A Comparative Study of the Performance of Load Balancing Algorithms Using Cloud Analyst

Abdulnasir Younis Ahmad
College of Education for Pure Science, University of Mosul, Iraq.
E-mail: abdulnasir.younus@uomosul.edu.iq

Asmaa Y. Hammo
College of Computer Science and Mathematics, University of Mosul, Iraq.
E-mail: asmahammo@uomosul.edu.iq

Received September 29, 2021; Accepted December 21, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19328

Abstract

Cloud computing technology has grown and expanded widely in recent years. Cloud computing is a distributed system that may be on physical or virtual machines. Because of the ease of working in the cloud, its ease of use, it's being an interactive model and a highly profitable business model, many users have used it. This led to an increase in the load on it, and thus the overload has become a problem that must be solved. Many algorithms have been proposed to balance the load on cloud. The essential are three: Round robin, Equally Spread Current Execution (ESCE), and Throttled. This research investigates the performance of those algorithms with increasing of number of users. Previous researches did not take into account the increasing of users. CloudAnalyst simulation tool is used for this comparison. The comparison is done according to DC (Data Centre) processing time, overall total response time (ORT), processing cost, and hourly data center. It was found that initially with a certain settings and for a certain number of users, there may be no difference that can be mentioned in performance between the three algorithms. Then the improvement in performance increases with the increasing number of users. Throttled algorithm showed best performance in the average of ORT with increasing number of users.

Keywords

Load Balancing, CloudAnalyst, Round Robin, Equally Spread Current Execution (ESCE), Throttled, Performance, Response Time, Cloud Computing.

Introduction

In recent years, cloud computing technology has grown and expanded widely. It represents a business model through which any organization can make massive investments in computing infrastructure.

Cloud computing contains five basic characteristics according to the aforementioned definition: Firstly, customers have access to resources according to their requests. By communication, interfaces enable them to obtain the required services. These services may be infrastructure, platforms, or software. Second, resources are accessed through the network. Third, resources can be supplied to customers outside the location with large sizes. This can achieve economic benefit for the supplier without the need for the customer to pay attention to the geographical location of the resource pools. Fourth, these resources provide the customer with high flexibility, as he can obtain more or fewer resources according to his need. (G. Sinha & Sinha, 2020).

Two approaches can be taken to classify cloud computing: location-based and service-based. Cloud computing can be classified as public, private, hybrid, or community-based, depending on its location. Everyone has access to computing in public, and the infrastructure is placed on the site of the service provider's company. As for private computing, it is only available to a group of individuals and organizations. While hybrid is the result of interconnection in one way or another between the public and the private and are used for specific purposes based on organizational requirements. Finally, community-based consists of an infrastructure to which a set of institutions that share data and management participate.

Cloud computing can be divided into three categories based on the services it provides: infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). In the (IaaS) model, the cloud provides primary IT resources such as the network, computers, and control over computing resources. (PaaS) takes the burden of managing basic infrastructure (usually computers and operating systems) from organizations and allows users to focus solely on using applications. The latter, which is the (Saas), allows the user to focus on using a specific program and free users from thinking about managing the infrastructure and services. (Kumar, 2019).

Because of the ease of working in the cloud, its ease of use, it's being an interactive model and a highly profitable business model, many users have used it. This led to an increase in the load on it, and thus the overload has become a problem that must be solved. Here, the concepts of load balancing and its many algorithms appeared (Singh, 2020).

The technique of assigning a load to each node of a distributed system in order for it to perform faster and more efficiently is known as load balancing. It is a parallel system approach for achieving optimal system circumstances in which tasks are distributed evenly among processors, reducing program execution time. (Abhinav Chand et al., 2018).

As measuring the performance of load balancing is an important issue and challenging task in a cloud environment, cloud simulators appear to experiment on the real cloud. Many cloud simulators exist. One of them is Cloudsim. CloudaAalyst is an amplification to Cloudsim. It provides a GUI to the users for easy understanding. CloudAnalyst is used for the analysis of the large scale applications which are running on the cloud (Jose & Kumar, 2017).

In this research, we investigated the performance of the algorithms Round Robin, Equally Spread Current Execution (ESCE), and Throttled. We compare their performance in term of overall response time (ORT) in a situation where the number of users is increased, and also compare the outperforming of algorithms on others when the number of users rises.

