Comparative study on load balancing and service broker algorithms in Cloud computing using cloud analyst tool

Dr. Payaswini P
Assistant Professor, Goa University, Goa, India.

In recent years, there has been a large increase in the number of cloud users as it provides an easy and flexible way to manage user data and applications. With the emerging technologies such as the Internet of things, cloud being the backbone, the load on the cloud servers has increased. The cloud data centers consist of servers hosting multiple virtual machines. One of the main challenges in cloud computing is to efficiently distribute the user service requests to different virtual machines in order to reduce the request processing time and to provide more user satisfaction. Load balancing algorithms basically address two issues: the selection of the data center and the distribution of the load on different virtual machines. Data center selection is handled by service broker policy and distribution of the load on virtual machines handled by VM load balancer. Load balancing techniques play a vital role in minimizing the response time and maximizing throughput and also ensures scalability and reliability. Hence, it has become an important research topic in the field of cloud computing. Cloud-analyst, a java based open source toolkit, is useful to simulate and analyse the load balancing algorithms. In this paper, a comparative study on different service broker policies and VM load balancing algorithms for cloud computing is presented with simulation results. The aim of this comparative study is to find the performance of different service broker policies and the load balancing algorithm tested under different scenarios.

Keywords: Cloud computing, virtual machine, load balancing, service broker policy, Cloud Analyst

1. INTRODUCTION

Cloud computing allows utilization of computing resources over the Internet (Mishra et al. [2020]). These computing resources are sharable and configurable according to the user needs and requires minimal management effort (Mell and Grance [2009]; Hogan et al. [2011]). Cloud computing consists of several interconnected servers, Virtual Machines (VMs), storage devices and Data Centers (DCs). It uses the pay as you go computing model, in which the user can access the computing resources over the Internet whenever needed and will be charged according to the usage of the resources (Chun and Choi [2014]). Cloud infrastructure has hardware and software in the DCs to provide these services. Moreover, these computing resources are supported on heterogeneous platforms such as desktop, laptop or any other mobile devices. The cloud services are categorised as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Fig. 1 shows some of the cloud services and its typical application. IaaS provides IT infrastructure support such as servers and VMs, storage media, networks, operating systems. PaaS allows the developer to focus on the deployment and management of the applications. SaaS provides a complete software which is completely managed by the service provider (Bojanova and Samba [2011]).

On the basis of Cloud Deployment models, there are four classes namely public, private, community and hybrid cloud. In Public Cloud, the cloud resources are owned and operated by the cloud provider and the user has no control over privacy or security. Whereas in private cloud, a dedicated cloud infrastructure such as storage and hardware are available for a particular organization, and offers greater security and privacy. The combination of public and private cloud is termed as hybrid cloud. In community cloud, many organizations jointly construct and share the cloud infrastructure and are overseen by them (Jadeja and Modi [2012]).

The cloud service providers have their own DCs. A DC houses servers and data storage facilities

