The Lighthouse Journal of Computational and Numerical Sciences, Vol. 01, Issue 1, Winter 2022, pp. 1-12

An Optimized Active Monitoring Load Balancing (OAMLB) Algorithm using Cloud Analyst Tool in Cloud Computing

Mahnoor Imran[1], Asmara Jadoon[2]

ABSTRACT

Cloud Computing (CC) is a quickly developing and extended technology that grant on request assistance to the users. Many organizations are embracing cloud technology as a consequence of its promising features like security, flexibility, efficiency, reasonable cost, scalability, freedom from backup plan etc. Cloud-Computing (CC) is growing rapidly, but it also faces some challenges, like reliability, resource allocation, data management, load balancing, fault tolerance, failure avoidance etc. Load balancing is one prominent research topic for researchers in the field of distributed computing. Many researchers proposed various techniques and approaches for managing the users' requests efficiently. The primary goal of this paper is to enhance the "Active_Monitoring_Load_Balancing (AMLB) Algorithm." The major downside of the AMLB algorithm is that it does not check the capacity of virtual machine (VM), it only checks their status (available/busy); and assigns the load to the available virtual machine (VM) regardless of the load that is accomplished by the available VM. This led to the problem of over utilization of the available VM. The proposed technique/algorithm named as "Optimized Active Monitoring Load Balancing" (OAMLB) algorithm overcomes the drawback of existing one. The proposed algorithm is run and tested in cloud analyst using various cloud analyst parameters. The experiment and analysis of OAMLB shows that it has 61% improved performance in regard to both "response time" and "data center processing time" with comparison to "Round Robin" algorithm.

Keywords: "*Cloud-Computing(CC), Cloud Analyst, Load balancing, Active Monitoring Load Balancing, VM, Response time (RT), Data Center processing time (DCPT)"*

INTRODUCTION

The term cloud referred to as the Network or Internet, which is presently located remotely. Cloud-Computing(CC) is pay-as-you-utilize model to clients regardless of their area and location [1]. The definition given by "NIST: Cloud-Computing(CC) is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or

[1] Dept. of Computer Science, Govt. Girls Degree College No. 01 Abbottabad, KP, Pakistan, **Corresponding Author's Email: asmarajadoon@gmail.com**

Manuscript Submitted: May 25, 2022
Manuscript Accepted: Nov 17, 2022

service provider interaction" [2]. Cloud-computing(CC) is an extended form of distributed computing [3]. Cloud model is composed of five key characteristics including the "on-demand self-service, broad network access, resource pooling, and rapid elasticity and measured Service" [2] [4]. Fig. 1 illustrating the Cloud-computing (CC) environment.

Figure 1: Cloud-Computing Environment (CCE)

Cloud-Computing (CC) encompassed three basic models [2] named as "Platform-as-a-Service (PaaS)" is the first model, "Infrastructure or Hardware-as-a-Service (IaaS/Haas)" is the second model and the last model is "Software-as-a-Service (SaaS)".

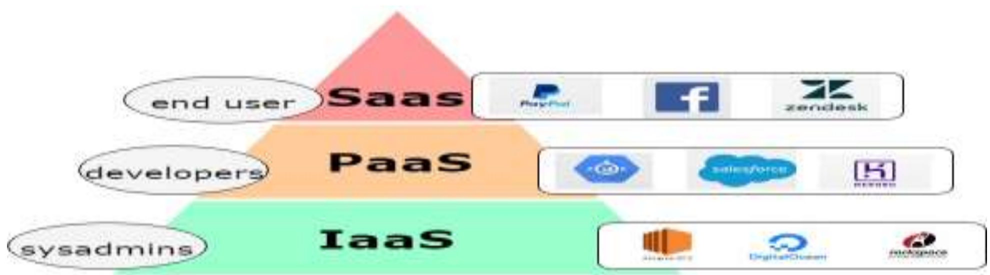

Figure 2: Service Models of Cloud-Computing (CC)

Cloud provides four deployment models that are represented in figure 3. Public, Private, Community and Hybrid cloud.

