QoS-aware Autonomic Cloud Computing for ICT

Sukhpal Singh and Inderveer Chana

Abstract Emergence of Information and Communication Technologies (ICT) plays an important role in networking sector by providing services through cloud-based systems. Based on application requirements of cloud users, discovery and allocation of best workload-resource pair is an optimization problem. Acceptable Quality of Service (QoS) cannot be provided to the cloud users until provisioning of resources is offered as a crucial ability. QoS parameters based resource provisioning technique is therefore required for efficient scheduling of resources. In this paper, QoS-aware autonomic resource provisioning and scheduling for cloud computing technique has been proposed. The proposed technique caters to provisioned resource distribution and scheduling of resources. The performance of the proposed technique has been evaluated through Cloud environment. The experimental results show that the proposed technique gives better results in terms of execution cost and execution time of different Cloud workloads.

Keywords Resource provisioning · Resource scheduling · Quality of service · Autonomic cloud · Information and communication technologies

1 Introduction

Cloud Computing enables resources (Infrastructure, Platform or Software) to be offered as services for Information and Communication Technologies (ICT). These resources are provided using a pay-as-you-use pricing plan [1]. To satisfy the request of customers, ICT-based service must be provided in accordance with required level of QoS. However, providing dedicated ICT-based Cloud services that

S. Singh (✉) · I. Chana
Computer Science and Engineering Department, Thapar University,
Patiala 147004, Punjab, India
e-mail: ssgill@thapar.edu

I. Chana
e-mail: inderveer@thapar.edu

S.C. Satapathy et al. (eds.), *Proceedings of International Conference on ICT for Sustainable Development*, Advances in Intelligent Systems and Computing 409, DOI 10.1007/978-981-10-0135-2_55

ensure user's dynamic QoS requirements and avoid Service Level Agreement (SLA) violations, a big challenge in Cloud Computing. Currently, Cloud services are provisioned and scheduled according to resources' availability without ensuring the expected performances [2]. To realize this, there is a need to consider important aspect of ICT-based service which reflects the complexity introduced by the Cloud management: QoS-aware autonomic management of ICT-based Cloud services [3]. QoS-aware aspect involves the capacity of the ICT-based service to be aware of its behavior to ensure the minimum execution time and cost.

Other reason of increase in execution cost is resources are running in idle or underutilization state. Efficient resource scheduling in Cloud is a challenging job and the scheduling of appropriate resources to Cloud workloads depends on the QoS requirements of Cloud applications [3]. Execution Time and execution cost in case of heterogeneous cloud workloads is very difficult to improve. Therefore, there is need of Cloud-based framework which schedules computing resources automatically by considering execution time and execution cost as a QoS parameter to provide efficient ICT services. Autonomic resource provisioning and scheduling is an ability to reduce cost and time in ICT-based autonomic systems which are self-optimizing. This research work focuses on one of the important aspects (QoS parameters) of self-optimization, i.e., execution time and execution cost.

The motivation of this paper is to design a cloud-based QoS-aware autonomic resource provisioning and scheduling technique for effective scheduling of resources for ICT-based services which considers execution time and cost as important QoS parameters. The main aim of this research work is: (i) to propose an autonomic resource provisioning and scheduling technique for execution of heterogeneous workloads for effective ICT services, (ii) to optimize the QoS parameters such as execution time and cost, and (iii) to implement and perform evaluation with existing work. Paper is structured as follows: Sect. 2 presents related work and contributions. Proposed technique is presented in Sect. 3. Section 4 describes the experimental setup used for performance evaluation and results. Section 5 presents conclusions and future directions.

2 Related Work

This section is presenting the related work of resource provisioning and scheduling in brief.

2.1 Resource Provisioning

Kuan et al. [4] proposed QoS-based autonomous SLA model of violation-filtering for IaaS and PaaS applying a SLA appraising many ways and penalty architecture and presenting a resource provisioning mechanism to manage resource efficiently

without specifying the QoS parameters. Linlin et al. [5] proposed SLA-based provisioning technique to reduce resource price and SLA deviations. The management of customer requests, and mapping them with resources is defined along with the supervision of different types of workloads by considering QoS such as execution time. Qiang et al. [6] presented a virtual environment-based framework for better supervision of infrastructure which offers separation among workloads running concurrently with similar resources and sanctions dynamic horizontal and vertical scalability to realize Service Level Objectives. Trieuet et al. [7] proposed a framework for dynamic workloads based on threshold number of dynamic periods for dynamically assigning and quickly provisioning of virtual resources to users based on their QoS requirements. Nikolas et al. [8] described self-adaptive resource provisioning approach to find the appropriate predicting ways for a given perspective through the use of decision tree. This approach considers QoS parameters like relative error and SLA violation, and minimizes both the parameters but does not consider execution time and execution cost.

