Computers, Materials & Continua
DOI:10.32604/cmc.2021.013865
Article

Tech Science Press

Intelligent Cloud Based Load Balancing System Empowered with Fuzzy Logic

Atif Ishaq Khan[1], Syed Asad Raza Kazmi[1], Ayesha Atta[1,*], Muhammad Faheem Mushtaq[2], Muhammad Idrees[3], Ilyas Fakir[1], Muhammad Safyan[1], Muhammad Adnan Khan[4] and Awais Qasim[1]

[1]Department of Computer Science, Government College University, Lahore, 54000, Pakistan
[2]Department of Information Technology, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, 64200, Pakistan
[3]Department of Computer Science and Engineering, University of Engineering and Technology Lahore, Narowal Campus, 51600, Pakistan
[4]Department of Computer Science, Riphah International University Lahore Campus, Lahore, 54000, Pakistan
[*]Corresponding Author: Ayesha Atta. Email: ayesha.atta@gcu.edu.pk
Received: 25 August 2020; Accepted: 01 November 2020

Abstract: Cloud computing is seeking attention as a new computing paradigm to handle operations more efficiently and cost-effectively. Cloud computing uses dynamic resource provisioning and de-provisioning in a virtualized environment. The load on the cloud data centers is growing day by day due to the rapid growth in cloud computing demand. Elasticity in cloud computing is one of the fundamental properties, and elastic load balancing automatically distributes incoming load to multiple virtual machines. This work is aimed to introduce efficient resource provisioning and de-provisioning for better load balancing. In this article, a model is proposed in which the fuzzy logic approach is used for load balancing to avoid underload and overload of resources. A Simulator in Matlab is used to test the effectiveness and correctness of the proposed model. The simulation results have shown that our proposed intelligent cloud-based load balancing system empowered with fuzzy logic is better than previously published approaches.

Keywords: Cloud computing; fuzzy logic; load balancing

1 Introduction

Cloud Computing is a novel paradigm [1] that has shown remarkable growth in the industrial and academic sectors. Cloud computing's essential features include on-demand self-service, extended network access, integration of resources, and consistent services. Cloud computing is classified into deployment and service models. The deployment of the cloud is made as public, private, hybrid, and community. In public deployment, a cloud provider owned the cloud infrastructure and made it available to the general public. A single organization exclusively manages the infrastructure and all operations in private deployment and may exist on-premises or off-premises. In a hybrid deployment, the cloud infrastructure consists of public and private cloud functionalities, while in the community deployment, the cloud infrastructure is for a particular

community. Cloud Computing provides its services through a set of virtualized resources such as servers, virtual machines, etc. Three fundamental service models, namely Infrastructure as service (IaaS), Software as service (SaaS), and Platform as service (PaaS), are used to offer cloud services.

The rapid growth in the cloud computing market has produced loads of increased work cloud computing demand and has created loads of increased workload on cloud data centers [2]. This is the reason that the main concerns of cloud computing are load balancing and energy consumption [3,4]. Load balancing is when the workload is assigned and reassigned among available resources to maximize the system's throughput. The process's primary concern is to minimize the cost, energy consumption, and response time to improve the overall system's overall resource utilization and performance [5,6]. Load balancing is the distribution of traffic, sending and receiving data, and data across all servers without delay with load balancing [7]. Cost efficiency, scalability, flexibility, and priority are the significant goals of load balancing. Load balancing algorithms are classified [8] depending on the system's state and on who initiated the process. In the case of the system's state, algorithms may be segregated as static or dynamic, while in the latter case, it depends on the requirement generated by the sender, receiver, or symmetric.

The remainder of this paper is organized as follows: Section 2 is the literature review. Section 3 presents the proposed model and evaluation of the output. Section 4 presents the simulation and discuss the results generated from the simulation. Sections 5 discusses the conclusions of the study.