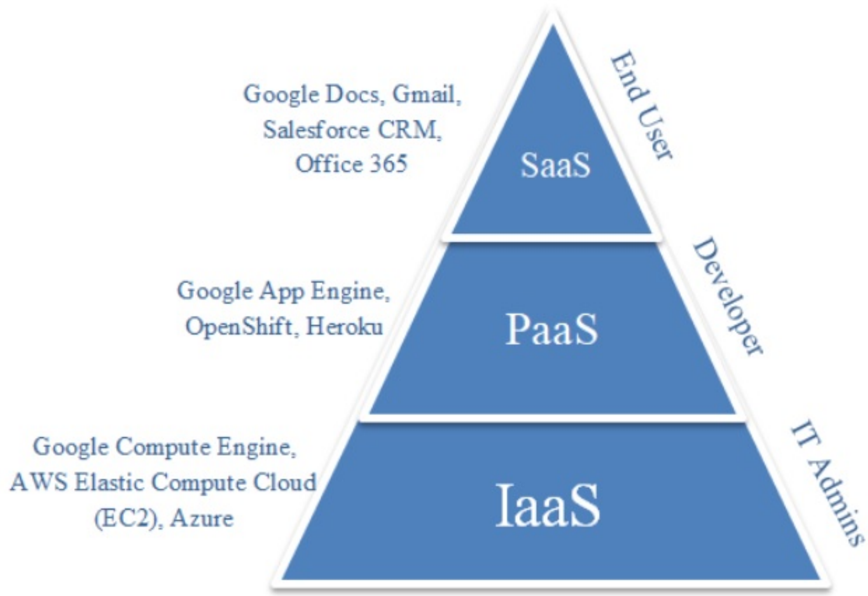

Figure 1. Cloud service model

for an organization and typically located at different geographic locations. The main feature of cloud computing is virtualization, in which multiple VMs are assigned to a single physical server (Xing and Zhan [2012]). The cloud resources which are in the virtual form are called cloud nodes. Even though the virtualization technique improves the efficiency of the DCs, there is a possibility of DCs getting overloaded when there is an increased number of user requests at the same time. As the number of cloud users are increasing, the main challenge is to assign the incoming requests to the cloud node in an efficient manner to provide very low response time. Load balancing algorithms mainly focus on efficient distribution of user requests to cloud nodes ensuring maximum user satisfaction (Xu et al. [2017]). The contributions of this paper are:

i) Basic introduction of load balancing in cloud computing.
ii) A survey on various existing load balancing algorithms.
iii) Comparative study of few existing load balancing algorithms using Cloud Analyst considering different configurations of the parameters.
iv) Performance comparison of the load balancing algorithms with regards to response time and cost of the service.

The rest of the paper is organized as follows: Section II provides introduction to load balancing in cloud computing, section III discusses the existing load balancing algorithms. Section IV, provides an overview of existing load balancing algorithms in Cloud Analyst tool. In section V, test scenarios and simulation of different load balancing algorithms and their behaviour under different loads are discussed. Section VI provides the summary and conclusion of the study.

2. LOAD BALANCING IN CLOUD COMPUTING

As discussed in the previous section, the Cloud system has a large number of DCs which share the load to complete the user task. The improper selection of the DC may increase load on VMs and might result in overloading of DC. Load balancing in cloud computing is a process of efficiently distributing the user tasks to different VMs and ensuring that no single VM is overwhelmed (Tiwari and Joshi [2016]). These algorithms should be capable of detecting overloaded and under loaded VMs and reassign the tasks to balance the load among them. Hence an efficient load balancing algorithm will distribute the work evenly among all the VMs and make sure that none of the VMs are overloaded. This will help in minimizing the resource consumption and to improve the resource utilization (Al Nuaimi et al. [2012]). Several research groups working on load balancing algorithms are mainly focusing on designing methods to assign the user task to the available VMs with minimal response time.

In cloud computing, the load balancing consists of two aspects; selection of DC and VM management at each DC (Jyoti et al. [2020]). The selection of the DC is handled by Service Broker policy also known as DC selection. The load on different VMs at each DC is managed by DC Controller. The Service Broker uses the service policies to select the most appropriate DC to respond to the user request (Manasrah et al. [2017]). Service broker helps in controlling the traffic between the DC and user bases. Hence, it acts as an intermediary between the cloud user and service provider. Many algorithms have been proposed by the researchers to improve performance of existing service broker policies. Some of these are discussed in the next section. The DC controller has a load balancer which decides the selection of VM for a particular user task (Kumar and Kalra [2016]). The servers in the DCs are dynamically configured to host multiple VM. Hence, it is very important to select the VMs properly by using an appropriate load balancing algorithm.

Many research articles have categorised the load balancing algorithms into static and dynamic (Xu et al. [2017]; Al Nuaimi et al. [2012]; Yongjun [2008]). Static algorithms, assign the user task to the VM based on prior knowledge such as processing power, storage capacity, and performance. These algorithms do not consider the current system state information and are not suitable for the environment with highly varying load. Dynamic load balancing algorithms assign the task based on the current state of the system and also uses prior collected information about the VM (Randles et al. [2010]). These algorithms require constant monitoring of the VM as task progress and also balance the load more efficiently (Jyoti et al. [2020]). In the next section some of the existing load balancing algorithms are discussed.

3. RELATED WORK

For any researcher, the first step to begin the research work is to understand the research area. Several articles have been published (Dillon et al. [2010]; Yang and Chen [2010]; Aguiar et al. [2014]) to provide the basic introduction to cloud computing, cloud services and deployment models and also to provide overview on research challenges in the cloud computing area. These papers will help the researcher to understand the basic architecture of cloud computing and the open research challenges. There has been a significant amount of research work done on cloud load balancing as well. The comparison of different static and dynamic load balancing algorithms is presented by Katyal and Mishra [2014]. The comparison was done based on different load balancing scenarios such as static, dynamic, centralized, distributed and hierarchical and workflow dependent. The paper also provides an insight into cloudsim simulation tool. Patel and Patel [2015] presented a comparative study on the VM load balancing algorithms and different service broker policies with various configurations in Cloud Analyst.