2.2 Resource Scheduling

Pandey et al. [9] presented a Particle Swarm Optimization-based heuristic technique to schedule the applications to Cloud resources that proceeds both computation and data transmission cost. Topcuoglu et al. [10] presented the HEFT algorithm to discover the average execution time of each workload and also the average communication time among the resources of two workloads. Wu et al. [11] suggested a Market Oriented Hierarchical Scheduling approach which contains of both service level scheduling and workload level scheduling. The service level scheduling deals with the Task to Service assignment and the workload level scheduling deals with the optimization of the Task to Virtual Machine assignment in local Cloud data centers. Yu et al. [12] proposed a Cost-Based Workflow Scheduling algorithm that reduces the execution cost, however, meeting the deadline for delivering results. Varalakshmi et al. [13] described an Optimal Workflow-based Scheduling framework to discover a solution that tries to meet the user-desired QoS constraints, i.e. execution time. QoS-aware autonomic resource provisioning and scheduling technique needs to consider the basic features of Cloud computing for ICT in order to execute the heterogeneous Cloud workloads automatically with minimum execution time and execution cost, which is not considered in above discussed existing work.

3 QoS-aware Autonomic Resource Provisioning and Scheduling Technique for ICT-Based Services

Provisioning and scheduling of resources in cloud is an important part of resource management system. Mapping of cloud workloads to appropriate resources is mandatory to improve QoS parameters like execution time and execution cost etc.

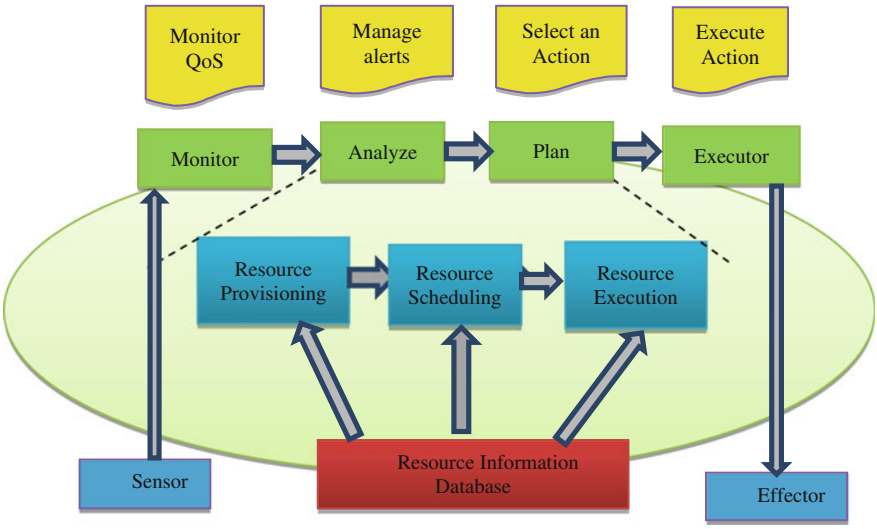

Fig. 1 Architecture of QoS-aware autonomic resource provisioning and scheduling technique

[14]. Based on QoS requirements, scheduling finds and maps the resources and workloads. Resource provisioning and scheduling in Cloud is done in the following steps [2, 3]: (i) understand the expectations and requirements of Cloud user, (ii) analyze and cluster the workloads through machine learning algorithm, i.e., K-Means based Clustering Algorithm, (iii) find the required number of resources, (iv) map the resources and workloads, and (v) schedule and execute the workloads on appropriate resources with minimum time and cost. This scheduling framework executes the workloads without self-optimization. But in the present scenario, there is need of Cloud-based technique for ICT which provisions and schedules computing resources automatically by considering execution time and execution cost as a QoS parameter. In this research paper, we focused on these two parameters and automated the existing framework [2, 3]. In this paper, we have extended the existing work by considering two important QoS parameters through automation for ICT-based service. Architecture of QoS-aware autonomic resource provisioning and scheduling technique is shown in Fig. 1. This technique is based on IBM's autonomic model [15] that considers four steps of autonomic system: (1) Monitor, (2) Analyze, (3) Plan, and (4) Execute.