2 Literature Review

In the research communities, the attention towards cloud computing is increasing day by day. The cloud infrastructure is designed to enable virtualized file-sharing of enhancing performance and develop an environment with efficient resource scheduling and effective load balancing. The user in the cloud environment uses virtualization to access their stored files and execute different tasks. In this context, efficient resource utilization and load balancing are very significant. Load balancing is the process of allocating and redistributing the workload among available resources to maximize throughput, reduce cost, response time, and power consumption, improve resource utilization, and performance [8]. A wildcard rule [9] is used to implement a load balancer. The implementation of the wild card rule improved the time spent on the switch. The Round-robin method is used for the flow of packets going to different ports. They give a new direction for load balancing by improving the load balance time by reducing the control plane's flow. Load Balancer is deployed to make the system fault-tolerant and highly available. It can be applied as an application load balancer or as a network load balancer. When a quick response is required, the load balancer is deployed as a network load balancer, while in case of high availability of different natures of applications like mobile application and desktop application, the application load balancer is deployed.

A queuing theory-based analytical model is presented in [10] to assess elasticity strategies' effects. The evaluation is made to measure the performance of cloud-based three-tier applications. They simulated the logic of CPU utilization based scale-out and scale-in actions. While in [11], Markov decision processes (MDPs) are used to propose a formal model for quantitative analysis of horizontal elasticity at the infrastructure level. The proposed model defines how VMs are adding to and removing from managed systems at run time. However, the work is limited to metrics of infrastructure strategies, and metrics related to deployed processes are not considered. A framework named ADVISE is proposed in [12] to evaluate elasticity in cloud services and applications. The proposed model's evaluation determines the typical elasticity behavior of cloud

services at run time by using a learning and clustering-based evaluation process. A framework named STRATFram is proposed in [13] that describes and evaluates the elasticity strategy for service-based business processes. The framework provides a service-based process holder with editors and languages to provide their proper elasticity model. It also provides service-based processes, a description, and the evaluation of elasticity strategies based on different elasticity models. The description part of the proposed model is used to conceal the complexities and underlying techniques to make the system more extensible.

Makespan time is the total time required to process a job or set of jobs for complete execution. Minimization of makespan time depends on the allocation of jobs to the virtual machines. In [14], a cloud resource broker-based architecture is designed and developed to attain minimum makespan time and improve resource utilization. The proposed model continuously monitors the virtual machine's workload, and decisions are made as per the defined threshold condition. They developed a heuristic-based load balancing algorithm and compared the Min–Min algorithm results, First Come First Serve, and Shortest Job First under all conditions. They proved that in the proposed algorithm, the makespan time is reduced by more than 10%, and the utilization ratio of cloud resources is enhanced by more than 30%. An algorithm based on the greedy technique is proposed in [15].

In the proposed algorithm, the makespan and execution time of tasks is minimized, but they do not consider task migration or virtual machine migration approach. However, the results of the proposed algorithm are not better in the real environment. Load balancing is performed through virtual machine migration and static and dynamic load balancing algorithms with their pros and cons. A dynamic algorithm using Honey bee behavior is proposed in [16]. The proposed algorithm Honey bee behavior inspired by load balancing (HBB-LB) is designed to minimize the response time while maximizing the throughput. The algorithms consider the priority of tasks so that the time to execute the job is minimal. The drawback of the proposed algorithm is that it may lead to starvation when low priority tasks remain in a wait state in the presence of a more priority-based queue. An efficient last optimal k interval-based dynamic task scheduling algorithm is proposed in [17], a modification cloud resource broker architecture [14]. With the proposed algorithm, the makespan time is reduced, and the ratio of tasks to meet deadlines is also increased. The scalability of the proposed algorithm is tested on a significant number of functions. The results show that the algorithm help makes smart decisions to manage the load of applications better.