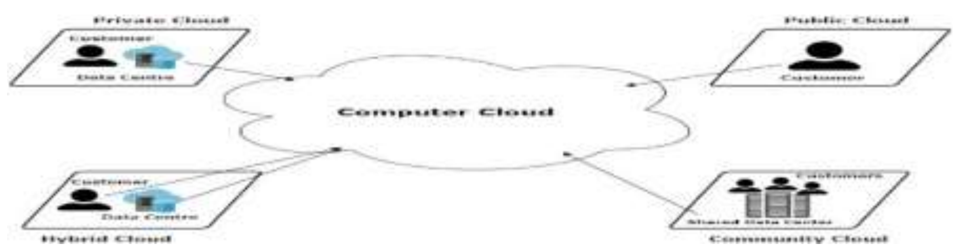

Figure 3: Deployment Models of Cloud-Computing (CC)

The primary benefits of Cloud-Computing(CC) are: less investment cost, upgraded performance, unlimited storage space, data backup and restoration, convenient availability of information, quick deployment, simplified scalability of services, deliver new services etc. [4].

The techniques/algorithms/polices used in cloud environment includes "Round Robin" (RR) algorithm, "Equally Spread Current Execution (ESCE)", "Throttled" and proposed algorithm named as "Optimized Version of Active_Monitoring_Load_Balancing (OAMLB) algorithm". The descriptions of the following load balancing (LB) algorithms are as follows:

Round Robin (RR) algorithm

"Round robin (RR)" algorithm [9] executes processes in a circular order based on their arrival time for fixed time quantum. All processes in the ready queue execute the process for this fixed time quantum. If a process's burst time (BT) is less than the set time quantum (TQ), then the subsequent process in the queue begins its execution[18] [20]. It is one the simplex scheduling techniques. Round Robin Algorithm make use of the principle of time scheduling [20].

Equally Spread Current Execution Algorithm

This algorithm redistributes the workload by evaluating the resources and allocates the load to the virtual machine (VM) with lower usage, effectively balancing the load across the systems. The load balancer maintains the queue [25]. This algorithm randomly checks the size of the load coming and works continuously by allocating the load to the Virtual Machines (VM) that is lightly loaded [25]. It is a dynamic load balancing (LB) algorithm that decide the priority by inspecting the size of the process. The load balancer examines the incoming jobs and then assigns these jobs to servers that are available. The load balancer also updates the list of the incoming jobs [21]. Here, major goal of data center is to communicate the user request to the Virtual Machine (VM) identified by the id [7].

Throttled Load Balancing Algorithm

It is an effectual approach [10], where users request are given to the Data Centre Controller (DCC) [21]. "Throttled" VM Load Balancer takes responsibility of keeping a virtual machine list and their states whether available or busy [22]. First all virtual machines states are set available. When DCC gets the users request it asks VM load balancer about virtual machine [5]. VM Load Balancer (VB) checks the VM index table from starting and sees which virtual machine (VM) can handle that particular load or index table is fully scanned. If appropriate virtual machine found then only VM load balancer returns the id of that particular virtual machine to Data Centre Controller (DCC). Then DCC allots the incoming job/requests to that particular virtual machine recognized by id [23] [20]. After that Vm Load Balancer modifies the allocation table and if appropriate Virtual Machine(VM) not found then Vm Load Balancer returns minus 1 (-1) value to DCC and DCC put the request in queue [18]. After finishing the processing of the allocated request response cloudlets are sent to DCC which in return send notification for de-allocation [24].

