BAUET JOURNAL
Published by
Bangladesh Army University of Engineering & Technology (BAUET)
Journal Homepage: https://journal.bauet.ac.bd/
DOI: https://doi.org/10.59321/BAUETJ.V4I2.14

An Evaluation of an Advanced Throttled Algorithm for Load Distribution in the Cloud Computing Environment

Tanjina Jahan[1*], Mohammod Abul Kashem[1], Md. Toufecul Islam[1]

[1]Dept. of Computer Science and Engineering, Dhaka University of Engineering and Technology (DUET), Gazipur-1707, Bangladesh

Abstract: The platform for cloud computing is also known as an Internet-based computing method where work is done without restrictions. In cloud computing, load-balancing is essential to spread the dynamic workload equally among all devices. In the present situation, load-balancing solutions have to be very efficient in allocating requests and ensuring appropriate hardware and software usage to avoid inadequate resources. It has become a major concern in the context of cloud computing. The goal of this study is to reduce processing and response times by introducing a load-balancing method. We suggest this method to increase the processing time of jobs to ensure it can help optimize the load-handling ability of cloud computing. This method is an enhancement of the throttled method and is called the Advanced Throttled Algorithm (ATA). It compares the output of the proposed algorithm to the throttled and round-robin algorithms. This study will present the results of these three algorithms. The suggested approach spreads the load evenly across virtual machines, as shown by the simulation outcomes. The recommended technique has enhanced the cloud data center's response and processing times, according to our simulations using the CloudAnalyts simulation tool.

Keywords: *Cloud Computing, Load-Balancing, Virtual Machine, Processing Time, Response Time*

Introduction: One kind of computing that describes the utilization of online resources like storage, networks, databases, servers, and software throughout the web is called cloud computing. With web apps, it offers people on-demand services. Concerning data backup and restoration, there is none. The main purposes of cloud computing are resource, information, and internet application sharing [1]. An efficient network of linked servers, virtual machines, data centers, storage devices, etc. makes up a cloud computing system. It is regarded as an advanced IT technology that depends on the embedded distribution of resources across many geographic areas to effectively provide consumers with services when they ask for them. One of the main drawbacks of cloud computing is the imbalanced load assignment caused by requests from various users being randomly assigned to different processors. Therefore, loads must be controlled, and load-balancing is one way to achieve this [2].

By distributing the load over many different systems, load balancing improves the overall performance of handling incoming requests. To fulfill requests, organizations are distributing the load through several computers. The technique known as "load-balancing" prevents a single server from being overloaded. An overloaded server may experience crashing, losing requests, and reducing downtime [3]. Load balancing is a technique for distributing workloads over several network connections on different computers. The primary goal of load balancing is to prevent servers from overloading and potentially collapsing [4]. Load balancing is necessary to obtain a rapid response in a cloud computing environment. It is a technique for increasing performance and speeding up processes by evenly distributing the workload among several computers or data centers. The improvement of load-balancing techniques in cloud computing is still an ongoing research topic. Investigating and putting into practice load-balancing methods that are now in use might enhance the functionality of cloud computing systems. Utilization of resources is a performance metric that measures how well resources are used. The efficient load-balancing method provides maximum usage of resources [5]. As a result, the suggested technique maximizes the use of every virtual computer that the data center in cloud systems creates. Numerous methods are put forward that allow for an equitable distribution of load and the lowest response time. To balance the demand, a load-balancer distributes the load uniformly and equitably across all available computers or servers.