Resource Scheduling is also put together with Load Balancing algorithms. An agent-based load balancing algorithm presented in [18] provides a dynamic load balancing for the cloud environment. The proposed algorithm only ensures the load balancing while the optimization concerns are not considered. As cloud computing is attaining attention quickly, geographically distributed data centers are also becoming very important. Therefore another critical aspect of cloud computing is energy efficiency. A k-mean Clustering algorithm is proposed in [19] that reduces energy consumption and delay by dynamically adjusting the number of machines. In Parallel scientific applications, the internal workload increases during the execution and demand resources to meet system requirements. In [20], a new elasticity controller is proposed for automatic resource provisioning. The fuzzy logic controller and autonomic computing are combined in the proposed controller. Their results justify that the proposed controller is more appropriate for elasticity in parallel applications. The implemented approach allows the application to request additional resources at run time. The fuzzy elasticity controller can collect information from the internal and external workload and deliver both the horizontal and vertical elasticity on the application's call.

They showed that the proposed controller minimized finish time by 64% and increased resource utilization by 36%.

In [21] Mamdani fuzzy inference system is used to present the smart city's idea in modeling complex traffic processes. They work on integrating cloud data, social networking services, and intelligent sensors to propose a framework in the context of smart cities. Their work to model complex traffic processes gives direction to understand traffic on the cloud. In [22], a multilayer Mamdani fuzzy inference system is used to classify the different stages of hepatitis. The proposed expert system classifies the stages of hepatitis as no hepatitis, acute HBV, or chronic HBV. The proposed method used two input variables at layer one and seven variables at layer two. The idea of classifying the stages is inspired to detect load as high, medium, and low in our proposed model. Our proposed work has also used the Mamdani fuzzy inference system with two input variables and one output variable.

3 Proposed System Model

Fig. 1 presented the system model architecture of our proposed intelligent cloud-based load balancing system empowered with fuzzy logic. In this research, dynamic load balancing is considered and proposed a new load balance system model for virtual machines based on the Round Robin Algorithm. The proposed model aims to find better processing and response time to the system's resources to manage its resources efficiently. The proposed model estimates the results efficiently for each virtual machine's status that either the machine is underload or overload. Fuzzy Logic is used to calculate the load status of each virtual machine. A user request is composed according to some rules. These requests are also referred to as jobs. To execute user requests, a suitable virtual machine is required. A virtual machine can perform many jobs.

The main things that are worth mentioning about a resource are its availability. The availability of the resource can be determined by computing its average load time. The time indicates whether a resource is an underload or overload. It also enforces to maintain the state of a resource. The state of a resource is considered as idle or busy. The idle state indicates that the resource is free and is not processing any request. At the same time, a busy state indicates that resource is processing some request. When a resource is in a busy state, it is required to compute the resource's load. The computed resource will provide information that either the resource is underload or overload. If the resource is underload, it can also be used to realize some other jobs.

The algorithm using in load balancing is started working before the job is processing on the server by using the parameters, e.g., virtual machine assigned load, speed of the processor, etc. The information is managed in each virtual machine and finds out the least loaded machine as shown in Tab. 1. The proposed model implements the technique of load balancing with the help of fuzzy logic.

3.1 Membership Functions

Membership value is defined by using the membership function as shown in Tabs. 2 and 3, which is between zero and one. A membership function can be written as.