3.1 Monitor [M]

Initially, *Monitor* is used to collect the information from sensors (*Sensors* (Wireless Sensor Network) get the information about execution time and cost of all the systems working under ICT based Cloud environment and update the information

time to time) for monitoring continuously the value of execution time and execution cost and transfer this information to next module for further analysis.

3.2 *Analysis and Plan [AP]*

Analyze and Plan module start analyzing the information received from monitoring module and make a plan for adequate actions for corresponding alert. In this step, based on QoS requirements of workload(s), resources are provisioned, scheduled, and executed.

3.2.1 Resource Provisioning

Monitor continually checks the status of resources provisioned, workloads queued, and SLA deviation. The objective of resource provisionor is to provision the resources to Cloud consumer without violation of SLA. The workloads submitted should be executed within their budget and deadline. Proposed ICT-based technique provisions and schedules the resources based on time and cost to the workloads automatically is shown in Fig. 2. Workload submitted by user to resource provisionor is stored into bulk of workloads for their execution. All the submitted workloads are analyzed based on their QoS requirements described in terms of SLA. Workload patterns are identified for better classification of workloads, then pattern-based clustering of workloads is done. QoS metrics for every QoS requirement of each workload are identified [2]. Based on importance of the attribute, weights for every cloud workload are calculated. After that, workloads are re-clustered based on K-Means based clustering algorithm for better execution. Calculate the value of

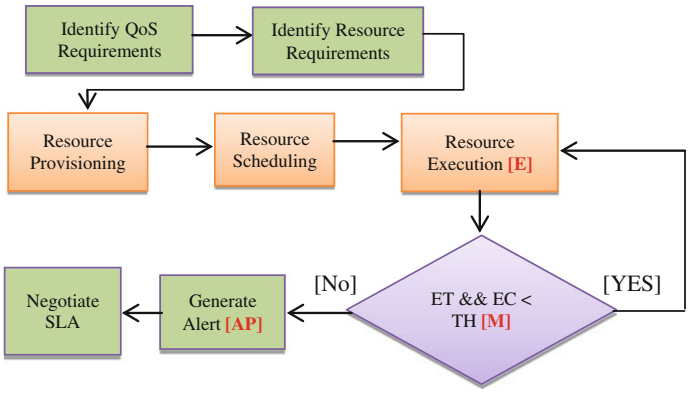

Fig. 2 Automatic execution of QoS-aware autonomic resource provisioning and scheduling technique

Execution time and Execution cost. If the value of workloads executes within deadline and budget [Execution Time (ET) and Execution Cost (EC) is lesser than Threshold Value (TH)] then it will provision resources otherwise generate alert for analyzing the workload again after resubmission of SLA by Cloud consumer.

3.2.2 Resource Scheduling

After successful provisioning of resources, Resource Scheduler (RS) takes the information from the appropriate workload after analyzing the various workload details which Cloud consumer demanded [3]. Decision tree is used to select the particular scheduling policy based on consumer workload details [3]. RS then collect the information available resources from Resource Information Database (RID). RID contains details of all the resources available in resource pool and reserve resource pool. Based on Cloud consumer details RS assigns resources and executes Cloud workloads.

3.2.3 Resource Execution

During execution of a particular Cloud workload, the Resource Executor (RE) will check the current workload. If the resources are sufficient for execution then it will continue with execution otherwise request for more resources. RE will check policy conditions and cost and time. If the Execution time and Execution cost is lesser than threshold value then RE will execute workloads, otherwise RE will generate alert. After successful execution of Cloud workloads, RE releases the free resources to resource pool and RE is ready for execution of new Cloud workloads.

3.3 Executor [E]

Executor implements the Plan after analyzing completely. The main objective of *executor* is to reduce the Execution time and Execution cost. Based on the output given by analysis and executor tracks, the new workload submission and resource addition generates the alert. *Effector* is used to transfer the new policies, rules, and alerts to other nodes with updated information.