It is important to measure the performance of proposed algorithms using different metrics. Mishra et al. [2020] discussed various load balancing techniques in both homogeneous and heterogeneous cloud environments. It also presents some of the important performance metrics to evaluate the system performance. Jyoti et al. [2020] presented a detailed study on work published in the year from 2015 to 2018 on existing service broker policy and load balancing algorithms. The paper provides in depth analysis of the different load algorithms along with their merits and demerits. The authors have used as many as eight key parameters to compare the algorithms along with the discussion on time complexity of these algorithms. Furthermore, the authors have presented the study on different simulation tools available for cloud computing. These survey articles are very useful for the amateur researchers to begin their work.

4. CLOUD ANALYST

Simulation helps to study the behavior of the algorithm in cloud environments before implementing them in real time. Cloudsim is a toolkit for modelling and simulation of Cloud computing systems proposed by Calheiros et al. [2011]. The tool allows modelling of cloud components such as user bases, VMs, DCs, and various resource provisioning policies. Cloud Analyst is a simulation tool based on cloud-sim, introduced by CLOUDS Laboratory (Wickremasinghe et al. [2010]). A detailed description of the simulation tool is given in (Buyya [2009]; Wickremasinghe et al. [2010]).

The Cloud Analyst tool provides an easy to use GUI through which users can configure the simulation parameters. This includes DC characteristics such as architecture, Operating System, Number of VMs, the cost of the services etc and VM parameters such as Memory Size, processing power and bandwidth. It defines six regions numbered from 0 to 5, basically are the representation of six main continents in the world. It has a feature to define a User Base which is a group of users generating the traffic for the simulation. The number of requests are grouped into a single Internet Cloudlet which can be configured by the user. The simulation provides graphical output for analyzing the results such as response time and service cost.

In Cloud Analyst, the implementation of load balancing is divided into two modules: DataCenterController and CloudAppServiceBroker. These modules are discussed in the next section. CloudAppServiceBroker uses broker policy to select the DC to service the user request. DC controller uses a VM load balancer to balance the load on different VMs. The separation of the load balancing task into two modules makes it easy to integrate new service broker policies and load balancing algorithms.

4.1 Service Broker Policies

The current version of Cloud Analyst implements three types of service broker policies: Closest Data Center, Optimize Response Time and Reconfigure Dynamically with Load (Manasrah et al. [2017]). The Closest DC is a proximity based broker policy, in which the nearest DC is selected. Here the DC with the lowest network latency is considered as nearest and hence the policy provides minimum response time (Limbani and Oza [2012]). However, it does not consider the channel bandwidth and the present load on the DC. This may result in overloading of the DC and may also result in congestion in the communication channel.

In Optimize Response Time policy, the broker selects the DC based on multiple parameters such as network latency, DC workload and response time of last task (Benlalia et al. [2019]). The broker continuously monitors the performance of all data centers. The traffic will be routed to the DC with the best response time.

Reconfigure Dynamically with Load (Rekha and Dakshayini [2014]) is a dynamic load based policy which is similar to the proximity based routing, but the broker increases and decreases the number of VMs allocated in the DCs based on the load it is facing. However, the simulation study shows that the Reconfigure Dynamically with Load policy takes more time for simulation to complete and gives higher response time compared to other two service broker policies.

Closest DC routing algorithm selects DC randomly, when there are more than one DC in the same geographical area. This may result in selecting a higher cost DC. To address the problem, few enhancements were proposed in the literature. An improved algorithm was proposed by Limbani and Oza [2012]. In this Algorithm, the cost of the VM is taken into consideration and the DC with lowest cost is selected. However the results indicate that the algorithm takes more data processing time compared to the conventional algorithm but provides a cost effective solution. Mishra and Bhukya [2014] proposed a priority based round robin service broker algorithm, which considers the speed of the DC. The algorithm introduced a preprocessing phase to prepare a priority list of DCs according to the speed. The DC selection algorithm routes user requests evenly to all DCs in a region.

Rekha and Dakshayini [2014] proposed a cost based service broker policy which reduces the waiting time and cost of the cloud service to users by considering the location of the DC. The algorithm routes the request during peak hours to a DC which is located in a different region with less load on the system. The simulation results show the algorithm achieves reduction in service cost and DC processing time.

An enhanced service broker policy is proposed by Rani et al. [2015]. In this algorithm, the authors have used a parameter Percentage of processor utilization, which is a ratio of useful CPU time over total CPU time. The efficiency of the system is calculated based on this parameter. For each DC, a threshold value for efficiency is defined. For the new request, the DC with highest efficiency value is selected and its efficiency value is updated based on current load. However, the simulation result shows slight increase in the DC request servicing time.