Our research introduces a new VM-assign method that is comparable to the throttle algorithm with a little difference. Throttled did not prefer the virtual machine with the lowest load, and the operation always starts at the beginning. On the other side, the proposed algorithm identifies the least-loaded VMs. If all of the VMs are busy, submit them to the VM Index. It distributes incoming workloads among the available virtual machines. Here, the virtual machine is allocated based on its load, that is, where the least-loaded VM is located, and a new request is then issued. This approach greatly reduces the virtual machine's underutilization; it is then compared to the current round-robin and Throttled procedures. Numerous load-balancing strategies, including the Honeybee-based load-balancing method, Round Robin, Active Monitoring Load-Balancer, Throttled Load-Balancer, etc., have been suggested for cloud environments [6]. The idea of the throttled load-balancing algorithm was modified to create the recommended method (ATA). This proposed technique, compared to Round Robin and Throttled,

Article history:
Received 16 May 2024
Received in revised form 29 September 2024
Accepted 08 October 2024
Available online 01 November 2024

Corresponding author details: Tanjina Jahan
E-mail address:tanjinajahan.cse@gmail.com
Tel: +8801615-566754

maximizes the use of virtual machines. The suggested approach outperforms the other two algorithms in terms of response time and processing time, according to the results of simulations carried out using the simulation model for the three algorithms.

Related works: An overview of the load-balancing techniques used in cloud computing is provided in the remaining section. The primary objective is to assign all incoming jobs to virtual machines that are accessible so that they may be handled right away.

The Central Load-Balancer (CLB) approach was suggested by the authors in [6], which prevents virtual machines from being overloaded or underloaded. According to each virtual machine's importance and status, CLB distributes the load among them. Through modeling, the researchers demonstrated why the load-balancing algorithm based on the CLB approach outperforms both the RR and throttled algorithms. However, the method they've provided doesn't take into account how much memory and CPU are being used right now, which may make the balance of the load more stable and dynamic.

A unique VM-assign technique was presented by the authors Domanal *et al.* [7] for the effective allocation of incoming jobs on virtual machines already present in cloud computing systems. The suggested technique focuses on identifying the virtual machine that is less loaded, after which incoming workloads are widely distributed to them. The authors demonstrated via simulation that the proposed technique overcomes the issue of wasteful resource and virtual machine consumption and performs better than the Active VM-load balancer algorithm suggested in [8].

Numerous methods have been presented according to the two settings, namely static and dynamic. Shridhar's *et al.* [9] suggested a Modified Throttled Algorithm explains how to distribute incoming work evenly across several virtual machines. The current Round Robin and Throttled algorithms are contrasted with the response time of the Modified Throttled Algorithm. Using Modified Throttled Algorithms, the research finds an effective way to manage the load by taking into account the available virtual machines (VMs) and the corresponding requests. Compared to Round Robin and Throttled, Shridhar's suggested Modified Throttled Algorithm provides a faster response time.

Cloud computing and load-balancing are two of the cloud computing resource allocation approaches identified in the current research by S. H. Sabeti *et al.* [10]. The author emphasizes load-balancing and makes an effort to ensure that all servers have about the same amount of work to do. In an attempt to speed up responses and processes, this research suggests using a load-balancing algorithm that combines elements of the ESCE and Throttled algorithms. To reduce the time spent checking for a suitable virtual machine that can handle longer tasks and improve response time, the algorithm first proposes the least busy machine. Two more virtual algorithms, Throttled and ESCE, are combined into a single hybrid algorithm that is proposed. All four algorithms were simulated using the same framework, and the results showed that the suggested method completed tasks more quickly and had a lower total number of iterations than the other three. Additional goals, such as reduced costs and enhanced performance, have not yet been attained due to scheduling and technical constraints.

In this work, S. Y. Mohamed *et al.* [11] present the Balanced Throttled Load-Balancing (BTLB) method. Results from other load-balancing algorithms, including round robin and AMLB, as well as the throttled load-balancing algorithm, are compared with those from BTLB. All four of these algorithms' efficacy will be shown in this analysis. The proposed technique is shown to decrease response times. The results were calculated using a cloud analyst simulation. After comparing simulation results with the four methods, the author may conclude that the balanced-throttled load-balancing approach has the fastest average response time.

V. Dhilip Kumar *et al.* [12] concentrate on workflow balancing using the suggested methodology in this study, and they provide a unique way to balance the load that controls the dynamic scheduling process. A pre-existing load-balancing approach is taken into consideration and adjusted somewhat to suit the current situation. The present throttled load-balancing strategy for workload distribution optimization is the basis for the recommended load-balancing strategy. The given I-throttled load-balancing approach is seen to increase load-balancing consistency, response time, and throughput by 6%, based on the simulation results.

The Cloud Analyst Simulator was used to evaluate the Modified Throttled Load-Balancing Algorithm, the FCFS Algorithm, and the Particle Swarm Optimization Algorithm by P. A. Pattanaik *et al.* [13]. According to the findings, Particle Swam is the optimization method that yields the quickest response time compared to the other two. Moreover, Particle Swam optimization has lower total server costs than the other two techniques. Since costs play a major role in the cloud, minimizing them should be a key concern in terms of both efficiency and customer happiness. Using the Particle Swam Optimization Algorithm, we were able to find a better distribution map that represents the ideal option for our resources. The simulation outcomes are recorded in terms of response time, data center processing time, efficiency, and arrival costs for all three methods.

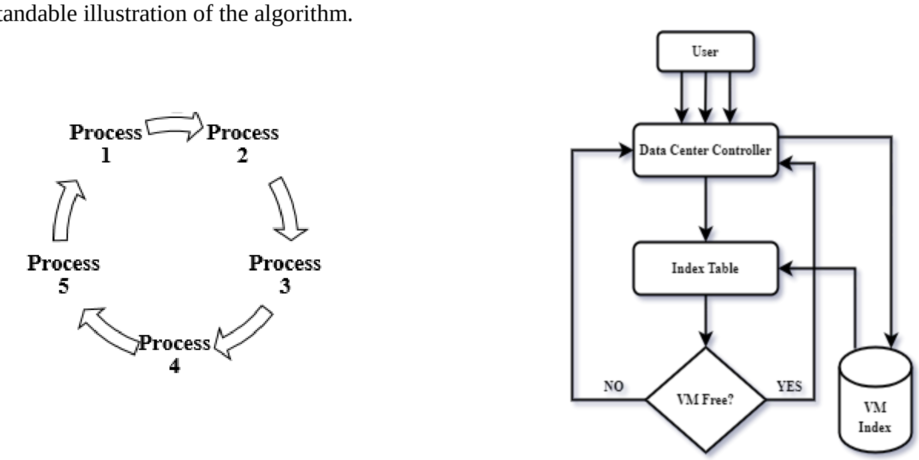
Saurabh Gupta *et al.* [14] introduced the Advanced Throttled Algorithm (ATH) in 2018. They offered a strategy similar to the Throttle technique, with minor changes to the VM index table. The proposed solution, known as the Advanced Throttled (ATH) approach, improves on the Throttled algorithm by uniformly dividing the workload among virtual machines. Using this approach, the load-balancer picks the least amount of load.

The Throttled Modified Algorithm (TMA) was introduced by Nguyen Xuan Phi *et al.* [15] in 2018 to decrease processing and response times in cloud computing. To indicate the availability of virtual machines, this approach uses two indexes: 0 denotes availability and 1 denotes unavailability. Rather than scanning the status of every VM, the two indexes table is used to keep the VM state. This speeds up processing and response times since it just needs to assign the request to the virtual machine that is listed in the available table. As the number of virtual machines rises, the algorithm's output also grows noticeably.

Existing Algorithms:

i. **Round Robin Algorithm:** One of the most basic quantum theory-based algorithms is the round-robin algorithm. The goal of Round Robin is to evenly rotate the load across the virtual machines. Round Robin assigns tasks to virtual machines (VMs) in the data center in a cyclical manner, regardless of their computing capability. This works well in data centers since virtual machines (VMs) are all equipped with the same amount of computing power. Regarding data centers, there are several extensive virtual machines (VMs) that can handle significant differences in power but are inefficient [15]. Fig.1 (a) illustrates the operational procedure of the Round Robin Algorithm. It operates cyclically.

ii. **Throttled Algorithm:** An entirely virtual machine-deployed dynamic algorithm is the throttled load-balancing algorithm. The user uses the throttled load balancer in this allocation to locate the appropriate virtual machine (VM) to complete the task. The VMs are organized into groups based on the number of queries they can handle. As soon as the client submits the request, the load-balancer detects it and looks for the group that can handle it with ease. The problem with this allocation is that it requires the load-balancer to look for an appropriate virtual machine, which would cause an operational delay [16]. Fig. 1(b) demonstrates the step-by-step operation of the Throttled Algorithm. It provides an understandable illustration of the algorithm.



(a) (b)

Fig. 1: (a) Flowchart of Round Robin Algorithm and (b) Flowchart of Throttled Algorithm.

Proposed Algorithm: Advanced Throttled Algorithm (ATA): The Advanced Throttled Algorithm is a Throttled algorithm with a few differences. This approach requires load-balancers to keep index records of busy virtual machines [19]. The load-balancer selects the least loaded VM among the busy ones. The load balancer separates free VMs from the least loaded ones. If not, send it to the Busy Virtual Machine index table. Fig. 2 shows how the proposed algorithm works, and it is the research model of the Algorithm.

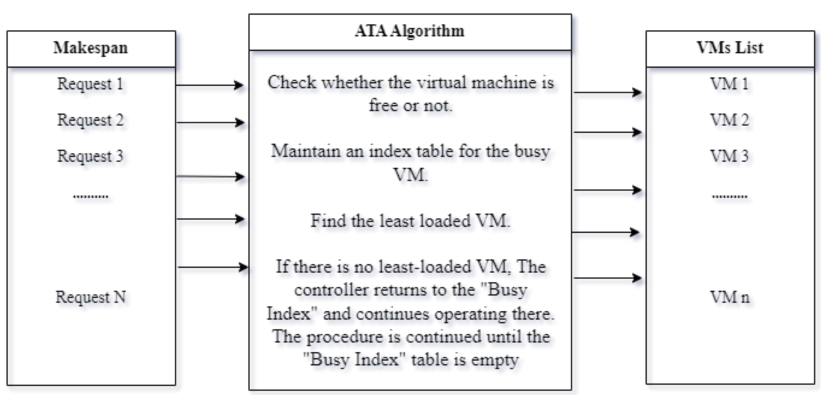


Fig. 2: Research Model of Advanced Throttled Algorithm (ATA).

Algorithm: The Advanced Throttled Algorithm (ATA), which is the suggested algorithm, is an advancement across the Throttled algorithm. This updated edition functions as follows:

Step 1: The load-balancer sent the Data Center Controller a fresh request and inquiry for the new job. It includes all of the virtual machines' state information, such as busy '1' or free '0'. Every virtual machine has a free status of '0' upon startup.

Step 2: The load-balancer will be checked to determine whether a VM is 'free' or 'busy', and the first VM will be 'free'.

If virtual machines are free:

- Sends the VM ID to the VM index, which then forwards the VM ID to the Data Center Controller.

If every virtual machine is busy, then:

- The load-balancer identifies and transmits VM IDs from the data center controller's "Busy Index" database.

Step 3: The least-loaded virtual machine on the busy index table will be examined by the load balancer.

If the virtual machine is least loaded:

- Proceed to step 2.

If the virtual machine is busy,

- The controller returns to the "Busy Index" and continues operating there. The procedure is continued until the "Busy Index" table is empty.

Fig. 3 demonstrates the step-by-step operation of the Advanced Throttled Algorithm (ATA). It provides an understandable illustration of the algorithm.

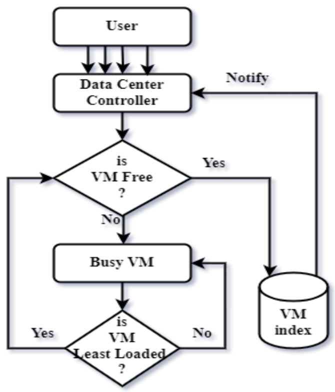


Fig. 3: Flowchart of Proposed Algorithm (ATA).

Simulation Environment: Utilizing the CloudAnalyst simulator is part of the implementation [17]. The majority of research on load-balancing in cloud computing has made use of this simulator, which is a visual design based on CloudSim. A cloud simulator called the cloud analyst provides an effective platform for simulating the real-time development of data centers. Its purpose was to examine how large-scale cloud apps behaved in various deployment scenarios. This simulator makes it simple to model and run simulations frequently. The load-balancing methods are analyzed using Cloud Analyst. Virtual machine load-balancing policies are comfortably covered by the CloudAnalyst simulator. After putting the load-balancing strategy into practice, the simulator's graphical user interface may accept the settings from an interaction and display the outcomes as tables and charts. Fig. 4 is the visual depiction of the Cloud Analyst Simulator. The simulation tool generated the simulation results.

The primary functionalities of a cloud analyst including [18]:

i. A graphical user interface that facilitates experimentation for users.
ii. A cloud analyst makes it simple to run tests with both the same and different settings and to display the results graphically.
iii. Cloud analysts are very configurable and flexible.

Fig. 4: Graphical User Interface of Cloud Analyst Simulator.

Simulation Results and Discussion: By preventing under- or over-loading situations, virtual machines are used efficiently to examine the results. The parameters in the simulation scenario were applied to the algorithm. The goal of this study is to shorten data center processing and response times. As a consequence, among the criteria for evaluating the outcomes are response time and processing time in data centers. Round Robin, throttled load-balancing, and the proposed method are the three most widely used dynamic load-balancing techniques that we are comparing here. In comparison to the original Throttled algorithm and the Round-Robin algorithm, our suggested approach has a shorter response time, according to the results of the study. This research installs an ATA algorithm using the Cloud Analyst tool for testing and simulation. This means that the proposed algorithm shortens the scheduler's computation time and makes load-balancing more efficient.

Table 1. User Database Settings

Name	Region	Requests per User per Hr.	Data Size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off-Peak Users
UB1	0	60	100	13	15	600000	60000
UB2	1	60	100	10	12	150000	15000
UB3	2	60	100	3	5	500000	50000
UB4	3	60	100	7	9	450000	45000
UB5	4	60	100	15	17	300000	30000
UB6	5	60	100	6	8	60000	6000

Simulating the algorithm is the most effective technique to test it. We established six user bases, one for each of the six geographical regions, and five data centers. Table I displays many parameters for each user base. The parameters of the primary implement section, which include the user bases, simulation length of time, and application placement setup, are shown in Table 1. The investigation with cloud simulation using the settings shown already, and then execute the Cloud Analyst load-balancing algorithm: Comparing the results, particularly the total response time and datacenter processing time, is recommended when implementing the suggested algorithms, Round Robin and Throttled, with the same input. This table 1 presents the User Database Settings variable in the Cloud Analyst Simulator. We may pick a different user base and region.

We have constructed two distinct cloud simulation settings, each with three scenarios. After that, we set up a random request environment with user bases from various regions, but the service is still hosted on the same cloud. The proposed ATA method is then put to use in a simulated environment. The suggested method is also performed and compared to two other algorithms, namely Throttled and round-robin.

Case 1: Run a simulation using 30 simulated virtual machines.

Table 2 includes information on the number of virtual machines, memory capacity, and bandwidth. Table 3 provides a visual representation of all of the specifications specified for each data center. Using the previously mentioned settings, the simulation was run for each load-balancing method, and the results were analyzed based on overall response time. Response Time: To measure the efficiency of a virtual computer, we utilize its response time. The algorithm is more efficient the lowers the cloud's predicted response time. The outcomes of Round Robin, Throttled, and the Proposed Algorithm are shown in Tables 4, 5, and 6, respectively. After conducting the simulations, we evaluated the total response time and data center processing time for each algorithm. The results for the Round Robin Algorithm are shown in Table 4, for the Throttled Algorithm in Table 5, and for the proposed Algorithm in Table 6.