This paper is organized as follows: Section 2 presents the related works on existing load balancing algorithms, Section 3, about the load balancing algorithms. Section 4 describes the CloudSimulator, Section 5, presents the Result and Performance Analysis, and Section 6, includes the Conclusion and Future work done in this paper.

Related Works

(Samal & Mishra, 2013) analyzes the various algorithms using CloudAnalyst as an analysis tool. The simulation was done using 6 user bases in different regions with 1000 users for each user base. Their result was the request time is the same in throttled and round robin. (Shoja, 2014) compares the two existing algorithms Round robin and throttled based on various parameters like response time and data processing time… etc. He announced that the response time, Overall response time, and Datacenter request servicing times are the same for the two algorithms. Only the cost for virtual machine usage per hour in throttled outperforms that in round robin algorithm. (Raghava & Singh, 2014) analyzed and compared various proposed algorithms addressing the issue of load balancing in Cloud Computing to provide a gist of the latest approaches in this research area. (Suguna & Barani, 2015) presented the results for the simulation assuming 6 user bases in six different regions with 1000 users in each user base. A comparison was done between ESCE and Throttled and showed a very small difference in the request service time between them. (Gangwar & Rana, 2014) compared the three existing algorithms for four user bases with 1000 users in each user base, and one data center. A simulation was repeated twice with changes in some factors. (Ghumman & Sachdeva, 2016) proposed a composite approach for load balancing using Equally Spread Current Execution (ESCE) and Throttled algorithms. The new algorithm was compared with each of ESCE and Throttled with different numbers of cloudlets to be announced that the proposed algorithm outperforms the other two in relation to average response time and earliest finish time. (Computing, 2017) simulated the three algorithms with three data centers, three user bases and with the default settings (i.e. 1000 users). It was found that round robin outperforms ESCE and Throttled by not more than 0.017%. In (Jose & Kumar, 2017), the existing algorithms were simulated in CloudAnalyst with six user bases and four data centers, all distributed in different regions. The simulation was executed once. It was found that ESCE algorithm have better response time with outperformance that not exceeds 0.01%. (S. & G., 2017) compare the three existing algorithms in CloudAnalyst with the default settings except using four data centers and six user bases altogether in one region. The simulation was done with the three existing service broker policies and it showed which load balancing algorithm did best with each policy. After reviewing load balancing and the three existing methods for the CloudAnalyst simulator, (Agarwal & Singh, 2019) discovered that the Throttled algorithm had a better overall response time than the other existing algorithms, but the data center time is similar. (Adityasaisrinivas et al., 2019) performed a comparison of the three existing algorithms. Algorithms were simulated in three situations. In each situation, they changed the numbers of data centers and user bases. They concluded that throttled outperformed the other two related to Average response time, average data center processing time, and cost. In (Adityasaisrinivas et al., 2019) the researchers performed the simulation with six user bases and six data centers, all spreads on six regions, and with 10000 users in each user base. They stated that the better load-balancing algorithm is throttled in relation to average response time, average data center processing time, and cost. (Singh, 2020) execute the simulation three times; once for each of the existing algorithms. The user bases were distributed in five regions. The performance was analyzed concerning average response time, hourly data center response time and the cost of Virtual Machine (VM) …etc. Simulation results indicated that throttled outperformed the others in all criteria except the cost which was the same for all. The Issues such as how performance and relative performance of algorithms change with a variant number of users was presented in this paper.

Load Balancing

Load balancing techniques redistribute the total load to specific node elements in a cloud pool. This distribution improves the use of resources and reduces the response time by sending the tasks to the lightweight nodes and avoiding sending them to the overburdened nodes, and as a result, the work is distributed evenly on the nodes (Shobha & Vivekanandreddy, 2018) The optimal use of resources, reducing response time, and reducing overload are all considered primary objectives of load balancing (Adityasaisrinivas et al., 2019).

To measure the performance for a load balancing algorithm, some metrics should be considered:

1. Throughput: It is the number of tasks completed per unit time. Increasing throughput leads to improved performance.
2. Overhead: It refers to the redundant cost needed to implement the algorithm. The less the overload, the better performance it is.
3. Migration time: To reduce the load, there is a need to migrate resources and tasks from one node to another. To increase performance, the time required for the migration process must be reduced.
4. Response time: It is the time between sending the request and receiving the response. To obtain better performance, this period must be reduced to a minimum.
5. Scalability: It is the adaptability of the algorithm to the increase of nodes in the system. Increasing this capability leads to improved performance. (Suguna & Barani, 2015).