A service broker policy proposed by Manasrah et al. [2017] considers network channel bandwidth, latency and the size of the job to select the DC. The algorithm sorts the DCs based on the delay and channel bandwidth between the user base and DC. From the simulation results, it is seen that the proposed service broker policy improves the response and processing time. However, the proposed method does not consider the service cost.

4.2 VM load balancer

The current version of Cloud Analyst implements three basic load balancing algorithms; Round Robin, Throttled and Equally spread load balancer.

In Round-robin load balancer, the time is divided into multiple slices and all the VMs in DC receive the load in circular order. The algorithm does not consider the processing power of the VM during task allocation. Hence, this method is suited when all the VMs of the DC have the same processing power.

Throttled load balancer defines a throttling threshold for the number of requests being processed in each VM. If more requests are received exceeding the threshold value, then the requests are queued until the next VM becomes available (Phi et al. [2018]).

In equally spread the load, also known as active monitoring, the load balancer tries to maintain equal workload on all the VMs of the DC. The balancer maintains information about the requests currently allocated to each of the VM. When a new request arrives, it identifies the least loaded VM to service the request (Mohapatra et al. [2013]).

Apart from these, there are a number of load balancing algorithms proposed by the researchers. A modified throttled load balancer proposed by Domanal and Reddy [2013] is an improved version of throttled load balancer. The throttled load balancer maintains an index table of VMs and its state keep track of current load on each VM. In the modified throttled algorithm, the searching time to select the VM from the index table is reduced by modifying the index table search method. In the proposed, the algorithm keeps track of the index of the last allocated VM. For a new user request, the index table search will start from the index location next to the last assigned VM, instead of starting from the beginning. This reduces the search time and so the response time.

Many researchers have applied optimization algorithms to design techniques for load balancing in cloud computing. Some of these algorithms are inspired by natural phenomena such as Ant Colony Optimization, Honeybee foraging and Particle Swarm Optimization. Artificial bee colony algorithm (Karaboga [2005]) is based on the intelligent foraging behavior of honey bee swarms. Dorigo et al. [2006] proposed an ant colony based approach to solve optimization problems. The algorithm is inspired by the behaviour of ants for searching the food. Another optimization algorithm, Particle Swarm Optimization (PSO) is based on social behaviour of bird flocking (Eberhart and Kennedy [1995]).

LD and Krishna [2013] proposed an algorithm based on honey bee behaviour for load balancing which also considers the priority of the task waiting in a VM queue. The algorithm calculates the capacity of each VM using the parameters such as number of processors, its processing power and communication bandwidth. The capacity of the DC is the summation of capacity of all the VM in it. Based on the threshold parameter and the current load on VM, it checks for the need of load balancing. Based on the load on VM, they are categorized as overloaded VMs, underloaded

VMs and balanced VMs. If a VM is overloaded, the task from the queue is removed and the algorithm finds a VM which has a minimum number of high priority tasks and ths balances the load.

Dam et al. [2014] proposed an ant colony based solution to balance the load by searching under loaded nodes. The proposed method initially allocates the VM based on First Come First Served basis and maintains an index table. When all the VMs become unavailable for the new user request, artificial ants are created to check the under loaded VM in the DC. The authors have used three parameters; communication channel bandwidth, the maximum capacity of each processor of VM and delay cost which is a penalty on the cloud service provider in the event of time taken to complete the job is more than the deadline advertised.

Some of the load balancing algorithms are based on VM migration techniques. In this technique, a running instance of a VM on a physical host is migrated to another to balance the load (Ramezani et al. [2014]). In order to migrate, the running instance of the VM is paused, its state data is copied and the VM will be resumed on the destination host. Ramezani et al. [2014] proposed load balancing using improved online VM migration technology with Particle Swarm Optimization (PSO) technique. The traditional migration technique would migrate the entire VM when loaded which results in more time and cost consuming. The migration process also requires a large amount of memory. Hence the authors proposed that rather than migrating the entire VM, only certain tasks be migrated. The proposed method helps in reducing the time and memory consumption also minimizes task execution time and task transfer time.

5. SIMULATION SET-UP

For the simulation study Cloud Analyst tool was used with the additional VM load balancing algorithms; Honeybee (Karaboga [2005]), PSO (Eberhart and Kennedy [1995]) and Ant Colony Optimization (Dorigo et al. [2006]). The load balancing algorithms are compared in terms of the average response time and service cost. The response time is the time gap from the submission of the user request to its completion. The total cost of the service includes cost per VM, memory cost, storage cost and data transfer cost. In all the experiments, the parameters such as VM cost, storage cost, memory size, processor speed and bandwidth for all the DCs were kept constant.