$$\mu_{S\cap L}(s,l) = \min\left[\mu_S(s), \mu_L(l)\right]$$

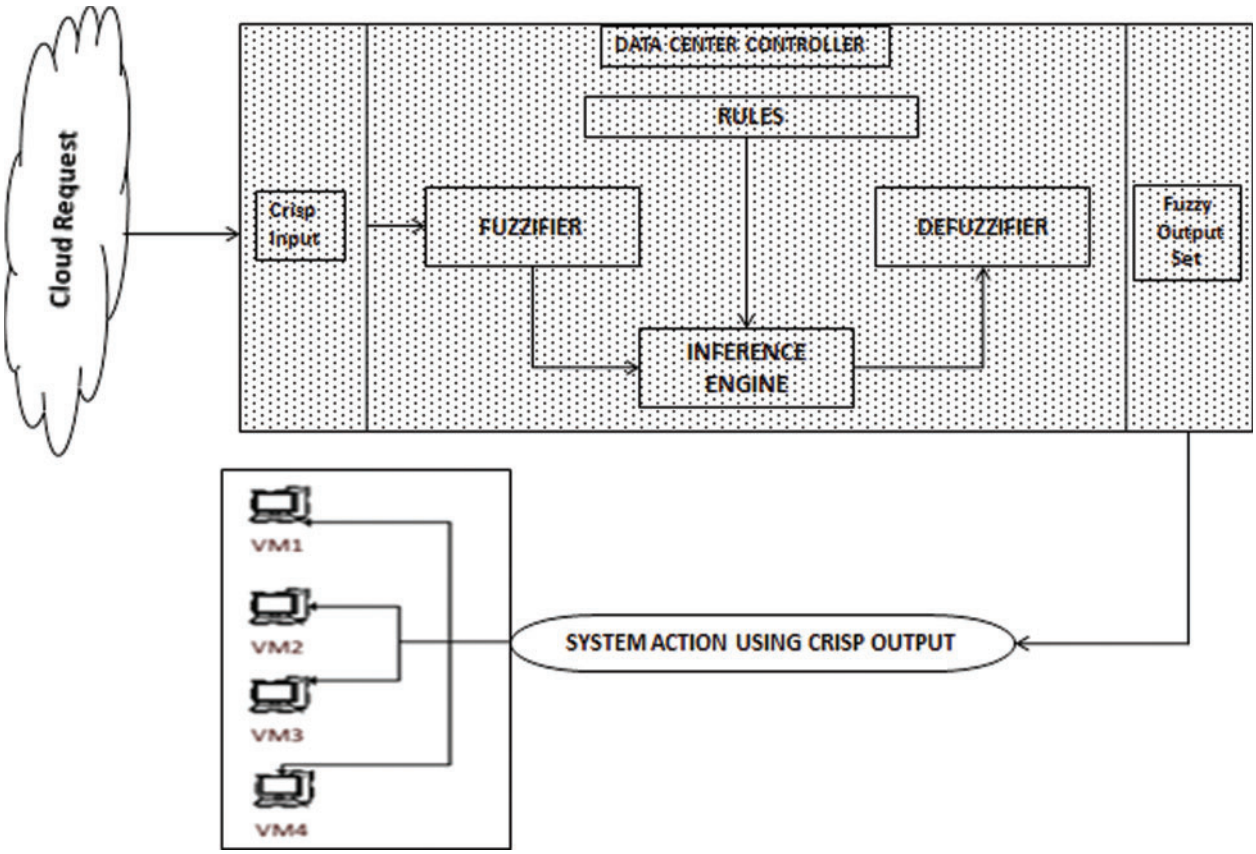


Figure 1: Proposed system model architecture of intelligent cloud-based load balancing system

3.2 Fuzzy Set Operations

Fuzzy set operations can be categorized into three types, named union (OR), intersection (AND), and Additive Compliment (NOT). Two fuzzy sets like Φ and ß defined on the universe Υ, $\chi \in \Upsilon$. It can be written as:

Intersection, [AND]: $\mu_{\Phi \cap ß}(\chi) = \min(\mu_{\Phi}(\chi), \mu_{ß}(\chi))$

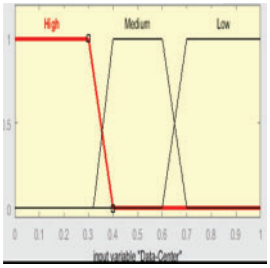
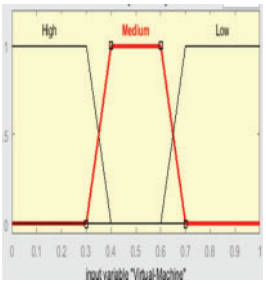
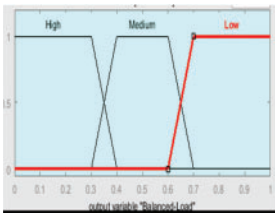
Union, [OR]: $\mu_{\Phi \cup ß}(\chi) = \max(\mu_{\Phi}(\chi), \mu_{ß}(\chi))$