Load balancing (LB) is the procedure of distributing the load on the various nodes which provides best resource utilization when nodes are overloaded or under loaded with jobs. Load balancer is required for efficient load balancing as it keeps the index table, that keeps the information of status of VM i.e., whether the VM is Busy/Available, of the available VMs. Number of techniques like "Round robin" Algorithm, "Active Monitoring Load Balancing Algorithm" and "Throttled Algorithm" etc., have been proposed to overcome the problem of load balancing. Many researchers have done their work for balancing the load efficiently.

The motivation of this research work is to determine the best technique for the distribution of load, which is appropriate for the requested job with respect to different performance parameters i.e., average response time, datacenter processing time etc., so, as to reduce the network traffic by balancing the load efficiently. The key contribution of this study is to present an optimized version of AMLB algorithm by assigning the load to the least loaded VM in order to avoid the overutilization of VM. Moreover, to compute the response time of the proposed algorithm i.e., “Optimized Active Monitoring Load Balancing”(OAMLB) Algorithm for the request of the user. Furthermore, to calculate the response time of the user request to the “Round robin” algorithm.

The paper is categories as follows: Section II explains the work presented by different authors in the field of cloud regarding load balancing. Section III explains the problem statement of our research work. Section IV briefs about the proposed methodology and technique. Simulation setup and modeling i.e., application and resource modeling is explained in V. Section VI demonstrate the results and comparative analysis. Section VII presents the conclusion and outline potential of future research.

LITERATURE REVIEW

Cloud Computing (CC) is the rapidly growing technology that gives many services and advantages to the users. Cloud-Computing (CC) provides many of the benefits in case of performance matrices to its users but increasing number of consumers results in an increasing traffic that effects the balancing of the load. In Cloud-Computing (CC), one of the primary challenges is “load balancing”. Many techniques/algorithms have been proposed to address the issue of load balancing. Numerous researchers have conducted extensive research in this field to enhance cloud technology, scaled with different parameters including response time, waiting time, CPU utilization etc. Some of the work done in this field by many researchers is as follow.

Narale et.al., [5], proposes the algorithm that mainly focus to minimize cost for data center transferring, cost for total Virtual Machines (VMs), Data Center Processing Time (DCPT) and Response Time (RT). This study is undertaken for the hybrid cloud by adding user bases from public and private cloud. And the results shows the great reduction in response time, data processing time and cost. Ramadhan et.al [6], proposed experimental load balancing model considers factors such as overall response time, request servicing, datacenter loading and tariff details of VMs obtained from the simulator to generate its output. The Cloud Analyst tool integrates the “Throttled” algorithm, which is commonly used by load balancers in cloud environments, where the response time is between the average ranges between the Ub1, the sources where the user’s requests are sent to server, and other.

Keswani et.al [7], perform the critical analysis of algorithms using the Cloud Analyst tool. He included following algorithms for his study i.e., “Round robin”, “Throttled” and Equally-Spread-Current-Execution algorithms. The simulation is carried out using Eclipse Neon.3, JDK 1.8. This algorithm is implemented for IaaS model in cloud environment and results shows the significant outcomes in terms of response time, data center processing time and total cost. The conclusion that comes to an end is that the “Throttled” algorithm is better among all. Phi et.al., in 2018 [8], proposes the Throttle Modifies Algorithm (TMA) to increase the response time of virtual machine in the cloud on cloud to deliver best services to end-users. This algorithm is implemented in Cloud Analyst tool of Cloud Sim Toolkit and it improves the response time (RT) and processing time (PT) of cloud data center (DC). Fatima et.al., [9], Proposed a hybrid algorithm that include both the properties regarding “Throttled” and “Equally Spread Current Execution algorithms”. The major goal of this algorithm is to overcome all the downsides of the “Throttled” and the current execution algorithms.

The above algorithm maintains the index table of the virtual machines and also the state of the VM’s like they are busy or not. Here the data center allocation and there cost is more accurate than others. Meftah et.al., [10], proposed a model for studying the consequence of the service broker policies under the different configurations of the data centers(DC). “Round robin” (RR), “Equally Spread Current Execution” and throttle were used for the study. The evaluation of Load Balancing (LB) algorithm will be based on the overall “response time (RT)” of application, as well as, the “total processing time” spent by data centre in handling the requests.