A number of experiments have been conducted by creating different scenarios to test the performance of the algorithms. Out of these, three have been presented based on the best results. The scenarios and its configuration details are discussed in the later part. The load on the system is varied from low to high. The transmission delay (in millisec) and bandwidth (in Mbps) between the regions are given in the table 1 and 2 respectively. Also the configuration of the advance parameters is given in table 3. The duration of simulation was set to 120 min. These parameters were kept constant for all the scenarios.

Region	**0**	**1**	**2**	**3**	**4**	**5**
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	100	200	200	500	500	25

Table I: Transmission delay(in ms) between the regions.

5.1 Scenario I: Each of the regions is configured to have a single DC.

The aim is to check the performance of the VM load balancing algorithms and service broker policies when the load on the system is low and the DC configuration is homogeneous. Hence, six DCs in six different regions are created with the same configuration for the DC parameters. Here, each of the six regions is configured with a single User base. The user base is configured

Region	0	1	2	3	4	5
0	2000	1000	1000	1000	1000	1000
1	1000	800	1000	1000	1000	1000
2	1000	1000	2500	1000	1000	1000
3	1000	1000	1000	1500	1000	1000
4	1000	1000	1000	1000	500	1000
5	1000	1000	1000	1000	1000	2000

Table II: The available bandwidth (in Mbps) between the regions.

Parameter	Value
The executable instruction length per request	512 Bytes
User grouping factor in user base	10
Request grouping factor in DC	10

Table III: Configuration of advanced parameter values.

such a way that the number of users requesting the cloud service was kept low. The parameter for the user base is as shown in Table 3. Each DC has been configured with 4 CPUs and 25 VM.

Name	Region	Request per user per hour	Data size per request(bytes)	Peak hours start	Peak hours end	Avg peak users	Avg off-peak users
UB1	0	60	10000	3	9	1500	150
UB2	1	60	10000	3	9	1000	100
UB3	2	60	10000	3	9	1300	150
UB4	3	60	10000	3	9	1200	120
UB5	4	60	10000	3	9	1600	160
UB6	5	60	10000	3	9	1400	140

Table IV: User base configuration for scenario I.

The average response time for scenario I is as shown in the fig 2. From the graph, it is clear that all the six VM load balancing algorithms show more or less the same response time for Closest DC and Optimize Response Time service broker policy. However, in case of Reconfigure Dynamically with Load service broker policy, Honeybee load balancing algorithm shows higher response time and Throttled load balancing algorithm shows lowest response time. Overall, Closest DC and Optimize Response Time service broker policy performs better in selecting the data center. These two service broker policies with equally spread the load and throttled load balancer provide lowest response time among the all the other selected VM load balancing algorithms.

The table 5 shows the comparison of the algorithms with respect to the cost of the service. There is no change in the cost of the service for service broker policies: closest DC and optimize response time. However, for Reconfigure Dynamically with Load broker policy, PSO algorithm gives minimum service cost.

5.2 Scenario II: Three regions with two DCs each and the system load is high.

The aim is to check the performance of the algorithms when a region has more than one DC of heterogeneous configurations. Here, each DC is configured with a different number of Hardware units and VMs. Two DCs each created in the regions 0, 3 and 4 respectively. The number of users in each user base are increased and as shown in the table 6. Also, the executable instruction length has been increased to 1000 bytes. The number of VM and the number of Hardware units allocated to each of the DCs are shown in table 7.

Since each region has two DCs of different processing power, it is important to choose the right DC in order to balance the load. From the graph (fig. 3), it is clear that except for the PSO algorithm, the response time for the remaining load balancing algorithms are almost the