Additive complement, [NOT]: $\mu_{\overline{\Phi}}(\chi) = 1 - \mu_{\Phi}(\chi)$

Table 1: Proposed system input/output variables

Sr. No.	I/O variable name
Input 1	Data center speed
Input 2	Virtual machine load
Output 1	Balanced load

Table 2: Graphical and mathematical MF of input/output variables

Variables	Membership Function(MF)	
Data Center Speed =S $\mu_S(s)$	$\mu_{S,High}(s) = \begin{cases} 1, & 0 \le s \le 0.3 \\ \frac{0.3-s}{1}, & 0.3 \le s \le 0.35 \\ 0, & s \ge 0.35 \end{cases}$	
	$\mu_{S,medium}(s) = \begin{cases} \frac{s-0.3}{1}, & 0.3 \le s \le 0.35 \\ 1, & 0.35 \le s \le 0.4 \\ \frac{0.7-s}{1}, & 0.4 \le s \le 0.6 \\ 0, & s \ge 0.6 \end{cases}$	
	$\mu_{S,Low}(s) = \begin{cases} 0, & 0.6 \le s \le 0.65 \\ \frac{0.7-s}{2}, & 0.65 \le s \le 0.7 \\ 1, & s \ge 0.7 \end{cases}$	
Virtual Machine Load =L $\mu_L(l)$	$\mu_{L,High}(l) = \begin{cases} 1, & 0 \le l \le 0.3 \\ \frac{0.3-l}{1}, & 0.3 \le l \le 0.35 \\ 0, & l \ge 0.35 \end{cases}$	
	$\mu_{L,Medium}(l) = \begin{cases} \frac{l-0.3}{1}, & 0.3 \le l \le 0.35 \\ 1, & 0.35 \le l \le 0.4 \\ \frac{0.7-l}{1}, & 0.4 \le l \le 0.6 \\ 0, & l \ge 0.6 \end{cases}$	
	$\mu_{L,Low}(l) = \begin{cases} 0, & 0.6 \le l \le 0.65 \\ \frac{0.7-l}{2}, & 0.65 \le l \le 0.7 \\ 1, & l \ge 0.7 \end{cases}$	
Balanced Load=B $(\mu_B(b))$	$\mu_{B,High}(b) = \begin{cases} 1, & 0 \le b \le 0.3 \\ \frac{0.3-b}{1}, & 0.3 \le b \le 0.35 \\ 0, & b \ge 0.35 \end{cases}$	
	$\mu_{B,Medium}(b) = \begin{cases} \frac{b-0.3}{1}, & 0.3 \le b \le 0.35 \\ 1, & 0.35 \le b \le 0.4 \\ \frac{0.7-b}{1}, & 0.4 \le b \le 0.6 \\ 0, & b \ge 0.6 \end{cases}$	
	$\mu_{B,Low}(b) = \begin{cases} 0, & 0.6 \le b \le 0.65 \\ \frac{0.7-b}{2}, & 0.65 \le b \le 0.7 \\ 1, & b \ge 0.7 \end{cases}$	

The rules used in Fuzzy systems are one of the essential components in the Fuzzy Inference System. The proposed model has nine rules based on the IF-THEN structure. Rules of the expert system can be written as: $RE = 1 \le n \le 9$.

RE^1 = IF Data − center is High AND Virtual − Machine is high THEN Balanced Load is high.

RE^2 = IF Data − center is Medium AND Virtual − Machine is Medium THEN Balanced Load is Medium.