4 Experimental Setup and Results

Tools used for setting Cloud environment are Microsoft Visual Studio, NetBeans IDE 7.1.2, CloudSim, IntegratedNETJavaWeb, and SQL Server. Microsoft Visual Studio 2010 is an Integrated Development Environment from

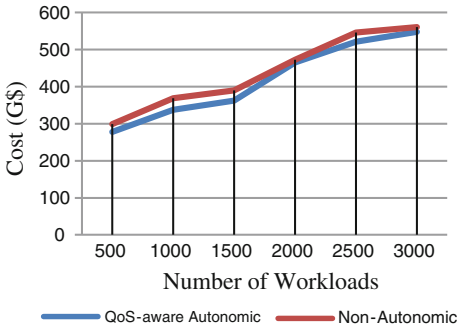

Fig. 3 Number of workloads versus cost

Microsoft. Cloud user interacts with ICT-based autonomic framework through Cloud Workload Management Portal (CWMP) to submit the workload details. User information, workload detail, and resource detail are stored in database through SQL Server. Cloud workload management portal is implemented in .NET framework and framework is running in Microsoft Visual Studio. We have explained the description of simulation environment in our previous work [3].

Figure 3 shows the cost of different number of workloads (500–3000) with QoS-aware autonomic resource provisioning and scheduling technique (QoS-aware Autonomic) and non-autonomic technique. Non-QoS based resource scheduling technique used for experimental evaluation in this paper has been designed by combining two traditional resource scheduling algorithms (First Come First Serve FCFS and Round Robin), in which resources are scheduled without considering QoS parameters. Cost is increasing with increase in number of workloads but QoS-aware autonomic resource provisioning and scheduling technique performs better. Cost of different number of resources (50–250) of QoS-aware autonomic resource provisioning and scheduling technique (QoS-aware Autonomic) is compared with non-autonomic technique as shown in Fig. 4. Cost is decreasing with increase in number of resources and result shows the QoS-aware autonomic technique executes the same number of Cloud workloads at a lesser cost.

Figure 5 shows the execution time of different number of workloads with QoS-aware autonomic resource provisioning and scheduling technique (QoS-aware

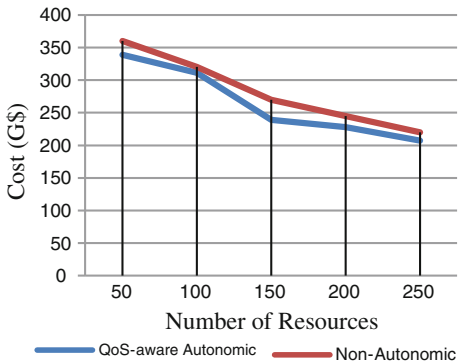

Fig. 4 Number of resources versus cost

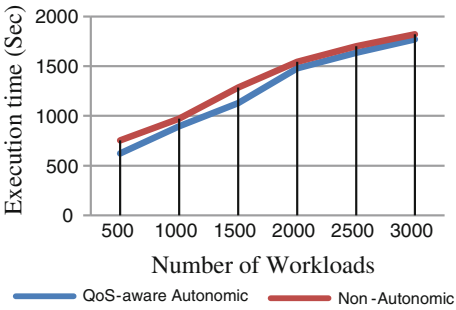

Fig. 5 Number of workloads versus execution time

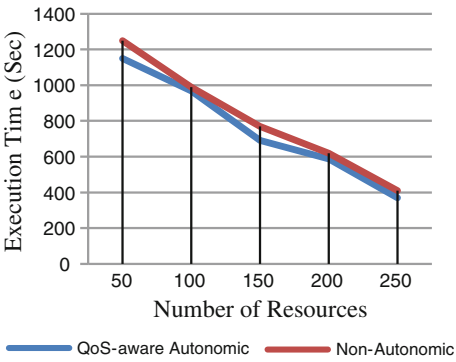

Fig. 6 Number of resources versus execution time

Autonomic) and non-autonomic technique. Execution Time is increasing with increase in number of workloads, but QoS-aware autonomic resource provisioning and scheduling technique performs better. Execution Time of different number of resources of QoS-aware autonomic resource provisioning and scheduling technique (QoS-aware Autonomic) is compared with non-autonomic technique as shown in Fig. 6. Cost is decreasing with increase in number of resources and result shows the QoS-aware autonomic technique that executes the same number of Cloud workloads at a lesser Execution Time.