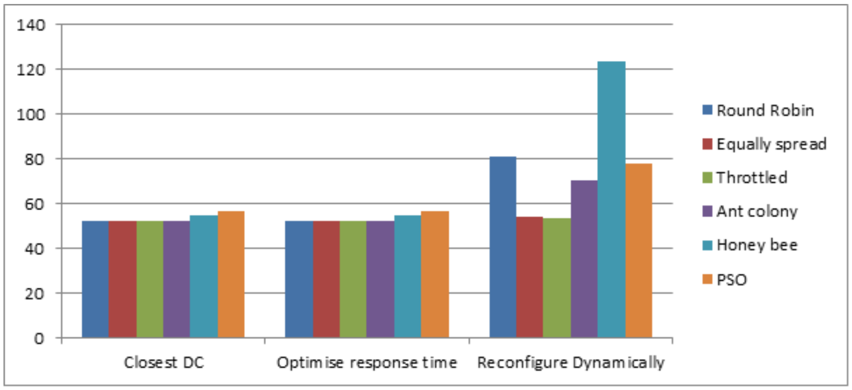

Figure 2. Avg response time (in ms) for scenario I

Algorithm	**Closest DC**	**Optimize Response Time**	**Reconfigure Dynamically with Load)**
Round Robin	134.06	134.06	193.69
Equally spread load	134.06	134.06	193.7
Throttled	134.06	134.06	193.34
Ant colony	134.06	134.06	193.63
Honey bee	134.06	134.06	193.54
PSO	134.06	134.06	192.91

Table V: Comparison of cost of the service for scenario I.

Name	**Region**	**Request per user per hour**	**Data size per request(bytes)**	**Peak hours start**	**Peak hours end**	**Avg peak users**	**Avg off-peak users**
UB1	0	60	10000	3	9	15000	1500
UB2	1	60	10000	3	9	10000	1000
UB3	2	60	10000	3	9	13000	1500
UB4	3	60	10000	3	9	12000	1200
UB5	4	60	10000	3	9	16000	1600
UB6	5	60	10000	3	9	14000	1400

Table VI: User base configuration for scenario II.

same for Closest DC and Optimize Response Time service broker policy. For this particular scenario, Honeybee algorithm showed consistent results for response time for all three service broker policies.

Table 8 shows the comparison of the algorithms with respect to the service cost. It can be concluded from the table that, PSO algorithm provides minimum service cost to complete the user request.

5.3 scenario III: Simulation to test the effect of length of the job.

In the previous two scenarios, there wasnt much difference in response time for VM load balancing algorithms with Closest DC and Optimize Response Time service broker policy. Hence, the length of the job is increased to check the performance of the algorithms. Here, the executable instruction length has been set to 100KBytes. Two DCs each created in the regions 0, 3 and 4 respectively. Each DC is configured to have 10 VMs each and 2 Hardware units consisting of 2 CPUs each. The configuration of the User base is the same as in scenario I. From the graph (fig. 4) it is clear that throttled load balancing algorithm gives minimum response time with Optimize Response Time and Closest DC service broker policies. Also, Honeybee VM load balancing algorithm shows

Data center	Region	No of HW units	No. of VM
DC1	0	2	15
DC2	0	4	25
DC3	3	2	15
DC4	3	6	45
DC5	4	6	45
DC6	4	4	25

Table VII: DC configuration for scenario II.

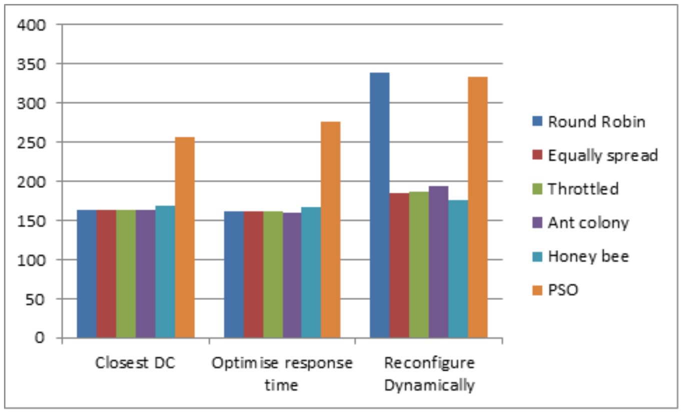

Figure 3 Average response time (in ms) for scenario II

Algorithm	Closest DC	Optimize Response Time	Reconfigure Dynamically with Load)
Round Robin	1073.85	1073.85	1131.4
Equally spread load	1073.85	1073.85	1131.51
Throttled	1073.85	1073.85	1131.31
Ant colony	1073.85	1073.85	1130.98
Honey bee	1073.85	1073.85	1130.76
PSO	1073.85	1073.85	1130.61

Table VIII: Comparison of cost of the service for scenario II.

minimum response time with Reconfigure Dynamically with Load service broker policy. The table 9 shows the comparison of the algorithms with respect to service cost. PSO load balancing algorithm shows lowest cost for all three service broker policies. Overall, from the simulation study, it has been observed that throttled and equally share load balancing algorithms perform better with respect to the response time along with Closest DC and Optimize Response Time service broker policies. However, PSO, Honey Bee and ant colony algorithms perform better with respect to the cost of the service in all the scenarios.