Falisha et.al., [11], discusses the different LB algorithms used in Cloud-Computing(CC) and thus determines the overall cost, requesting services and overall response time in virtual machines to check which algorithm works best to handle the load. The results concluded that the ESCE algorithm handles the load in the Cloud-Computing (CC) in terms of average response rate. Alworafi et.al [12], presented “An Enhanced Task Scheduling Deadline-Aware based model” to lessen the makespan and escalating the utilization of resources within the defined deadline constraint. The proposed technique arranges jobs in ascending order based on their priority length. The presented technique improves the performance by lessen the “average of makespan, the mean of total average response time, the number of violations, the violation ratio and the failure ratio”.

Rani et.al., [13], presented the technique that has combined features of “Round robin”, “Throttled” and ESCE algorithms. Goal of this paper is to decrease the overall response time and data center processing time. Banerjee et.al [14], proposed Generic RR algorithm for real time system for adjusting time quantum dynamically so that the generic algorithm improve performance by reducing “average waiting time, turnaround time and number of context switches” as compared to RR, also to overcome the operating system overhead. Indusree et.al., [15], proposed an “Enhanced Round robin Scheduling Algorithm” has been proposed, incorporating a burst-time based time quantum calculation method that determines the time quantum on the burst time of processes already in the ready queue. According to results, the “Enhanced Round Robin Scheduling Algorithm” yields a reduction in the turnaround time, the waiting time and a decrease in the context switching in comparison to RR algorithm.

Problem Formulation

Cloud environment provides better services to the users. With the increase in the number of users on cloud, the existing resources hosted on the cloud will become unavailable to some extend which results in the problem of delay between user and "Cloud Service Provider (CSP)", so it becomes necessary to balance the load in such a way that none of the resource is overloaded nor under loaded due to which there is more chances of failure occurrence and wastage of resources. The major purpose of this research work is to maximize the performance of "Active Monitoring Load Balancing Algorithm", whereas the major downside is, it doesn't check the capacity of the VM instead it checks VM status and assigns load regardless of load already assigned to the VM.

Proposed Algorithm/Technique

This paper presents the improved version of AMLB algorithm named as "Optimized Active Monitoring Load Balancing (OAMLB) algorithm". AMLB algorithm is the load balancing algorithm and it balances the load by assigning the load to the available VM, but, its major disadvantage is that it doesn't check the capacity of VM instead it checks the status of the VM i.e. available/busy. This lead to the problem of over utilization of VM. In order to overcome this problem this research paper presents the new approach for balancing the load efficiently i.e. OAMLB. The proposed algorithm checks the capacity of VM instead of state of VM. First user submits request, these requests are made for Twitter and this data is considered as input data for processing. These requests are passed on to the datacenter controller, contains the physical components for storage, datacenter controller receives the requests and passed these requests to the load balancer named as "Optimized Active Monitoring Load Balancing Algorithm"(OAMLB). Here, the working of proposed algorithm starts. Load balancer maintains the index table of all available VMs. When the user requests are passed on to datacenter controller, it first selects VM randomly, the status (available/busy) of the selected Virtual Machine (VM) is checked, if it is busy then load balancer selects any other VM. If not then load balancer checks the current allocations of that selected VMs, number of requests that are currently accomplished by the selected Virtual Machine (VM). If number of requests accomplished by the VM is less than average allocations of the selected VM then the request is assigned to that VM. If number of requests accomplished by the Virtual Machine (VM) is greater than average allocations of the selected VM then the request is allocated to any other VM having less average allocation so as to balance the load in such a way that neither of the VM is over loaded nor under loaded.

