow. This will enhance the mapping of any given intent into a suitable personalized response by single or multiple skill sets. The purpose of using the Watson discovery service is to search and determine such skillset across the web which is best suitable for getting the response of the intents and entities. Through the help of REST API as shown in Fig. 20.5, the actions will be taken to get the relevant response for the intent from the external sources of information. COVID-19 details have been already maintained by the web portal authorized by the World Health Organization and Government of India. This information could be accessed through the API by passing the required set of parameters, and then the web action will be sending back the response suitable.

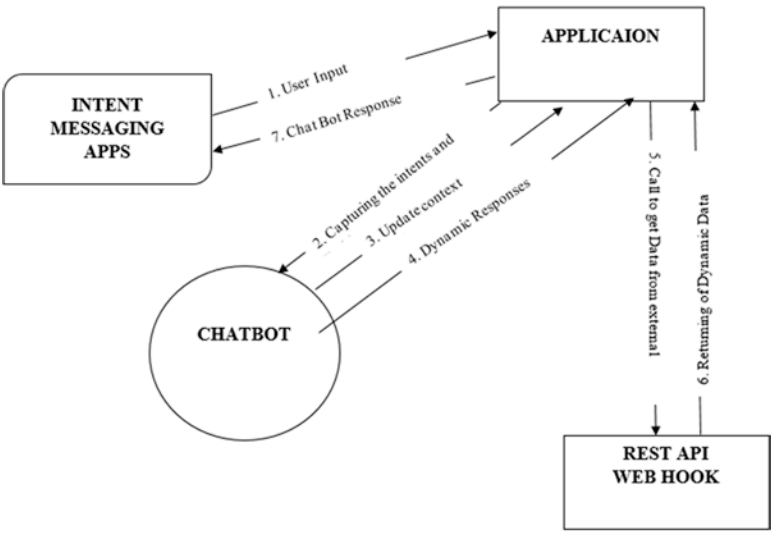


Fig. 20.5 Sample set of interaction among the components

20.4 Serverless Operational and Deployment Model

The execution of code in the serverless deployment model is only on-demand on a per request basis. One server process has been instantiated for per request demand to scale up inherently. This platform executes the code in response to the events. Also support multi-programming language to implement the actions like Node/JS, Python, PHP, Java and etc. The API Gateway support allows mapping the API end-points to IBM Cloud function action. It provides high-security features and easy socialization. The highly computationally intensive task which requires higher-end computational and storage resources could be outsourced to a powerful serverless platform even without changing the programming language.

The JSON schema describes the structure of the request and response bodies to an operation in such a way that any applications can be able to interact with the service irrespective of the communication protocol. Building applications using REST API is becoming a trend with the help of micro-services. For example, the COVID-19 micro-services [10] might provide a REST API with a single operation for accessing the information about COVID-19 confirmed cases and testing information for a supplied location or city. The IBM API Connect and IBM integration bus have been used for establishing the communication among the service consumer and service provider irrespective of the operating system, platform, and type of communication and transport protocols used at the application side.

20.4.1 Different Forms of Responses in Taking Actions for the Questions

The intents which are assumed to be raised by the public or patients [11] could be categorized into many types such as dialog, deflect, map, app navigation, information retrieval, and API. It will be very easy for the intent designer if we divide the complexity of representing it into the assistant with the help of this categorization as given in Table 20.2.

20.5 Sample Set of Intents

The sample set of intents designed for covering all the users' queries is given in Table 20.3 along with the intent title and its various ways of representing them.