6. CONCLUSION

Load Balancing in cloud computing is essential for efficient utilization of the resources. The two important key factors in load balancing are selection of the DC and assigning the requests to available VM. Since there are many DCs across different geographical locations, appropriate selection of DC plays an important role to improve the performance and also to reduce the cost. In this paper, a comparative study of some of the load balancing algorithms and service broker policies is presented using Cloud Analyst tool. The comparison is done with respect to average response time and cost of the service considering different scenarios. The study shows that the

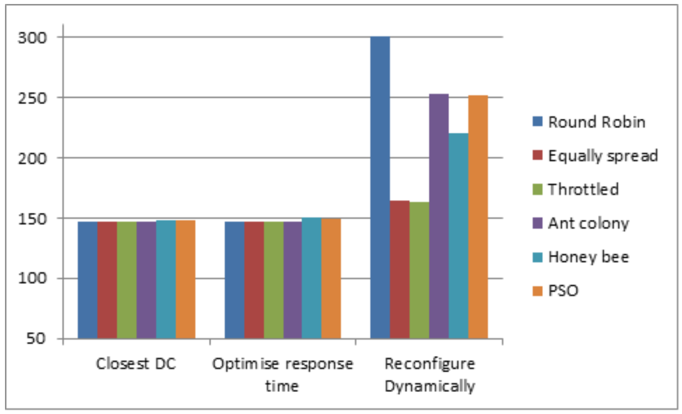

Figure 4 Average response time (in ms) for scenario III

Algorithm	**Closest DC**	**Optimize Response Time**	**Reconfigure Dynamically with Load)**
Round Robin	93.53	93.53	101.12
Equally spread load	93.53	93.53	101.14
Throttled	93.53	93.53	101.51
Ant colony	93.53	93.53	100.84
Honey bee	93.53	93.53	100.7
PSO	93.53	93.53	99.97

Table IX: Comparison of cost of the service for scenario III.

load balancing algorithm works on the workload created and the results vary from one scenario to another. After various rounds of simulation it can be summarized that the Optimize Response Time service broker policy gives the best result for most of the scenarios. However, when the DCs and user base are uniformly distributed, Closest DC, also gives better results. From the simulation study, it is seen that throttled and equally share load balancing algorithms performs better. The comparison of algorithms with respect to service cost shows that PSO algorithm performs better in all the scenarios.

References

Aguiar, E., Zhang, Y., and Blanton, M. 2014. An overview of issues and recent developments in cloud computing and storage security. *High Performance Cloud Auditing and Applications*, 3–33.

Al Nuaimi, K., Mohamed, N., Al Nuaimi, M., and Al-Jaroodi, J. 2012. A survey of load balancing in cloud computing: Challenges and algorithms. In *2012 second symposium on network cloud computing and applications*. IEEE, 137–142.

Benlalia, Z., Beni-hssane, A., Abouelmehdi, K., and Ezati, A. 2019. A new service broker algorithm optimizing the cost and response time for cloud computing. *Procedia Computer Science 151*, 992–997.

Bojanova, I. and Samba, A. 2011. Analysis of cloud computing delivery architecture models. In *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*. IEEE, 453–458.

Buyya, R. 2009. Cloudanalyst: A cloudsim-based tool for modelling and analysis of large scale cloud computing environments. *Distrib. Comput. Proj. Csse Dept. Univ. Melb*, 433–659.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., and Buyya, R. 2011. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience 41,* 1, 23–50.

Chun, S.-H. and Choi, B.-S. 2014. Service models and pricing schemes for cloud computing. *Cluster computing 17,* 2, 529–535.

Dam, S., Mandal, G., Dasgupta, K., and Dutta, P. 2014. An ant colony based load balancing strategy in cloud computing. In *Advanced Computing, Networking and Informatics-Volume 2*. Springer, 403–413.

Dillon, T., Wu, C., and Chang, E. 2010. Cloud computing: issues and challenges. In *2010 24th IEEE international conference on advanced information networking and applications*. Ieee, 27–33.

Domanal, S. G. and Reddy, G. R. M. 2013. Load balancing in cloud computingusing modified throttled algorithm. In *2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 1–5.

Dorigo, M., Birattari, M., and Stutzle, T. 2006. Ant colony optimization. *IEEE computational intelligence magazine 1,* 4, 28–39.

Eberhart, R. and Kennedy, J. 1995. A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. Ieee, 39–43.

Hogan, M., Liu, F., Sokol, A., and Tong, J. 2011. Nist cloud computing standards roadmap. *NIST Special Publication 35*, 6–11.