Table 7 displays the average response and processing times of Round Robin, Throttled, and Proposed Algorithms together. The average response and processing times for the three methods are graphically represented in Fig. 5 (in milliseconds). The suggested method has decreased average response time and processing time across users, as the chart illustrates.

Table 2. User and VM configuration setting

Data Center	Number of VM	Memory	Bandwidth
DC1	6	512	1000
DC2	6	512	1000
DC3	6	512	1000
DC4	6	512	1000
DC5	6	512	1000

Table 3. Displays the Parameters for data center configuration

Name	Region	Arch	OS	VMM	Cost per VM $/Hr.	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical HW Units
DC1	0	×86	Linux	Xen	0.1	0.05	0.1	0.1	5
DC2	1	×86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC3	2	×86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC4	3	×86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC5	4	×86	Linux	Xen	0.1	0.05	0.1	0.1	1

Table 4. Round Robin Algorithm

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	239.30	36.43	643.06
Data Center processing time:	158.73	0.09	321.24

Table 5. Throttled Algorithm

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	142.53	36.43	502.17
Data Center processing time:	82.32	0.09	152.09

Table 6. Advanced Throttled Algorithm (ATA)

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	100.73	36.43	498.20
Data Center processing time:	41.45	0.09	145.50

Table 7. Shows the average response time and data center processing time, expressed in milliseconds (ms), for three different methods

Algorithms	Round Robin	Throttled	**Proposed**
Average Response Time (ms)	239.30	142.53	**100.73**
Average Processing Time (ms)	158.73	82.32	**41.45**

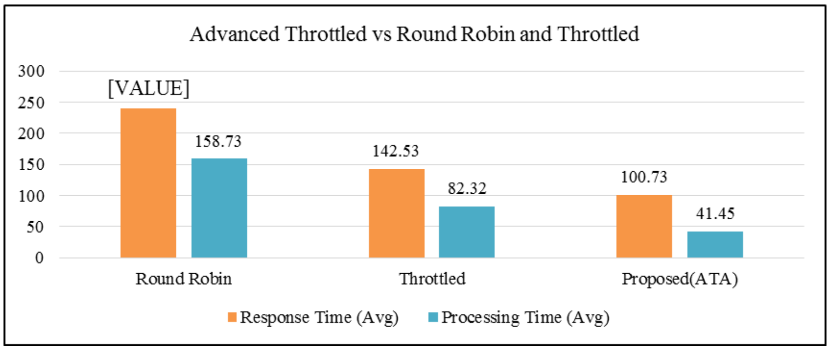


Fig. 5: Average response time and processing time in milliseconds using 30 VM for the three algorithms.

Case 2: Run a simulation using 60 simulated virtual machines

Table 8 includes information on the number of virtual machines, memory capacity, and bandwidth and table 9 provides a visual representation of all of the specifications specified for each data center. Using the previously mentioned settings, the simulation was run for each load-balancing method, and the results were analyzed based on overall response time. Response Time: To measure the efficiency of a virtual computer, we utilize its response time.

Table 8. User and VM configuration setting

Data Center	Number of VM	Memory	Bandwidth
DC1	12	512	1000
DC2	12	512	1000
DC3	12	512	1000
DC4	12	512	1000
DC5	12	512	1000

Table 9. Shows the parameters of the Data Center Configuration

Name	Region	Arch	OS	VMM	Cost per VM $/Hr	Memory Cost $/s	Storage Cost $/s	Data Transfer Cost $/Gb	Physical HW Units
DC1	0	×86	Linux	Xen	0.1	0.05	0.1	0.1	5
DC2	1	×86	Linux	Xen	0.1	0.05	0.1	0.1	2
DC3	2	×86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC4	3	×86	Linux	Xen	0.1	0.05	0.1	0.1	1
DC5	4	×86	Linux	Xen	0.1	0.05	0.1	0.1	1

Table 10. Round Robin Algorithm

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	203.14	36.90	648.59
Data Center processing time:	121.59	0.09	324.35