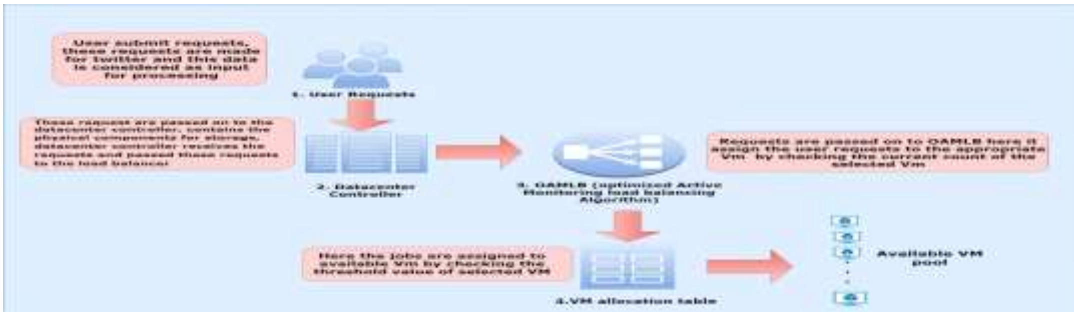

Figure 4: Proposed OAMLB Algorithm

Modeling and Simulation Setup

In this scenario, we are using Cloud Analyst with Java programming in Eclipse for hardware used in the form of Intel Corei5 with 8 GB RAM. **Cloud Analyst** [7] [8] is a GUI based tool based on the Cloud Sim architecture. It has been developed at University of Melbourne. In this simulation there are 20 user bases [1] that holds the user requests and 5 datacenters and each datacenter has 25 VMs [16].

Resource Modelling

Table 1 shows the assumption at the level of virtual machine specification and physical machine that is used in this simulation.

Table 1: Physical parameters of Data Center

Parameter of Datacenter	Values(used in Simulation)
VM-image size	10000
VM-Bandwidth	1000
VM-Memory	512
VMM of Datacenter	Xen
OS of Datacenter	Linux
Memory per machine of DC	204800
Storage per machine of DC	100000000
Grouping Factors on Request	400
Executable instruction Length	500
Available per BW of DC	1000000
Processing speed of DC	10000

Application Modelling

The number of internet users increases in the evening from 7 p.m to 10 p.m. The number of social media users are recorded up to 2.5 billion, recently. We considered the traffic load of popular social networking site like Facebook in our simulation. Active users recorded over 2.45 billion in 2019 [17]. In our simulation, we observed that the most users use Facebook in evening from 7:00-9:00 pm.

Table 2: Assumption of Application Usage

Regions	User(million)
“North America” (R0)	247
“South America” (R1)	271.1
“Europe”(R2)	385
“Asia”(R3)	396
“Africa”(R4)	139
“Oceania”(R5)	15

RESULTS AND DISCUSSIONS

Response time:

For each user base the “response time, hourly based response time and the overall response time” computed by Cloud Analyst for each load balancing technique. The graph indicates that there is decrease in “average response time” of both “Round robin” and “Optimized Active Monitoring Load Balancing Algorithm” as the no. of VMs increases. But, graph clearly specifies that the on increasing number of VMs average response time of OAMLB is much smaller as compared to the RR algorithm. It means that OAMLB algorithm increases the performance up to 61% in regards to “average response time”.

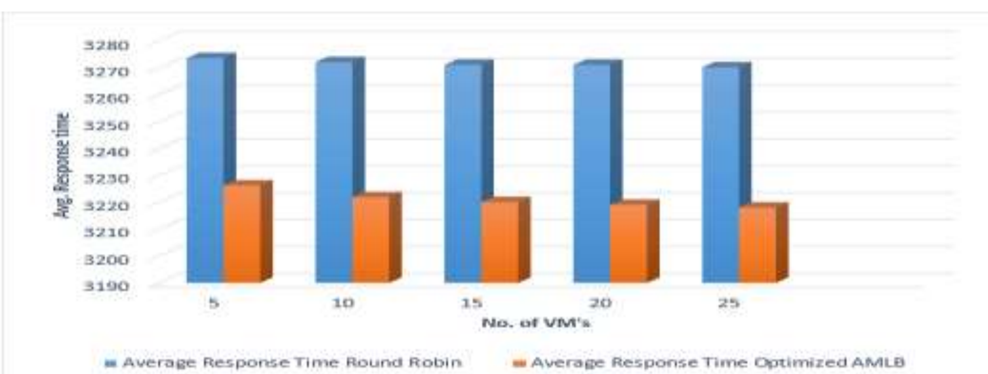

Figure 5: Comparison of Response time (RT) of RR & OAMLB

Average Datacenter Processing Time (DCPT):

Graph below depicts the results on the basis of average response time. On X-axis number of VM’s are mentioned and on Y-axis average response time is mentioned. It is evident from graph as the number of VM’s multiplies average response time of both “Round robin” and “Optimized Active Monitoring Load Balancing Algorithm” decreases i.e. number of VMs has inverse relation with “response time”, when the number of VMs increases average response time decreases. But, graph clearly specifies that the on increasing number of VMs average response time of OAMLB is much smaller as compared to the RR algorithm. It means that OAMLB algorithm increases the performance up to 61% in terms of "average response time".

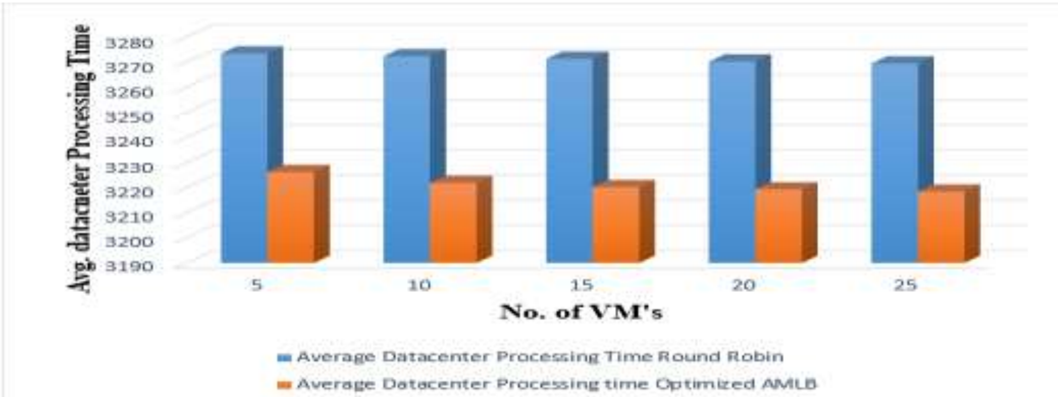

Figure 6: Comparison of Avg. DC processing of RR & OAMLB

Response Time by Region:

Below graph represents the average response time(RT) of each user base of both existing algorithms i.e., "Round robin" and proposed Optimized Active Monitoring (LBA). It clearly specifies that the response time of the proposed OAMLB algorithm is lesser than RR in each region.

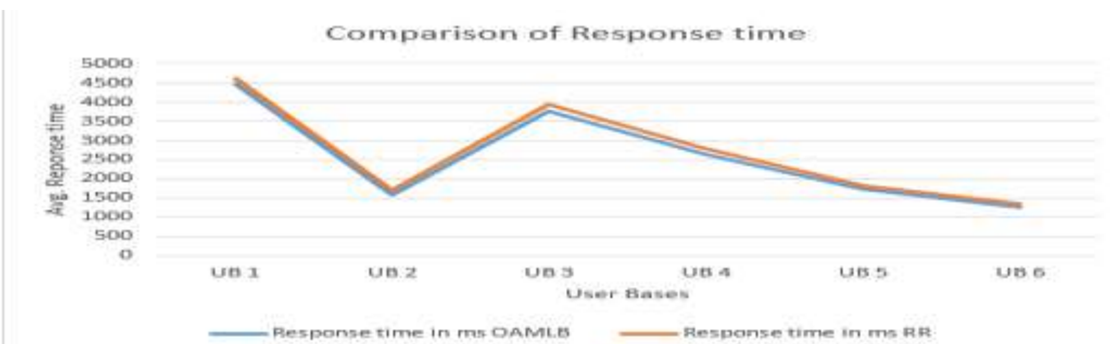

Figure 7: Comparison of Response time of OAMLB & RR on hourly bases in each user base