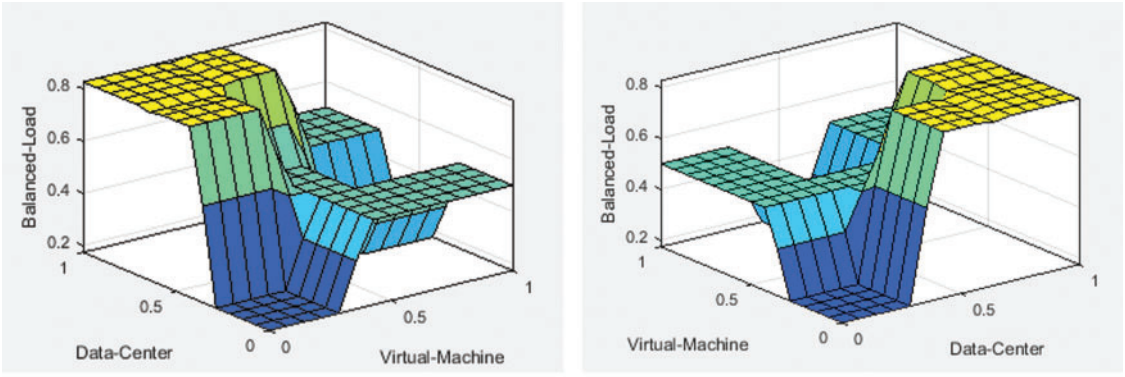
RE^3 = IF Data-center is Low AND Virtual-Machine is low THEN Balanced Load is low.

Figs. 2a and 2b shows the proposed system's surface view that represents the data distribution relationship in terms of input and output. A three-dimensional output surface can be generated

with the Surface Viewer, where two of the inputs vary. It shows the relationship of Balanced load with datacenter and virtual machine.

Table 3: Lookup table for fuzzy inference system

Rules	Data-center-speed (S)	Virtual-machine-load (L)	Balanced-load (B)
1	H	H	H
2	H	M	M
3	H	L	M
4	M	H	L
5	M	M	M
6	M	L	H
7	L	H	L
8	L	M	L
9	L	L	M



(a) (b)

Figure 2: (a) Rules surface of balanced load based upon the values of virtual machine and data-center. (b) Rules surface of balanced load based upon the values of data-center and virtual machine

4 Results and Discussion

Simulation results are generated by using the MATLAB R2017b tool with two input variables and one output variable. The proposed intelligent cloud-based load balancing system's performance is shown in Figs. 3a–3c.

Fig. 3a shows that if the data center's value is medium and the virtual machine is medium, then each machine load will be in balanced as medium. Similarly in Fig. 3b represent that if the value of datacenter is medium and the virtual machine is high, then each machine will be in low

balanced load, and it also observed that if the value of datacenter is medium and the virtual machine is low, then each machine will be in high balanced load.