Table 20.2 Sample set of intent categories

Intents	Type	Response
How do I know the details of COVID	DIALOG	Guide the user through set of steps to get the guidelines by WHO
I am heavily affected by COVID	DEFLECT	Transfer to medical agent
Where is the nearest COVID-care Center	MAP	Application launches map with directions
I need to buy medicine for COVID-19	APP NAVIGATION	Bring user to buy medicine by connecting medishop
Can I get the symptoms of COVID patient	INFORMATION RETIREVAL	Bring back an answer
How many new cases?	API	Bring the new cases count through web actions/functions

Table 20.3 Sample set of intents

Intents	Intents_examples
StayingSafeAtWork	How do I keep employees who interact with customers safe?
Spreadviafood	Does the virus stick to packaged food?
	Does it spread on food packaging?
	Can the virus that causes COVID-19 be spread through food, including
	Do I need to wash my groceries?
	Spread via food
NegativeTestResults	Can you have corona virus and still test negative?
	If you test negative can you still get COVID-19 later?
	Negative test results
	Does a negative test mean a person does not have COVID-19?
	Can a person test negative and later test positive for COVID-19?
WhyTheName	Why the name
	Why was COVID-19 used to name this disease?
	Is COVID an abbreviation?
	Why is covid-19 the name for this disease?
	Why is the disease being called coronavirus disease 2019, COVID-19
	Is corona virus the same as COVID-19?
RiskToChildren	Do children catch covid-19?
	Risk to children
	Are children at higher risk of getting sick?
	Are kids at increased risk of infection?
	Can corona virus infect an infant?
	What is the risk of my child becoming sick with COVID-19?
PrepareFamily	How can my family and I prepare for COVID-19?
	Should my household plan ahead?

Table 20.4 Sample categories of intents

Categories of FAQ
Speculation
Transmission
Nomenclature
Reporting
Societal response
Societal effects
Origin
Prevention
Treatment
Testing
Comparison
Economic effects
Symptoms
Having COVID
Individual response

20.5.1 Categorization of Intents

The frequently asked questions and answers were collected from the various authorized sources about the COVID-19. The questions are summarized with different categorizations as shown in Table 20.4.

The frequently asked such questions were categorized into different types as shown in Table 20.5 along with a sample set of questions.

20.6 Chatbot with Webhooks

In order to retrieve the latest relevant information and updates related to COVID-19, the Chatbot uses webhooks with the help of cloud action functions to connect and interact with the approved data sources where the authorized information about the disease is stored [12]. By making such external API callouts, the Chatbot leverage information bypassing the parameter which carries the information about the object of interest to the respective API. We can define the function and can choose the runtime environment which is suitable for the developer's option.

Webhooks: Webhook is an API web call back endpoint concept with a lightweight way of implementing the event-driven reaction. As shown in Fig. 20.6 [13], the webhook delivers information to other external applications that perform programmatic functions. When the webhook is associated with a dialog skill of an assistant, it triggers the action when an event occurs due to the interaction of the respective node of the dialog component of the Chatbot [14]. Mostly the webhooks will post data across the applications either in the format of JSON or in XML.

Table 20.5 Sample set of intents and its category

Sample question	Category
When do you think the world will stabilize after covid ends	Speculation
When covid will end	Speculation
When covid will stop	Speculation
If president trump opens the economy will we see a second wave of covid	Speculation
Is there a risk of a second wave of covid like the second wave of the Spanish flu which killed more people compared to the first wave	Speculation
How would the pandemic covid be different in America if trump wasnt president	Speculation
Will covid go away	Speculation
How long will social distancing for covid last	Speculation

Are pregnant women more susceptible to the covid virus and will it harm the fetus	Transmission
Can bats in United States spread covid back to people	Transmission
What is the risk of my child becoming sick with covid	Transmission
Should covid death be cremated	Transmission
Why are covid patients contagious for so long before they show the symptoms	Transmission
Are children as likely to get covid as adults	Transmission
Can i get covid from my pet or other animals	Transmission
Can people who recover from covid be infected again	Transmission
Are certain blood types more resistant to acquiring the covid when exposed than others	Transmission
How do you wash vegetables and fruits these days of the covid epidemic	Transmission
---	

Fig. 20.6 Polling versus webhooks

It is not required to keep on polling to check the occurrences of the event; rather webhook monitors the event and generates the notification to the user through the POST request sent to the callback URL that was specified during the configuration of the webhook registrations. For example, whenever a file or database record is modified or updated that time the registered webhook call back generates the notification and sends the required information to the assistant/Chatbot.