Comparative analysis

Two load balancing techniques/algorithms named as "Round robin" (RR) and "Optimized Active Monitoring Load Balancing" (OAMLB) are compared in this paper. Table 4 demonstrates the comparative analysis of two algorithms i.e. "Round robin" (RR) and Optimized AMLB algorithm and the results simulated by cloud analyst tool.

Table 4: Comparison Results of RR and OAMLB

Algorithms	**"Round robin" (RR)**		**Optimized AMLB (OAMLB)**	
No. of VM's	**Average Response Time**	**Avg. DC processing Time**	**Average Response Time**	**Avg. DC processing Time**
5	3273.55	3273.55	3226.24	3226.24
10	3272.11	3272.43	3221.87	3221.57
15	3271.09	3271.39	3220.06	3220.26
20	3271.04	3270.34	3219.03	3219.33
25	3270.05	3269.55	3218.03	3218.43

CONCLUSION AND RECOMMENDATIONS

Cloud-Computing (CC) is the immense area of research and the prominent research topic is load balancing so this study primarily focuses on the load balancing. Load Balancing (LB) aims to enhance satisfaction of users and optimizing utilization of resource, while simultaneously minimizing response time and reducing the number of job rejection. Several algorithms/techniques for balancing the load efficiently have been proposed. We have simulated two load balancing (LB) algorithms for the execution of the user requests in the cloud environment. The performance of the algorithms has been observed using the various scheduling parameter like "Response time, DC processing time and Cost". In this paper, according to the experiment and analysis the proposed Optimized AMLB algorithm has 61% improved performance in regards to both Response Time (RT) and DC Processing Time (DCPT) with compared to existing "Round robin" (RR) algorithm. In future, more work on Optimized AMLB algorithm can make the proposed technique more ideal in cloud environment.

REFERENCES

[1] Elmakki, H. T. E., & Noureldien, N. A. (2018). *Comparison Between Three Load Balancing Algorithms in Cloud Computing Environment* (Master's thesis, University of Science & Technology).

[2] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. [3] S. Garg, R. K. Dwivedi, and H. Chauhan, "Efficient Utilization of Virtual Machines in Cloud Computing using Synchronized Throttled Load Balancing," no. September, pp. 4–5, 2015.

[4] Apostu, A., Puican, F., Ularu, G., Suciu, G., & Todoran, G. (2013). Study on advantages and disadvantages of Cloud Computing–the advantages of Telemetry Applications in the Cloud. *Recent advances in applied computer science and digital services*, *2103*.3.

[5] Narale, S. A., & Butey, P. K. (2018, April). Throttled load balancing scheduling policy assist to reduce grand total cost and data center processing time in cloud environment using cloud analyst. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 1464-

1467). IEEE.

[6] Ramadhan, G., Purboyo, T. W., & Latuconsina, R. (2018). Experimental model for load balancing in cloud computing using throttled algorithm. *International Journal of Applied Engineering Research*, *13*(2), 1139-1143.

[7] B. Keswani, J. Rathore, and V. S. Rathore, "ISSN (O): 2455-5738 GADL Journal of Inventions in Computer Science and Communication Technology (JICSCT)," no. 1, pp. 6–11, 2018.