5 Conclusions and Future Directions

In this paper, QoS-aware autonomic resource provisioning and scheduling technique for ICT-based services has been presented and this technique has been validated in Cloud environment and the experimental results perform better in terms of cost and execution time. The proposed ICT-based autonomic resource provisioning and scheduling technique considers heterogeneous workload for resource scheduling and uses the autonomic model to improve cost and time. This framework considers only two QoS parameters of self-optimization. Further, this technique can be extended by incorporating other QoS parameters like reliability, availability, energy, etc.

Acknowledgments One of the authors, Sukhpal Singh (SRF-Professional), gratefully acknowledges the Department of Science and Technology (DST), Government of India, for awarding him the INSPIRE (Innovation in Science Pursuit for Inspired Research) Fellowship (Registration/IVR Number: 201400000761 [DST/INSPIRE/03/2014/000359]) to carry out this research work. We would like to thank Dr. Maninder Singh for his valuable suggestions.

References

1. Singh, S., & Chana, I. (2012). Cloud based development issues: A methodical analysis. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, *2*(1), 73–84.
2. Singh, S., & Chana, I. (2015). Q-aware: Quality of service based cloud resource provisioning. *Computers and Electrical Engineering—Journal—Elsevier*. (http://dx.doi.org/10.1016/j.compeleceng.2015.02.003).
3. Singh, S., & Chana, I. (2015). QRSF: QoS-aware resource scheduling framework in cloud computing. *The Journal of Supercomputing, 71*(1), 241–292.
4. Lua, K., Yahyapoura, R., Wiedera, P., Yaquba, E., & Jehangiria, A. I. (2013). QoS-based resource allocation framework for multidomain SLA management in clouds. *International Journal of Cloud Computing 1*,(1). (ISSN 2326-7550).
5. Wu, L., Garg, S. K., & Buyya, R. (2011). Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)* (pp. 195–204).
6. Li, Q., Hao, Q., Xiao, L., & Li, Z. (2009) Adaptive management of virtualized resources in cloud computing using feedback control. In *1st International Conference on Information Science and Engineering (ICISE)* (pp. 99–102).
7. Chieu, T. C., Mohindra, A., Karve, A. A., & Segal, A.(2009). Dynamic scaling of web applications in a virtualized cloud computing environment. In *IEEE International Conference on e-Business Engineering, ICEBE'09* (pp. 281–286).
8. Herbst, N. R., Huber, N., Kounev, S., & Amrehn, E. (2014). Self-adaptive workload classification and forecasting for proactive resource provisioning. *Concurrency and Computation: Practice and Experience 26*(12), 2053–2078.
9. Pandey, S., Wu, L., Guru, S., & Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*.
10. Topcuoglu, H., Hariri, S., & Wu, M.-Y. (1999). Task scheduling algorithms for heterogeneous processors. In *Heterogeneous Computing Workshop, (HCW'99)*.
11. Wu, Z., Liu, X., Ni, Z., Yuan, D., & Yang, Y. (2013). A market-oriented hierarchical scheduling strategy in cloud workflow systems. *The Journal of Supercomputing, 63*(1), 256–293.
12. Yu, J., Buyya, R., & Tham, C. K. (2005). Cost-based scheduling of scientific workflow applications on utility grids. In *Proceeding of IEEE e-Science and Grid Computing*.
13. Varalakshmi, P., Ramaswamy, A., Balasubramanian, A., & Vijaykumar, P. (2011). An optimal workflow based scheduling and resource allocation in Cloud. In *Advances in Computing and Communications* (pp. 411–420). Berlin: Springer.
14. Chana, I., & Singh, S. (2014). Quality of service and service level agreements for cloud environments: Issues and challenges. In *Cloud Computing* (pp. 51–72). New York: Springer.
15. Kephart, J. O., & Walsh, W. E. (2003). An architectural blueprint for autonomic computing. Technical Report, IBM Corporation (2003), 1–29, IBM. Retrieved December 25, 2014 from http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf.