Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3453–3467, DOI 10.12694/scpe.v25i5.3165

ENHANCED THROTTLED LOAD BALANCING FOR VIRTUAL MACHINE ALLOCATION IN MULTIPLE DATA CENTERS

P.HANUMANTHA RAO *AND DR.P.S.RAJAKUMAR †

Abstract. ”Cloud computing” hosts software and other services in remote data centers that customers can access worldwide. The user may access all the services and applications online. The IT industry has benefited greatly from the proliferation of cloud computing. On the flip side, organizations moved their operations to the cloud as a result of industrial automation. A surge in demand for cloud computing was directly correlated to the quick migration of businesses. Businesses looking to minimize expenses without sacrificing service quality will find this approach to be ideal. Considering the meteoric rise of cloud computing, service providers are delighted. Contrarily, distributing resources is a challenging task. Cloud computing overcomes some of its most fundamental obstacles, one of which is the load-balancing approach employed by load-balancers to economically optimize costs while minimizing time expenditures. Quick services for cloud customers and minimal cost for cloud providers are the goals of the optimal resource allocation method. This research suggests a novel approach to increase task processing time, which can aid in increasing cloud computing's load balancing capabilities. The proposed method Enhanced Throttled Load Balancing Algorithm (ETLBA) is an upgrade to the original Throttled Algorithm, which efficiently performs resource allocation and load balancing. The proposed ETLBA is contrasted with the existing algorithms, Round Robin, Active Monitoring Load Balancing Algorithm (AMLBA) and Throttled Load Balancing Algorithm (TLBA) to display the efficacy. Cloud Analyst tool simulates the proposed and existing methods. According on the results of the simulations, the proposed algorithm ETLBA achieves better outcomes than the popular existing algorithms in terms of processing time, request processing time, and datacenter cost. It shows 18% reduction in response time, 7% reduction in data center processing time, 16% reduction in data center request processing time and 4% less data center cost compared to the existing solutions. ETLBA performs better by selecting virtual machines using a prioritized index table and consumption index. It limits idling resources, improves response as well as reduces processing times, and cloud costs compared to conventional solutions.

Key words: Cloud Computing, Resource Allocation, Round Robin, Load Balancing, Throttled Load Balancing, Enhanced Throttled Load Balancing, Improved Response time, Data center Cost.

1. Introduction. “Cloud computing” is a system that lets people share and access large amounts of data and other resources. After using a service, users only pay for what they spent. The open environment of cloud computing allows to demonstrate how data, software packages, and distributed resources are stored.

Many multinational corporations (MNCs) offer cloud services, including Microsoft, Amazon Web Services, and many more [1]. ”Software as a service,” ”infrastructure as a service,” and ”platform as a service” are the three most prevalent models of cloud computing [2]. In this context, “Infrastructure as a service” is exemplified by AWS EC2 instances. Google Apps is an example of software as a service, while Microsoft Azure is an example of a platform as a service.

Cloud computing primarily assists in the sharing of resources, data, and internet-based applications. Using web-based apps, it offers consumers on-demand services. Concerns regarding data backup and restoration are unfounded. Fig. 1.1 shows the elaborated infrastructure of the cloud computing with all the components included.

The benefits of cloud computing is its usability and cost-effectiveness of its administration. It features many desirable traits, including, but not limited to, reliability, virtualization, multitasking, improved framework cost, and referenced highlight assistance. The innovation that is stepping forward is cloud computing. A large number of startups nowadays use cloud computing. Instead of purchasing the foundation, business innovators are saving time, money, and office space by connecting the cloud benefits using PCs. In cloud computing, users

*Research Scholar, Dept. of CSE, Dr.M.G.R.Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu India. (hanumanthraovbit@gmail.com).
† Professor, Dept. of CSE, Dr.M.G.R.Educational and Research Institute, Maduravoyal, Chennai, Tamil Nadu India. (Rajakumar.subramanian@drmgrdu.ac.in).