Jadeja, Y. and Modi, K. 2012. Cloud computing-concepts, architecture and challenges. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*. IEEE, 877–880.

Jyoti, A., Shrimali, M., Tiwari, S., and 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.

Karaboga, D. 2005. An idea based on honey bee swarm for numerical optimization. Tech. rep., Citeseer.

Katyal, M. and Mishra, A. 2014. A comparative study of load balancing algorithms in cloud computing environment. *arXiv preprint arXiv:1403.6918*.

Kumar, A. and Kalra, M. 2016. Load balancing in cloud data center using modified active monitoring load balancer. In *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring)*. IEEE, 1–5.

LD, D. B. and Krishna, P. V. 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied soft computing 13,* 5, 2292–2303.

Limbani, D. and Oza, B. 2012. A proposed service broker policy for data center selection in cloud environment with implementation. *International Journal of Computer Technology & Applications 3,* 3, 1082–1087.

Manasrah, A. M., Smadi, T., and ALmomani, A. 2017. A variable service broker routing policy for data center selection in cloud analyst. *Journal of King Saud University-Computer and Information Sciences 29,* 3, 365–377.

Mell, P. and Grance, T. 2009. Draft nist working definition of cloud computing. *Referenced on June. 3rd 15,* 32, 2.

Mishra, R. K. and Bhukya, S. N. 2014. Service broker algorithm for cloud-analyst. *International Journal of Computer Science and Information Technologies 5,* 3, 3957–3962.

Mishra, S. K., Sahoo, B., and Parida, P. P. 2020. Load balancing in cloud computing: A big picture. *Journal of King Saud University-Computer and Information Sciences 32,* 2, 149–158.

Mohapatra, S., Rekha, K. S., and Mohanty, S. 2013. A comparison of four popular heuristics for load balancing of virtual machines in cloud computing. *International Journal of Computer Applications 68,* 6.

Patel, H. and Patel, R. 2015. Cloud analyst: an insight of service broker policy. *International Journal of Advanced Research in Computer and Communication Engineering 4,* 1, 122–127.

Phi, N. X., Tin, C. T., Thu, L. N. K., and Hung, T. C. 2018. Proposed load balancing algorithm to reduce response time and processing time on cloud computing. *Int. J. Comput. Networks Commun 10,* 3, 87–98.

Ramezani, F., Lu, J., and Hussain, F. K. 2014. Task-based system load balancing in cloud computing using particle swarm optimization. *International journal of parallel programming 42,* 5, 739–754.

Randles, M., Lamb, D., and Taleb-Bendiab, A. 2010. A comparative study into distributed load balancing algorithms for cloud computing. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 551–556.

Rani, P., Chauhan, R., and Chauhan, R. 2015. An enhancement in service broker policy for cloud-analyst. *International Journal of Computer Applications 115,* 12, 5–8.

Rekha, P. and Dakshayini, M. 2014. Cost based data center selection policy for large scale networks. In *2014 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*. IEEE, 18–23.

Tiwari, P. K. and Joshi, S. 2016. A review on load balancing of virtual machine resources in cloud computing. In *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 2*. Springer, 369–378.

Wickremasinghe, B., Calheiros, R. N., and Buyya, R. 2010. Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In *2010 24th IEEE international conference on advanced information networking and applications*. IEEE, 446–452.

Xing, Y. and Zhan, Y. 2012. Virtualization and cloud computing. In *Future Wireless Networks and Information Systems*. Springer, 305–312.

Xu, M., Tian, W., and Buyya, R. 2017. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurrency and Computation: Practice and Experience 29,* 12, e4123.

Yang, J. and Chen, Z. 2010. Cloud computing research and security issues. In *2010 International Conference on Computational Intelligence and Software Engineering*. IEEE, 1–3.

Yongjun, L. 2008. Lixiaole, and sun ruxiang,load balancing algorithms overview,. *Information Development and Economy 18*, 134–136.

Dr. Payaswini P received M.Sc. degree in Computer Science in 2010 and a Ph.D degree in Computer Science in 2017 from Mangalore University, Mangalore, Karnataka. She is currently working as an Assistant Professor at Goa University, Goa. She worked as Research Fellow on UGC Major research project in the area of Mobility managements in 4G networks. She was the recipient of DST INSPIRE scholarship available for first rank holders in Post Graduate Degrees in Basic and Natural Sciences. She has published 6 papers in peer reviewed journals and international conferences and a chapter in an edited book published by IGI Global, an International publisher. Her current research interests include computer networks, Mobile communication and Cloud computing.