Table 11. Throttled Algorithm

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	126.76	36.93	474.15
Data Center processing time:	68.47	0.09	152.60

Table 12. Advanced Throttled Algorithm (ATA)

	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	92.11	36.94	474.15
Data Center processing time:	34.58	0.09	146.01

Table 13. Shows the average response time and data center processing time, expressed in milliseconds (ms), for three different methods

Algorithms	Round Robin	Throttled	**Proposed**
Average Response Time (ms)	203.14	126.76	**92.11**
Average Processing Time (ms)	121.59	68.47	**34.58**

The algorithm is more efficient the lowers the cloud's predicted response time. The outcomes of Round Robin, Throttled, and the Proposed Algorithm are shown in Figures 10, 11, and 12, respectively. After simulation, we determined the Round Robin Algorithm's total response time and data center processing time. Table 10 displays the results. After simulation, we determined the Throttled Algorithm's total response time and data center processing time. Table 11 displays the results. After simulation, we determined the proposed Algorithm's total response time and data center processing time. Table 12 displays the results. Table 13 displays the average response times and processing times of Round Robin, Throttled, and Proposed Algorithms together.

The average response and processing times for the three methods are shown in milliseconds in Fig. 6. The suggested method has decreased average response time and processing time across users, as the chart illustrates. In terms of data center processing and system flexibility, the results of the studies conducted in the first two situations indicate that the proposed (ATA) algorithm is quicker than the other approaches. The proposed (ATA) algorithm improves load-balancing over the throttled and round-robin approaches.

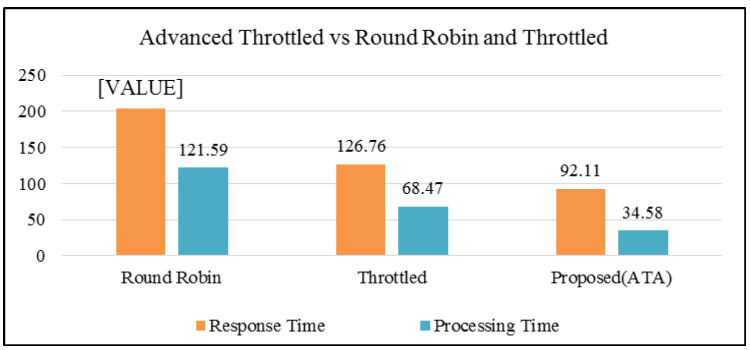


Fig. 6: Average response time and processing time in milliseconds using 60 VM for the three algorithms.

Conclusion: The following paper provides a summary of cloud computing systems, standards, and the latest load-balancing approaches. In addition, it investigates the technique of enhancing cloud computing efficiency by developing the load balancer. By simulating and utilizing the Cloud Analyst simulator, we develop and evaluate load-balancing approaches like Round Robin, Throttled, and ATA (Advanced Throttled Algorithm). While applying these three techniques for load-balancing, we believe that designing a new load-balancing algorithm is essential to work in cloud computing, where load-balancing can be a challenging task. The suggested approach is meant to enhance response time and processing time. To achieve this goal, the algorithm provides a virtual machine with less utilization, which may reduce response time. When the aforementioned simulation results for the three algorithms are compared, we can infer that the Advanced Throttled Load Balancing Algorithm (ATA) has a superior overall reaction time to the throttled and round-robin methods. In addition, this research contributes to load-balancing more successfully than well-known algorithms like Throttled and Round Robin. Load-balancing hasn't been able to accomplish additional objectives, including cutting expenses and raising production, because of time and technological constraints.