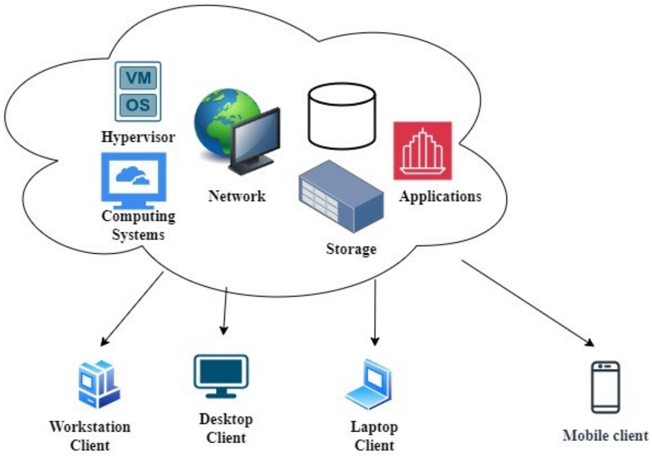

Fig. 1.1: Cloud Computing Infrastructure

can access services on an as-needed basis and pay for them in advance. This is why cloud computing services are becoming increasingly popular among customers with occasional needs.

Cloud computing aims to improve efficiency, decrease computational power requirements for thin devices, decrease operational expenses, increase security, and speed up data processing. In addition to these benefits, it makes the system more resilient to changes, faster at processing various data sets, less expensive overall (hardware, software, and maintenance costs), more energy efficient and less space hungry on discs [21].

In order to keep up with demand, organizations are distributing the burden among multiple servers. A technique known as ”load balancing” is employed to ensure that no single server is overwhelmed. Delays, request drops, or even crashes may occur if the system is overloaded. In order to distribute the processing load among numerous servers, Load Balancing makes use of network connections. It reduces total response time and maximizes throughput [3]. To avoid any server going down due to overload, load balancing is essential.

When it comes to cloud environments, load balancers are efficient. Servers can be rendered inoperable due to crashes caused by extremely heavy workloads. All procedures rely on consistently fast response times and high availability of services. In addition to detecting downed servers, load balancers can reroute requests to up and operating servers. If one server is too busy, a load balancer can send requests to another. Load balancers primarily aim to ensure that servers are in better health. Servers and users are regulated by a load balancer.

It processes data packets sent by networks and applications. In order to perform multi-server request distribution, a load balancer employs a number of techniques. There are two tiers to load balancing: Level one, which entails establishing a link between the applications or services and the virtual machines that are being requested. Layer 2 entails establishing a link between physical hosts and virtual machines [4],

Load balancing is a vital component of cloud computing, for efficient resource allocation across multiple data centers. Many existing and popular load balancing techniques are used for resource allocation. But response time is the crucial factor that evaluates the efficiency of these techniques. The primary objective of this article is to present a novel approach for improvising the response time and optimizing the data center costs when compared with existing algorithms. The proposed method Enhanced Throttled Load-Balancing Algorithm (ETLBA) aims at efficient load balancing by improvising the existing throttled load balancing technique. This study presents a contrary of outcomes obtained through the use of “Cloud-Analyst Simulator”. The results demonstrate that the suggested algorithm decreases the total processing time spent for the requests and response time of the datacenter.

This is the remaining structure of the paper: Section 2 presents work in this area along with the description of the several load-balancing algorithms. The proposed algorithm, along with its flowchart and pseudo-code, is presented in Section 3. The paper’s section 4 details the experimental setting. While the results of the

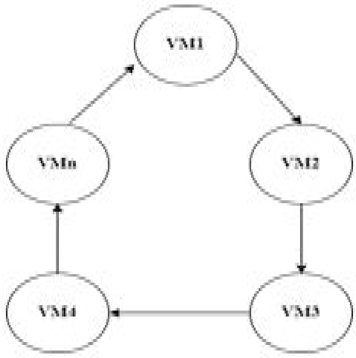

Fig. 2.1: Round Robin Algorithm Process

comparison and analysis are presented in Section 5. The paper is concluded in Section 6.

2. Literature Review.

2.1. Load-Balancing Algorithms.

2.1. Load-Balancing Algorithms. There are a plethora of loadbalancing techniques available in the cloud. Active Monitoring LoadBalancing (AMLB), Round Robin (RR), and Throttled (TLB) are the three most used load-balancing algorithms [5]. The algorithms that were employed are evaluated using "Cloud Analyst Simulator".