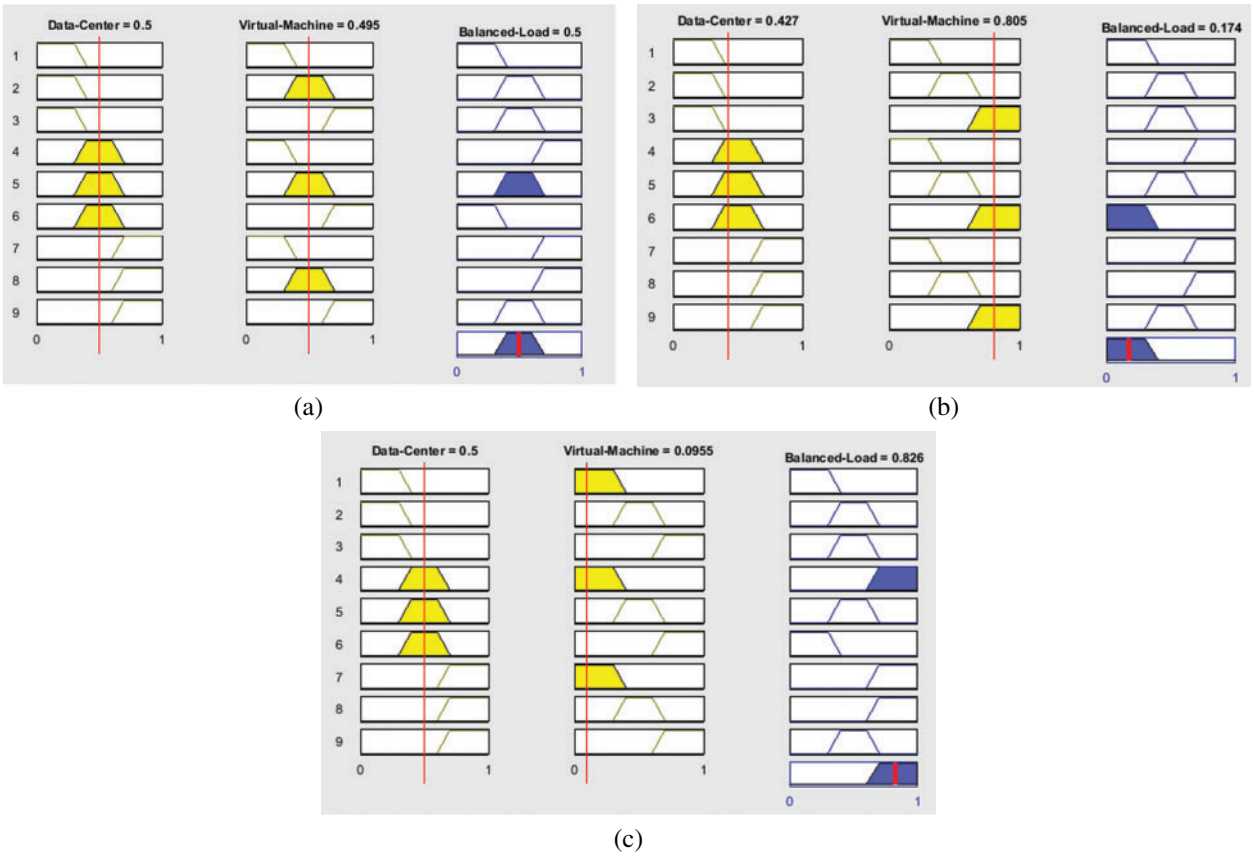


Figure 3: (a) Lookup diagram for medium balanced load. (b) Lookup diagram for low balanced load. (c) Lookup diagram for high balanced load

5 Conclusion

Cloud computing uses dynamic resource provisioning and de-provisioning in a virtualized environment. The growing demand for cloud computing has caused an increasing load in cloud data centers. Elasticity in cloud computing is one of the fundamental properties, and elastic load balancing automatically distributes incoming load to multiple virtual machines. The proposed intelligent cloud-based load balancing expert system is developed using the Mamdani fuzzy inference system divided the level of balanced load in the cloud computing environment into high, medium, and low. The simulation results confirmed that the proposed system continues a fair decision for load balancing in each machine and can be used in top hours, quickly, and efficiently.

Acknowledgement: Thanks to our families & colleagues, who supported us morally.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology*, vol. 53, no. 6, pp. 50, 2009.

[2] M. Ala'Anzy and M. Othman, "Load balancing and server consolidation in cloud computing environments: A meta-study," *IEEE Access*, vol. 7, pp. 141868–141887, 2019.

[3] Y. Jadeja and K. Modi, "Cloud computing concepts, architecture and challenges," in *Int. Conf. on Computing, Electronics and Electrical Technologies*, Tamil Nadu, India, pp. 877–880, 2012.

[4] C. Preist and P. Shabajee, "Energy use in the media cloud: Behaviour change, or technofix?," in *IEEE Int. Conf. on Cloud Computing Technologies and Science*, Indianapolis, Indiana USA, pp. 581–586, 2010.