The overall performance of the system is also affected by the complexity of the algorithm. It is also necessary to define and use an effective load scheduling system to reduce traffic and processing time. (Ramasubbareddy et al., 2019).

There are many cloud load balancing algorithms. three of them are originally existed and built in the CloudAnalyst tool. We will present these algorithms and compare them according to some important characteristics, such as DC (Data Centre) processing time, total response time, processing cost, and hourly data center.

A. Round Robin RR

The time quantum is used in round robin, which is the time slice assigned to the task to complete part of its work assigning the time slice to another task. The amount of this time is important as increasing the length of this period robs the round robin of its usefulness and efficiency and transfers it to performance similar to the first come first served. On the other hand, the severity of its shortness increases the overhead on the system. Therefore, determining the amount of time quantum is of great importance to the algorithm designer.

B. Equally Spread Current Execution Load (ESCR)

In this algorithm, a list of all virtual machines and their availability is maintained. When a request is received by the load balancer, the list is scanned to find the virtual machine that is suitable to grant the request. If it is found, then it is allocated to the request. The algorithm work on the principle of distributing the load equally on the virtual machines. To do so, the system keeps track of the allocation of each machine in the system in a current allocation table. The algorithm maximizes the throughput but it has its drawbacks such as its central failure and the looseness of the fault tolerance feature.

C. Throttled Load Balancing

This algorithm is based on the principle of assigning only one request at a time and queuing the others till accomplishing the first one (El Karadawy et al., 2020). In this algorithm, A VM index table is maintained by the load balancer. The state of the machine whether it is busy or free is indicated in the table (Singh, 2020). Initially, when the data center is requested for a VM, the request is transferred to the load balancer. The load balancer in turn scans its table to find the appropriate VM. The load balancer informs the DC when a virtual machine is found. So, DC sends the task to the specified VM and notify the load balancer to update the index table (Azmat, 2019). If no VM is detected, the load balancer sends a value to DC indicating that no VM is accessible and that the task is queued. (Singh, 2020).

CloudAnalyst tool

There are two methods of testing the algorithm in a cloud computing environment. The first was tested in a real environment like Amazon EC2 and the second was using a simulation tool that simulates a cloud computing environment. Many researchers resort to simulation because the real test limits the experiment to the range of potentials of the underlying environment and makes it difficult to reproduce results (Ramadhan et al., 2018). Also, measuring performance in the cloud environment is very difficult and takes a long time and access to the infrastructure requires money (G. Sinha & Sinha, 2020).

The simulation framework is characterized by the following:

1. Comprehensive cloud computing modeling and component installation support, including data centers, virtual machines, scheduling, and policy modification.
2. Support virtualization services; independent, and hosted on data center nodes.
3. Flexibility to replace execution policy between shared space allocation and time-sharing.

There are various cloud computing simulators that are used for evaluating the performance and security of cloud systems. CloudSim, its extensions such as CloudSimEx, WorkFlowSim, SimpleWorkFlow, CloudReports, and ClouudAnalyst, SPECI, OCT, EMUSIM…etc. (U. Sinha & Shekhar, 2015). A researcher can choose one that grants his requirements. As for our research, we choose the CloudAnalyst.

CloudAnalyst is a GUI tool that is based on Cloudsim architecture. The tool is very useful for researchers and designers of cloud systems. It is mainly used to simulate the virtual machine allocation using the load balancing algorithms. Using Cloudanalyst, main components of the cloud system such as the Internet, collection of users, data centers, and service brokers can be configured.

Cloud Analyzer has several features including:

1. Ease of use: the simulation does not require more than a keypress.
2. GUI- based output: Where the output is in the form of tables and charts that help in understanding and comparing the results.
3. Ability to repeat: The possibility of repeating the simulation with the same parameters to give the same results, so that the simulation results are not random.

Ability to save the results: The ability to save experiments with their inputs and results in files that can be transferred to other computers locations. (Ramadhan et al., 2018).

Several components exist in CloudAnalyst that make its usage easy and flexible. These components are: See Figure 1.

1. Geographically, the world is divided into six regions (0 to 5). Each one represents one of the six major continents: 0 represents North America, 1 represents South America, 2 represents Europe, 3 represents Asia, 4 represents Africa, and 5 represents Australia.
2. Internet: In which a suitable transmission latency and transfer latency delay around the world are introduced and can be configured.