Table 20.6 Sample code for webhook API

```
if (params.state) {
// Based on the state name passed in the parameter,the total cases will be retrieve
state = getStateCode(params.location);
const uri = "https://localhost:3000/state?postalKey=${key[state]}&format=json&apiKe
const data = await request({
            method: "GET",
            uri: uri,
            json: true,
          });

  return {
      result: `Total Cases: ${data.covid19.confirmed[0]}\nTotal Deaths: ${data.cov
        };
       }
```

Sample code which collects the responsive information with the help of webhook from the API is given in Table 20.6.

20.7 Multi-cloud Environment for Deploying the Chatbot

Multi-cloud environments have been constructed on top of the container platform which in turn provide consistent transparent access, governance, and automation of provision of services to the consumers over the edge. The major benefit of multi-cloud infrastructure [15] is to reduce the risks and deliver more automated services through the unified management platform running top of containers. In addition to that the multi-cloud also offers event-driven infrastructural services to the subscribed users with the multi-cluster facility. If we deploy the Chatbot in the cloud environment, then it will be simplifying the process of integrating the resources required for retrieving the relevant information through the Chatbot. The multi-cloud platform enhances hazard-free integration and avoids continuous in-house maintenance. In some situations, the customer's front-end application is not capable of interacting with the respective API service for gathering sufficient information. At that point in time, the multi-cloud interconnects the seamless access of the services to facilitate the user's request upon retrieving the responsible information across multiple clouds. Usually, the on-premise environment need not have the provision of integrating all such data sources directly whereas the multi-cloud goes for solving this issue. The multi-cloud architecture delivers more flexibility, scale-up, speed up, and over trust environment for self-service and guided analytics such as mobile analytics and embedded analytics over the data sources from multiple channels of data sources. Through multiple cloud service providers, whoever provides similar kind of services could be connected through the multi-cloud environment for exchanging the services among them. Multi-cloud refers to the consumption of several cloud services using multiple hybrid clouds irrespective of the deployment models used such as private and public. As shown in Fig. 20.7 rather than using one

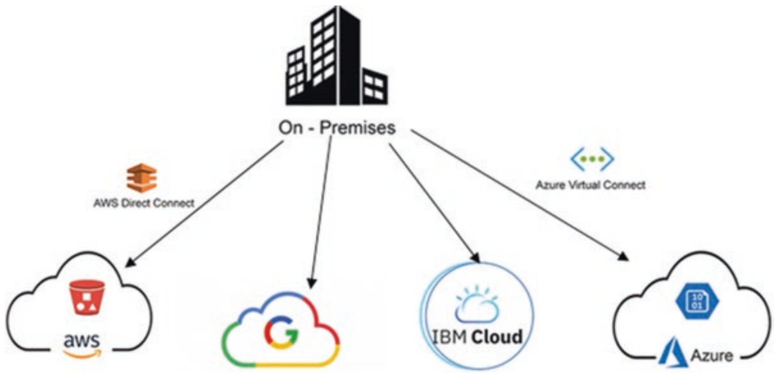


Fig. 20.7 Multi-cloud interactions

simple cloud, the business scenario can be offered with services by multiple providers to reach the objectives of the business requirement.

The business industries need not depend completely on a single vendor and able to get the services from any service provider by reducing the cost and leveraging the flexibility over the deployment. The top three business drivers to establish the cloud are sourcing, architecture, and governance. The multi-cloud platform provides a more strategic solution in deciding the backup recovery and workload protection with load balancing. The seamless migration is possible between different multi-cloud service providers without disturbing the existing operations of the running services.