References:

[1] Amrutanshu Panigrahi, Sahu, B., Saroj Kumar Rout, & Amiya Kumar Rath. M-Throttled: Dynamic Load Balancing Algorithm for Cloud Computing. (2020). 3–10.
[2] Garg, S., Dwivedi, R. K., & Chauhan, H. Efficient utilization of virtual machines in cloud computing using Synchronized Throttled Load Balancing. 2015 1st International Conference on Next Generation Computing Technologies (NGCT). (2015).
[3] Elnagar, N. G., Elkabbany, G. F., Al-Awamry, A. A., & Abdelhalim, M. B. Simulation and performance assessment of a modified throttled load balancing algorithm in a cloud computing environment. International Journal of Power Electronics and Drive Systems, 12(2022), 2087–2087.
[4] Mirobi, G. J., & Arockiam, L. Dynamic Load Balancing Approach for Minimizing the Response Time Using An Enhanced Throttled Load Balancer in Cloud Computing. IEEE Xplore. (2019, November 1).
[5] Mekonnen, D., Megersa, A., Sharma, R. K., & Sharma, D. P. Designing a Component-Based Throttled Load Balancing Algorithm for Cloud Data Centers. Mathematical Problems in Engineering, (2022). 1–12.
[6] Soni, G., & Kalra, M. A novel approach for load balancing in cloud data center. 2014 IEEE International Advance Computing Conference (IACC). (2014).
[7] Domanal, S. G., & Reddy, G. R. M. Optimal load balancing in cloud computing by efficient utilization of virtual machines. 2014 Sixth International Conference on Communication Systems and Networks (COMSNETS). (2014).
[8] Kumer Paul, S., Ala Walid, Md. A., Rani Paul, R., Uddin, Md. J., Rana, Md. S., Kumar Devnath, M., Rahman Dipu, I., & Haque, Md. M. (2024). An Adam based CNN and LSTM approach for sign language recognition in real time for deaf people. In Bulletin of Electrical Engineering and Informatics (Vol. 13, Issue 1, pp. 499–509). Institute of Advanced Engineering and Science. https://doi.org/10.11591/eei.v13i1.6059
[9] Domanal, S. G., & Reddy, G. R. M. Load Balancing in Cloud Computing using Modified Throttled Algorithm. 2013 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). (2013).
[10] Sabeti, S. H., & Mollabgher, M. Proposing a load balancing algorithm with an integrative approach to reduce response time and service process time in data centers. Brazilian Journal of Operations & Production Management, 16(4), (2019). 627–637.
[11] Mohamed, S. Y., Taha, M. H. N., Elmahdy, H. N., & Harb, H. A Proposed Load Balancing Algorithm Over Cloud Computing (Balanced Throttled). International Journal of Recent Technology and Engineering (IJRTE), 10(2), (2021). 28–33.
[12] V. Dhilip Kumar, J. Praveenchandar, Arif, M., Brezulianu, A., Oana Geman, & Ikram, A. Efficient Cloud Resource Scheduling with an Optimized Throttled Load Balancing Approach. Computers, Materials & Continua, 77(2), (2023). 2179–2188.
[13] Pattanaik, P. A., Roy, S., & Pattnaik, P. K. Performance study of some dynamic load balancing algorithms in a cloud computing environment. 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN). (2015).
[14] Gupta, S., Dixit, N., & Yadav, P. An Advanced Throttled (ATH) Algorithm and Its Performance Analysis with Different Variants of Cloud Computing Load Balancing Algorithm. Communications in Computer and Information Science, 385–399. (2018).
[15] Xuan Phi, N., Tin, C. T., Ky Thu, L. N., & Hung, T. C. Proposed Load Balancing Algorithm to Reduce Response Time and Processing Time on Cloud Computing. International Journal of Computer Networks & Communications, 10(3), (2018). 87–98.
[16] Alkhatib, A. A., Alsabbagh, A., Maraqa, R., & Alzubi, S. Load Balancing Techniques in Cloud Computing: Extensive Review. Advances in Science, Technology and Engineering Systems Journal, 6(2), (2021). 860–870.
[17] N. Le, H., & C. Tran, H. ITA: The Improved Throttled Algorithm of Load Balancing on Cloud Computing. International Journal of Computer Networks & Communications, 14(1), (2022). 25–39.
[18] Hamid Shoja, Hossein Nahid, & Azizi, R. A comparative survey on load balancing algorithms in cloud computing. (2014).