3. Cloud Application Service Broker: This component selects which data center will handle requests from a specific user base. Service Proximity-based routing, Performance Optimized routing, and dynamically reconfiguring routing are the three types of service brokers implemented in CloudAnalyst.
4. User Base: This refers to a group of users who are responsible for generating traffic for the simulation. This group may consist of thousands of users and be considered as a single unit.
5. Internet Cloudlet: Represent the requested task generated by the user.
6. Data Center Controller: It is in charge of VM construction and removal, as well as the routing of User Base queries to the VMs.
7. VMLoad Balancer: According to a particular policy, It decides which cloudlet should be assigned to which virtual machine. As previously stated, CloudAnalyst provides three types of VMLoad Balancers: Round Robin Load Balancer, ESCE, and Throttled Load Balancer.
8. GUI: This component shows the simulator's graphical user interface and serves as a front-end controller. It allows users to set simulation parameters, save and load simulation configurations, run simulations, and save experiment results. (Nandwani et al., 2015) (Suguna & Barani, 2015).

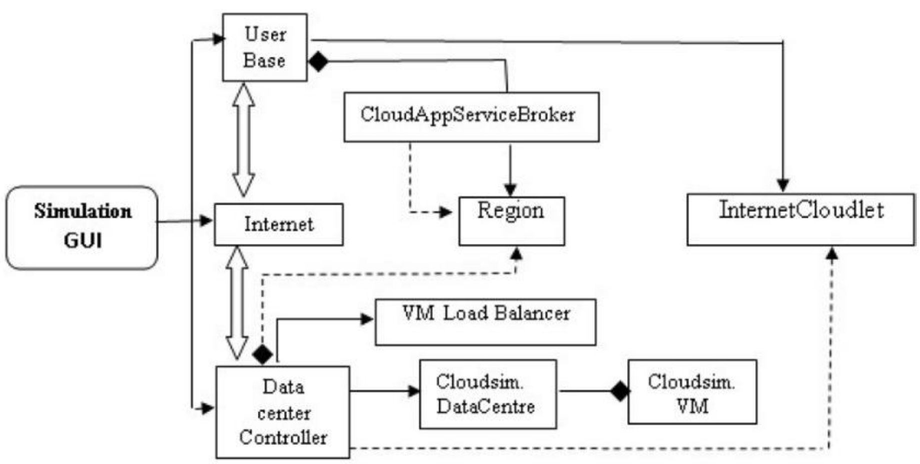

Figure 1 Main components of CloudAnalyst simulator

Performance Analysis

To compare the three algorithms, The CloudAnalyst tool was used. They were simulated with different numbers of users. Default settings of the tool for the data center, transmission delay, available bandwidth between regions, and costs were kept. User bases and data centers are both placed in the same region to neglect the effect of the geographical distance on the results. One data center and five user bases are all used in region (0), Figure (2).

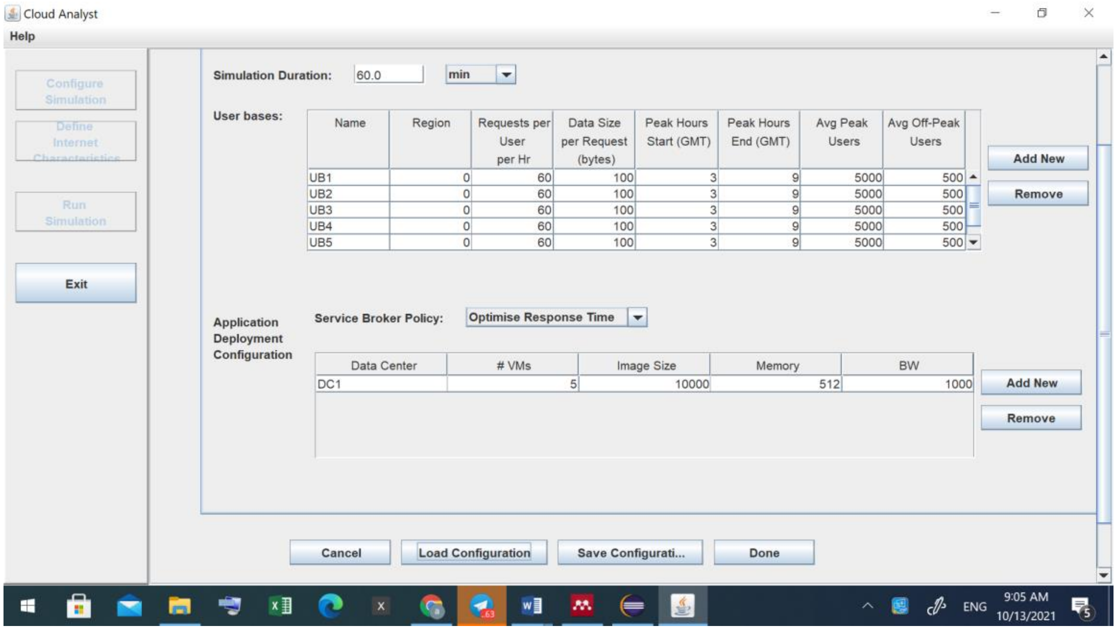

Figure 2 Graphical interface for setting the user bases

Initially, simulation was done with 5000 users in each user base (figure 3) and (tables 1,2). The number of users incremented constantly by 5000 users at a time until it reached 125000 users in each user base. Optimize response time used as a service broker policy. Figure 4 shows how is the average overall response time changes with the increasing number of users.

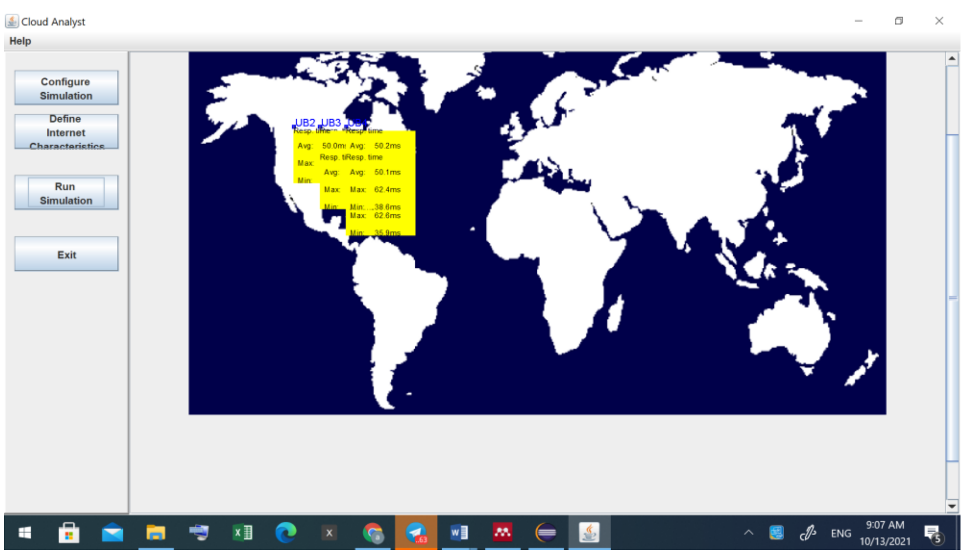

Figure 3 Graphical output screen

Table 1 Overall response Time for a simulation execution for 5000 users in each user base

	Avg(ms)	Min(ms)	Max(ms)
Overall response time	50.14	35.89	63.13
Data Center Processing Time	0.49	0.07	2.10

Table 2 Response time of all user bases for 5000 users in each user base

Userbase	Avg(ms)	Min(ms)	Max(ms)
UB1	50.11	39.14	62.40
UB2	49.96	38.64	62.40
UB3	50.20	38.38	63.13
UB4	50.22	38.39	63.09
UB5	50.15	38.65	62.37
UB6	50.19	35.89	62.64