[8] Phi, N. X., Tin, C. T., Thu, L. N. K., & Hung, T. C. (2018). Proposed load balancing algorithm to reduce response time and processing time on cloud computing. *Int. J. Comput. Netw. Commun*, *10*(3), 87-98.

[9] Fatima, N., & Anugrah, S. (2018). Naïve shared based approach of load balancing named as hybrid algorithm in cloud computing. *IJSRCSEIT*, *3*(1), 262-269.

[10] Meftah, A., Youssef, A. E., & Zakariah, M. (2018). Effect of service broker policies and load balancing algorithms on the performance of large scale internet applications in cloud datacenters. *International Journal of Advanced Computer Science And Applications*, *9*(5).

[11] Falisha, I. N., Purboyo, T. W., Latuconsina, R., & Robin, A. R. (2018). Experimental model for load balancing in cloud computing using equally spread current execution load algorithm. *International Journal of Applied Engineering Research*, *13*(2), 1134-1138.

[12] Alworafi, M. A., Dhari, A., El-Booz, S. A., Nasr, A. A., Arpitha, A., & Mallappa, S. (2019). An enhanced task scheduling in cloud computing based on hybrid approach. In *Data Analytics and Learning* (pp. 11-25). Springer, Singapore.

[13] Rani, S., Saroha, V., & Rana, S. (2017). A hybrid approach of round robin, throttle & equally spaced technique for load balancing in cloud environment. *Int. J. Innov. Adv. Comput. Sci. (IJIACS)*, *6*(8), 2347-8616.

[14] P. Banerjee, R. Shree, and R. K. Verma, "Generic Round Robin Scheduling for Real Time Systems," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 5, pp. 148–155, 2017.

[15] Indusree, J. R., & Prabadevi, B. (2017, November). Enhanced round robin CPU scheduling with burst time based time quantum. In *IOP Conference Series: Materials Science and Engineering* (Vol. 263, No. 4, p. 042038). IOP Publishing.

[16] Panwar, R., & Mallick, B. (2015, October). Load balancing in cloud computing using dynamic load management algorithm. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)* (pp. 773-778). IEEE.

[17] "Facebook users by country 2019 Statista." [Online].Available: https://www.statista.com/statistics/268136/top-15-countries-based-on-number-of-facebook-users/. [Accessed: 28-Feb-2020].

[18] Patil, L., & Patil, N. N. (2015). International Journal of Modern Trends in Engi-

neering and Research. *Restoration*, *2*(03).

[19] Rani, S., Saroha, V., & Rana, S. (2017). A hybrid approach of round robin, throttle & equally spaced technique for load balancing in cloud environment. *Int. J. Innov. Adv. Comput. Sci.(IJIACS)*, *6*(8), 2347-8616.

[20] Li, D. C., Wu, C., & Chang, F. M. (2005). Determination of the parameters in the dynamic weighted Round-Robin method for network load balancing. *Computers & operations research*, *32*(8), 2129-2145.

[21] Sharma, T., & Banga, V. K. (2013). Efficient and enhanced algorithm in cloud computing. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, 2231-2307.

[22] Domanal, S. G., & Reddy, G. R. M. (2014, January). Optimal load balancing in cloud computing by efficient utilization of virtual machines. In *2014 sixth international conference on communication systems and networks (COMSNETS)* (pp. 1-4). IEEE.

[23] Dash, M., Mahapatra, A., & Chakraborty, N. R. (2013). Cost effective selection of data center in cloud environment. *International Journal on Advanced Computer Theory and Engineering (IJACTE)*, *2*, 2319-2526.

[24] Hashem, W., Nashaat, H., & Rizk, R. (2017). Honey bee based load balancing in cloud computing. *KSII Transactions on Internet and Information Systems (TIIS)*, *11*(12), 5694-5711.

[25] Li, T., Baumberger, D., & Hahn, S. (2009). Efficient and scalable multiprocessor fair scheduling using distributed weighted round-robin. *ACM Sigplan Notices*, *44*(4), 65-74.