[5] P. Singh, P. Baaga and S. Gupta, "Assorted load balancing algorithms in cloud computing: A survey," *Computer Applications*, vol. 143, no. 7, pp. 34–40, 2016.

[6] S. Goyal and M. K. Verma, "Load balancing techniques in cloud computing environment: A review," *Advance Research in Computer Science and Software Engineering*, vol. 6, no. 4, pp. 583–588, 2016.

[7] I. N. Ivanisenko and T. A. Radivilova, "Survey of major load balancing algorithms in distributed system," *Information Technologies in Innovation Business*, Kharkiv, Ukraine, pp. 89–92, 2015.

[8] D. Kashyap and J. Viradiya, "A survey of various load balancing algorithms in cloud computing," *Scientific and Technology Research*, vol. 3, no. 11, pp. 115–119, 2014.

[9] R. Wang, D. Butnariu and J. Rexford, "Open flow-based server load balancing gone wild," in *Proc. HOT-ICE'11*, Boston, MA, USA, 2011.

[10] B. Suleiman and S. Venugopal, "Modeling performance of elasticity rules for cloud based applications," in *IEEE Int. Enterprise Distributed Object Computing Conf.*, Vancouver, BC, Canada, pp. 201–206, 2013.

[11] A. Naskos, E. Stachtiari, P. Katsaros and A. Gounaris, *Probabilistic Model Checking at Runtime for the Provisioning of Cloud Resources*, Vienna, Austria: Runtime Verification, 275–280, 2015.

[12] G. Copil, D. Trihinas, H. L. Truong, D. Moldovan, G. Pallis *et al.,* "Advise—a framework for evaluating cloud service elasticity behavior," in *Int. Conf. on Service-Oriented Computing*, Paris, France, pp. 275–290, 2014.

[13] A. B. Jrad, S. Bhiri and S. Tata, "Stratfram: A framework for describing and evaluating elasticity strategies for service-based business processes in the cloud," *Future Generation Computer System*, vol. 97, pp. 69–89, 2019.

[14] M. Kumar and S. C. Sharma, "Load balancing algorithm to minimize the makespan time in cloud environment," *World Journal of Modeling and Simulation*, vol. 14, no. 4, pp. 276–288, 2018.

[15] B. Shaoo, D. Kumar and S. K. Jena, "Analysing the impact of heterogeneity with greedy resource allocation algorithms for dynamic load balancing in heterogeneous distributed computing system," *Computer Application*, vol. 62, no. 19, pp. 25–34, 2013.

[16] D. Babu and P. Venkata, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," *Applied Soft Computing*, vol. 13, no. 5, pp. 2292–2303, 2013.

[17] M. Kumar and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," *Computer and Electrical Engineering*, vol. 69, pp. 395–411, 2018.

[18] A. Singh, D. Junejab and M. Malhotraa, "Autonomous agent based load balancing algorithm in cloud computing," *Advanced Computing Technologies and Applications*, vol. 45, pp. 832–841, 2015.

[19] Q. Zhang, M. F. Zhani, E. Boutaba and J. L. Hellerstein, "Dynamic heterogeneity-aware resource provisioning in the cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 1, pp. 14–28, 2014.

[20] T. Bhardwaj and S. C. Sharma, "Fuzzy logic-based elasticity controller for autonomic resource provisioning in parallel scientific applications: A cloud computing perspective," *Computers and Electrical Engineering*, vol. 70, pp. 1049–1073, 2018.

[21] K. Iqbal, M. A. Khan and S. Abbas, "Intelligent transportation system for smart-cities using mamdani fuzzy inference system," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2, pp. 94–105, 2018.

[22] G. Ahmad, M. A. Khan, S. Abbas, A. Athar, B. S. Khan *et al.,* "Diagnosis Hepatitis B by using multilayer mamdani fuzzy inference system," *Journal of Healthcare Engineering*, vol. 2019, 6361318, 2019.