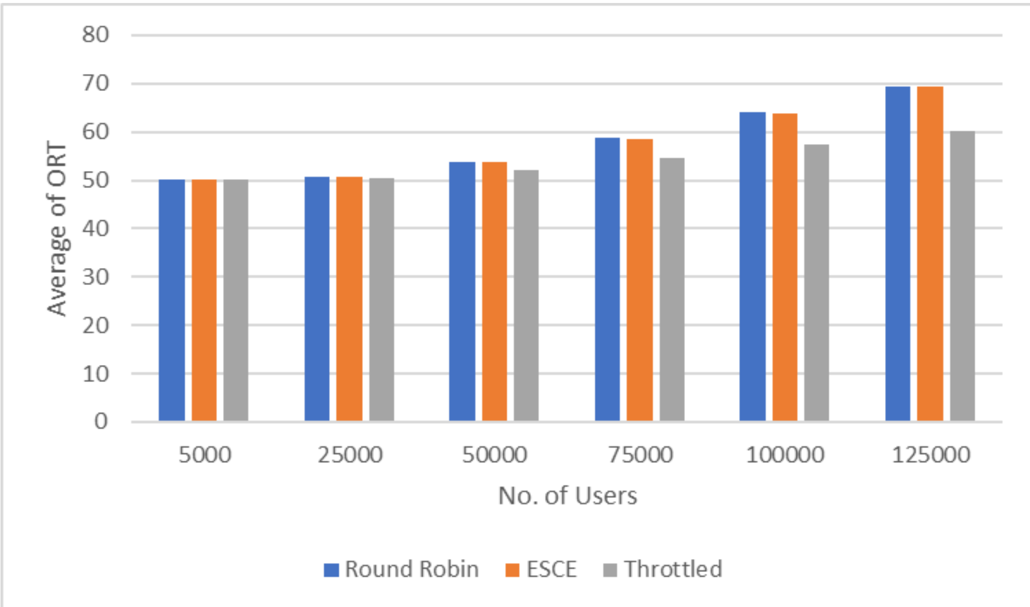

Figure 4 Changing of ORT avg. with no. of users

To calculate the percentage outperformance of an algorithm A on Algorithm B we used the formula:

$$\text{Percentage Outperformance} = ((ORT_{AlgB} - ORT_{AlgA})/ORT_{AlgA})*100$$

Figure (5) shows how an algorithm outperforms others with the increasing number of users.

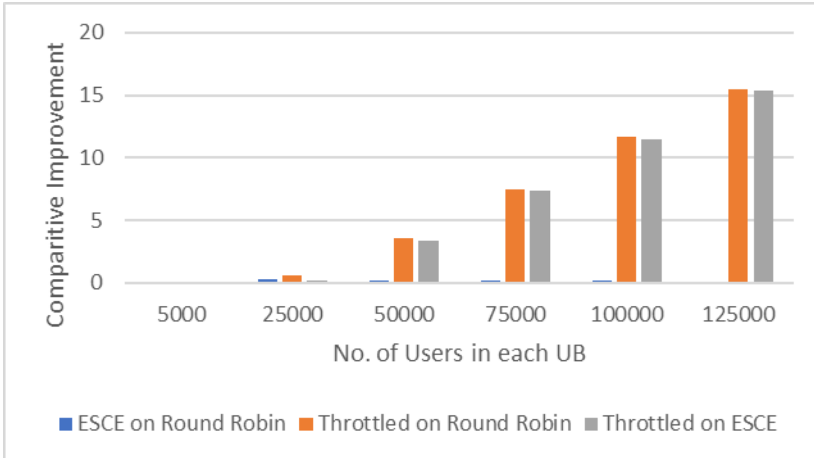

Figure 5 Percent outperformance of algorithms on each other

Figure (6) shows how is the average overall response time change with the increasing range number of users.

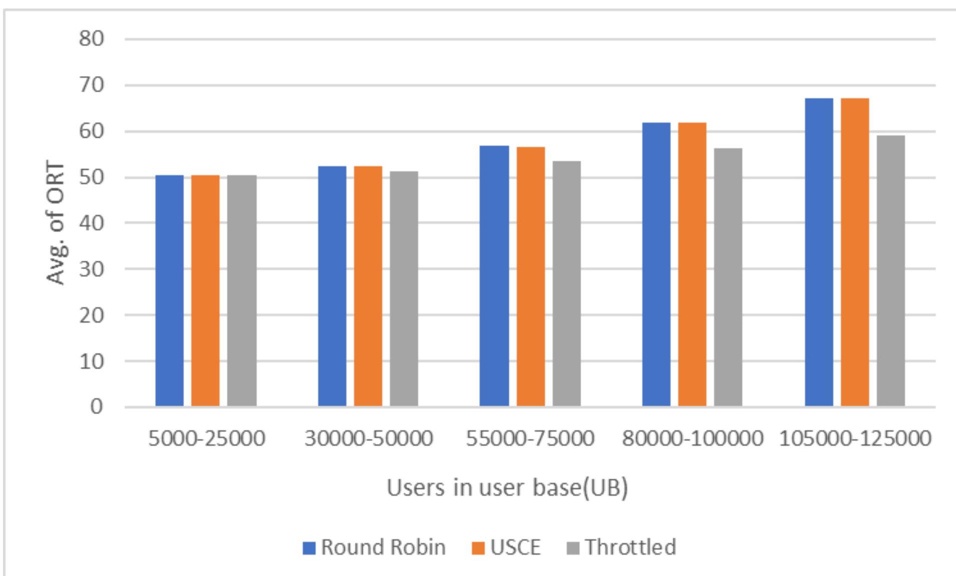

Figure 6 Changing of ORT avg. with ranges of no. of users

Figure (7) shows how an algorithm outperforms others with increasing ranges of the number of users.

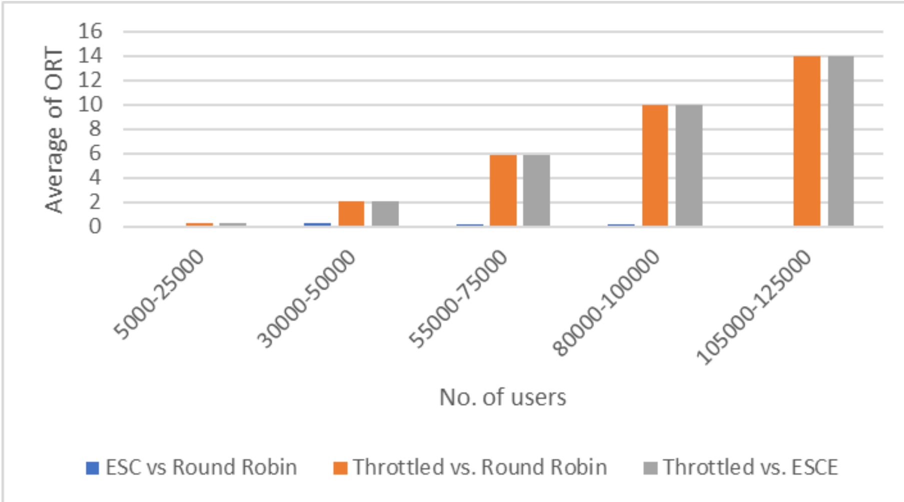

Figure 7 Percent outperformance of algorithms on each other for different ranges of users

Table (3) shows the outperformance of algorithms on each other.

Table 3 The outperformance of algorithms on each other

No. of users in each UB	Average of ORT (ms)			Percentage Improvement in Avg. of ORT		
	RR	ESCE	Throttled	ESCE to RR	Throttled to RR	Throttled to ESCE
5000	50.14	50.14	50.13	0	0.019948135	0.019948135
75000	58.73	58.63	54.63	0.170561146	7.505033864	7.321984258
125000	69.47	69.39	60.14	0.115290388	15.51380113	15.38077818

Conclusion

Comparing the work in this paper to previous related works, we realized that firstly, the numbers of users in user bases were rather small which shows very slight differences in results among the studied algorithms, secondly, the simulations were almost done very few times and very few cases in which the simulations did not show differences related to various changed cases, and thirdly, the geographical distance between datacenters and user bases that affects the average response time of the algorithms when scattering bases and datacenters randomly on the continents.

It was found through the results that initially (at less than 5000 users) there is almost no difference that can be mentioned in performance between the three algorithms. Then the improvement in performance increases with the increasing number of users. The percentage improvement in performance in ORT for ESCE's on RR was 0%, and 0.019948%, 0.019948% for Throttled on ESCE, and Throttled on RR respectively at 5000 users in each user base. This ratio is increased to 0.17056% for ESCE on RR, to 7.50503% and 7.32198% for Throttled on RR, and Throttled and ESCE respectively at 75000 users in each user base. This ratio reaches 0.11529 for ESCE on RR, to

15.513.80% and 15.38077% for the Throttled on RR, and the Throttled on ESCE respectively at 125000 users.

We conclude that the performance of the algorithm is not a constant value, and also the relative performance for different algorithms. They change related to the changed values of different parameters. To study and analyze aspects of cloud computing, the researcher has to know that many different parameters affect his work. Thus, he has to focus on not more than two parameters in one experiment.

As for future work, we propose the simulation to be done considering other affecting parameters such as data centers, number, and properties of virtual machines …etc.

References

Abhinav Chand, N., Hemanth Kumar, A., & Teja Marella, S. (2018). Cloud Computing based on the Load Balancing Algorithm. *International Journal of Engineering & Technology*, *7*(4.7), 131. https://doi.org/10.14419/ijet.v7i4.7.20528

Adityasaisrinivas, T., Govinda, K., Manivannan, S. S., & Swetha, E. (2019). *Analysis of Load Balancing Algorithms using Cloud Analyst*. *6*, 684–687.

Azmat, T. (2019). A Novel Approach Towards an Efficient Load Balancing Algorithm in Cloud Computing. *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 572–577.

Computing, C. (2017). *Performance Comparison of Load Balancing Algorithms using Cloud Analyst in Cloud Computing.* 509–513.

El Karadawy, A. I., Mawgoud, A. A., & Rady, H. M. (2020). An Empirical Analysis on Load Balancing and Service Broker Techniques using Cloud Analyst Simulator. *Proceedings of 2020 International Conference on Innovative Trends in Communication and Computer Engineering, ITCE 2020*, 27–32. https://doi.org/10.1109/ITCE48509.2020.9047753

Gangwar, I., & Rana, P. (2014). *Juxtaposition of Load Balancing Algorithms in Cloud Computing using Cloud Analyst Simulator*. *97*(2), 21–26.

Ghumman, N.S., & Sachdeva, R. (2016). An Efficient Approach for Load Balancing in Cloud Computing using Composite Techniques. *International Journal of Research in Engineering and Applied Sciences*, *6*(2), 145–149. http://euroasiapub.org/wp-content/uploads/2016/09/15EASFeb-3245-1.pdf

Jose, M., & Kumar, D.S. (2017). *Performance Analysis on Load Balancing by Roundrobin, Throttled & Active Monitoring Algorithms Using Cloudanalyst*, *5*(6), 92–100.

Kumar, P. (2019). *Issues and Challenges of Load Balancing Techniques in Cloud Computing : A Survey Issues and Challenges of Load Balancing Techniques in Cloud Computing : A Survey*. *February*. https://doi.org/10.1145/3281010

Nandwani, S., Achhra, M., Shah, R., Tamrakar, A., & Raksha, P. S. (2015). *Analysis of Service Broker and Load Balancing in*. *4*, 92–98.

Raghava, N.S., & Singh, D. (2014). *Comparative Study on Load Balancing Techniques in Cloud Computing*, *1*(1).

Ramadhan, G., Purboyo, T.W., Latuconsina, R., & Robin, A.R. (2018). *Experimental Model for Load Balancing in Cloud Computing Using Throttled Algorithm*, *13*(2), 1139–1143.

Ramasubbareddy, S., Adityasaisrinivas, T., Govinda, K., Manivannan, S.S., & Swetha, E. (2019). Analysis of load balancing algorithms using cloud analyst. *International Journal of Recent Technology and Engineering*, *7*(6), 684–687.

Sutagatti, S.S., & Kulkarni, S.G. (2017). Comparative Analysis and Evaluation of Load Balancing Algorithms. *International Journal of Computer Applications*, *171*(5), 6–11. https://doi.org/10.5120/ijca2017915031

Samal, P., & Mishra, P. (2013). *Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing*, *4*(3), 416–419.

Shobha, D.R., & Vivekanandreddy, P. (2018). Implementation of Load Balancing Algorithms in Cloud Computing Environment using Cloud Analyst Simulator. *International Journal of Recent Trends in Engineering and Research*, *4*(7), 22–27. https://doi.org/10.23883/ijrter.2018.4351.o8zop

Shoja, H. (2014). *A Comparative Survey on Load Balancing*.

Singh, A. (2020). Performance Evaluation of Load Balancing Algorithms Using Cloud Analyst. *10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 156–162.

Sinha, G., & Sinha, D. (2020). Enhanced Weighted Round Robin Algorithm to Balance the Load for Effective Utilization of Resource in Cloud Environment. *EAI Endorsed Transactions on Cloud Systems*, *6*(18). https://doi.org/10.4108/eai.7-9-2020.166284

Sinha, U., & Shekhar, M. (2015). Comparison of Various Cloud Simulation tools available in Cloud Computing. *Ijarcce*, *4*(3), 171–176. https://doi.org/10.17148/ijarcce.2015.4342

Suguna, D.S., & Barani, R. (2015). Simulation of Dynamic Load Balancing Algorithms. *Bonfring International Journal of Software Engineering and Soft Computing*, *5*(1), 1–7. https://doi.org/10.9756/bijsesc.8061