20.8 Experimentation

Initially, the information about COVID-19 is collected from the authorized sources such as CDC (Centre for Disease Control and Prevention) [16], ICMR (Indian Council of Medical Research), WHO (World Health Organization), etc. Then all such information is structured in a format suitable for converting them into the form of intents, entities, and dialogs. The easiest format for integrating and representing all such information is through JSON. The frequently asked such questions and answers collected from the guidelines of the authorized sources were formulated as a JSON file [17]. Rather than feeding each of the intents, entities, and dialogs individually into the assistant, the Watson tool will automatically transform the intents, entities, and dialogs stored in the JSON to the Watson assistant in a bulk mode. It is required to identify the various ways by which we can gather the dynamic information such as total confirmed cases, the total number of death, and the total number of the test taken so for with respect to the regional location. This challenging task has been achieved with the help of remote REST API designed and supplied by the authorized API portal. Through the Webhook facility in Watson Assistant, we are

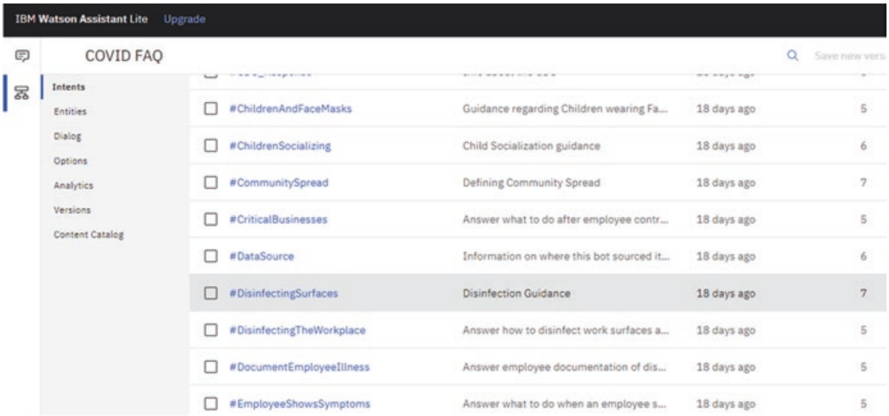


Fig. 20.8 Sample set of intents designed in Watson

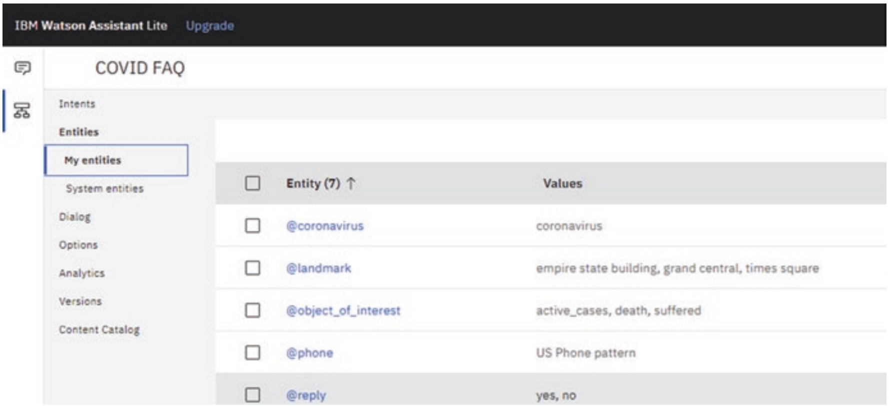


Fig. 20.9 Sample set of entities designed in Watson

able to establish a connection with the API callout and can collect the information about the updates on COVID-19. The screenshot of the design of the assistant dialog skill is visualized in Fig. 20.8.

The entities created for the above intents were visualized in Fig. 20.9. The symbol @ is being used for differentiating the terms whether it refers to intents or entities. For intents the symbol # was used; for entity the symbol @ was used in the design time of the assistant dialog skill.

The dialog which is used for the conversation between the Chatbot and the user is shown in Fig. 20.10. The dialog node actually connects the intents/entites to the respective response generated by the bot.

The deployed Chatbot assistant can be integrated to any web application, social media applications, or messenger application. The sample screen shot of the Chatbot is shown in Fig. 20.11.

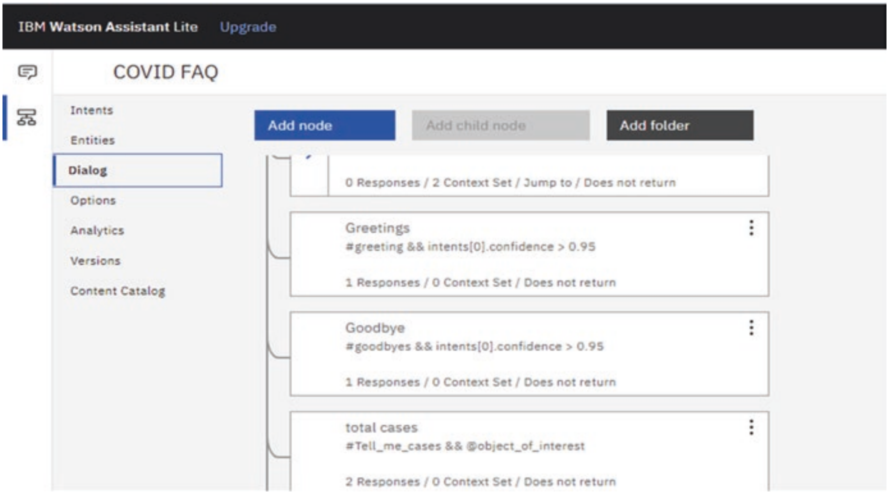


Fig. 20.10 Sample set of dialog nodes designed in Watson

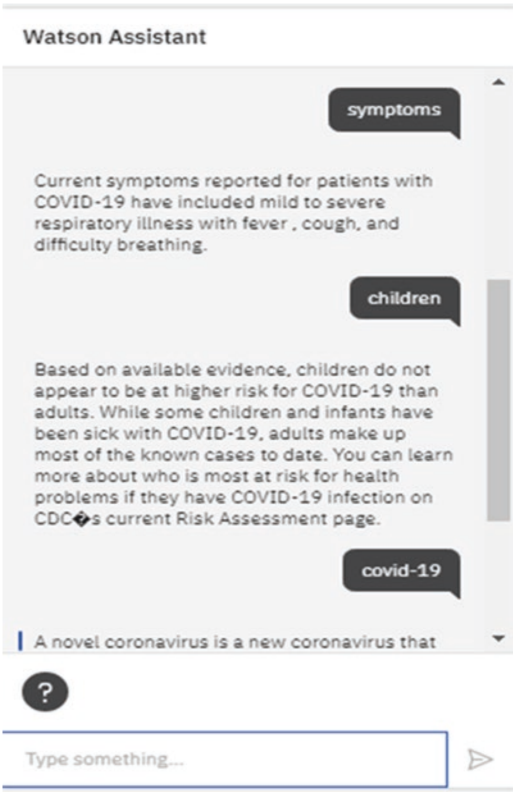


Fig. 20.11 Sample screenshot of the deployed assistant Chatbot

20.9 Conclusion

The AI-powered Chatbot integrated system can outperform well for doing the repeated and cyclic responsive reply in an efficient and intellectual manner. The healthcare organization can consider utilizing the experience of the patients and provides the best services to its consumers. Chatbot leverages the technologies of AutoAI and AutoML, including intent classification, named entity recognition, factoid, and recurrent question answering with dialogue management. Currently, this model is supporting only one language (English); in the future the model can be enhanced to support a multilingual platform so that multi-lingual patients could get the benefit of the bot. As part of future work, a knowledge-based expert system can be built for supporting the Chatbot assistant, capable of interpreting and responding to the answers in terms of biological information.

References

1. https://www.gartner.com/smarterwithgartner/top-trends-on-the-gartner-hype-cycle-for-artificial-intelligence-2019/.
2. https://documenter.getpostman.com/view/10724784/SzYXXKmA?version=latest; https://api.covid19india.org/ COVID19-IndiaAPI.
3. Sedoc, J., Ippolito, D., Kirubarajan, A., Thirani, J., Ungar, L., & Callison-Burch, C. (2019, June). Chateval: A tool for chatbot evaluation. In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations) (pp. 60–65).
4. Forkan, A. R. M., Jayaraman, P. P., Kang, Y. B., & Morshed, A. (2020, May). ECHO: A tool for empirical evaluation cloud Chatbot. In 2020 20th IEEE/ACM international symposium on cluster, cloud and internet computing (CCGRID) (pp. 669–672). IEEE.
5. Ionescu, B., Gadea, C., Solomon, B., Trifan, M., Ionescu, D., & Stoicu-Tivadar, V. (2015, May). A chat-centric collaborative environment for web-based real-time collaboration. In 2015 IEEE 10th jubilee international symposium on applied computational intelligence and informatics (pp. 105–110). IEEE.
6. Wang, H., Zhang, C., & Li, M.. Social TV real-time chatting application design. In International symposium on wireless personal multimedia communications (WPMC).
7. Bharti, U., Bajaj, D., Batra, H., Lalit, S., Lalit, S., & Gangwani, A. (2020, June). Medbot: Conversational artificial intelligence powered chatbot for delivering tele-health after covid-19. In 2020 5th international conference on communication and electronics systems (ICCES) (pp. 870–875). IEEE.
8. https://www.juniperresearch.com/press/press-releases/chatbot-interactions-retail-reach-22-billion-2023.
9. Ahmed, M. N., Toor, A. S., O'Neil, K., & Friedland, D. (2017). Cognitive computing and the future of health care cognitive computing and the future of healthcare: The cognitive power of IBM Watson has the potential to transform global personalized medicine. *IEEE Pulse, 8*, 4–9. https://doi.org/10.1109/MPUL.2017.2678098
10. Montenegro, J. L. Z., da Costa, C. A., & da Rosa, R. R. (2019). Survey of conversational agents in health. *Expert Systems with Applications, 129*, 56–67. https://doi.org/10.1016/J.ESWA.2019.03.054

11. Madeira, R. N., Pereira, C. M., Clipei, S., & Macedo, P. (2018). ONParkinson – Innovative mHealth to support the triad: Patient, carer and health professional. In *Pervasive computing paradigms for mental health* (pp. 10–18). Springer. https://doi.org/10.1007/978-3-319-74935-8_2
12. Zhao, R., Romero, O. J., & Rudnicky, A. (2018). Sogo: A social intelligent negotiation dialogue system. In *Proceedings of the 18th international conference on intelligent virtual agents, ser. IVA '18* (pp. 239–246). ACM. [Online]. http://doi.acm.org/10.1145/3267851.3267880
13. What is a Webhook & how to use it to track email marketing activity. https://www.sendinblue.com/blog/what-is-a-webhook/.
14. Oh, K., Lee, D., Ko, B., & Choi, H.-J. (2017). A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. 18th IEEE international conference on mobile data management (MDM) (pp. 371–375).
15. D. Jurafsky and J. Martin (2018) Dialog systems and chatbot, speech and language processing.
16. National Canter for Immunization and Respiratory Diseases (NCIRD). Division of viral diseases. https://www.cdc.gov/coronavirus/2019-ncov/faq.html#Coronavirus-Disease-2019-Basics.
17. Bapat, R., Kucherbaev, P., & Bozzon, A. (2018). Effective crowdsourced generation of training data for Chatbot natural language understanding. In Web engineering.

Index

R. Nagarajan et al. (eds.), *Operationalizing Multi-Cloud Environments*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-74402-1

D

G

H

I