2.1.1. Round-Robin Algorithm. When it comes to time-sharing systems, the useful and most popular load-balancing method is Round-Robin. With Round-Robin, virtual machines (VMs) are distributed fairly in a circular order. Assuming the processing powers of the virtual machines are equal, this method provides benefits such as being easy to understand and implement, guaranteeing fairness in operations, and handling tasks promptly.

Distributing client requests among servers is made easy with this method. Requests from clients are received sequentially by each server. Since it is simple to both understand and apply, it ranks high among the most popular algorithms [6].

This method iteratively routes client requests to the servers that are available. When server processing and storage capacities are close to one another, it works well. This process sends requests to the node with the fewest connections, which means that at any one moment, a few of nodes can be experiencing excessive load, while others are completely unoccupied. Fig. 2.1 depicts the process of Round Robin Algorithm.

Within this algorithm, there is no famine. Round Robin's drawbacks include a lack of scalability and flexibility, the fact that some nodes may be under tremendous strain while others sit idle, and the fact that the former allocation status of the virtual machine is not preserved.

2.1.2. Weighted Round-Robin Algorithm. The Weighted Round-Robin algorithm takes its cue from the Round-Robin method and employs a weight table to allocate tasks to virtual machines according to their capacities. It executes circular distribution using this table. The algorithm's benefits: enhances the Round-Robin algorithm by incorporating a weight-table of virtual machines' processing capacities into its rotational operation; this makes the method more efficient than the original in situations when the processing powers of the virtual machines vary. The lack of scalability and flexibility, as well as the inability to restore the prior allocation status of virtual machines, are further drawbacks. Fig. 2.2 depicts the process of the algorithm.

2.1.3. Active Monitoring Load Balancing Algorithm. Because of the unpredictable nature of cloud computing, certain servers may experience excessive load during equally distributed current execution, while others may be inactive or barely touched. Similarly, by shifting resources from overburdened to underutilized servers, load distributing boosts performance. One important characteristic of cloud computing is efficiently distributing resources in the cloud and scheduling, which is used to evaluate the system's performance [7].

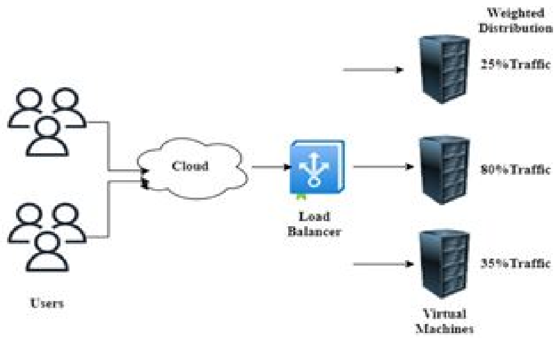

Fig. 2.2: Weighted Round Robin Algorithm Process

Optimization of costs, as a result of faster response and processing times, is affected by the attributes taken into account.

Clients submit their jobs to the computer system here. Each job that is submitted to the cloud is added to a stack and queued as it arrives. After making an estimate of the job's size, the cloud manager verifies the virtual machine's availability and capacity. The job scheduler assigns the specified resource to the queued work without delay as soon as the job size and the size of the accessible asset (virtual machine) are in agreement [8].

The major drawback of this algorithm is, in order to assign resources to a free virtual machine, it sequentially evaluates each virtual machine which is time consuming process.

2.1.4. Throttled Load Balancing Algorithm. A static load-balancing method best describes this technique. Here, we begin by verifying the values of each virtual machine's index. For system resource allocation, the a request has been forwarded to the point where the load balancer parses a table. To update the allocation policy, a particular load balancer is notified of the request, which then either returns it to the requester or processes it in reverse [9]. The complete procedure of deallocating the system begins once the allocation of the system is successful.

The increased performance and utilization are the results of this mechanism's provision of more sharing and allocation of system resources. The throttling threshold is typically set at 1. The threshold might be set to a user-specified value with little configuration.

At all times, this algorithm's fixed quantity of cloudlets are allotted to a single virtual machine. The quantity of virtual computers (VMs) available at the datacenter determines how incoming requests are dealt with if there are more request groups. Aside from that, it waits for the next available virtual machine to become available.

In comparison to the Active Monitoring Load Balancing (Optimal) technique, this algorithm represents an incremental improvement. Initially, this algorithm begins its search from the most recently allocated virtual machine all the way up to the nth virtual machine. Nevertheless, there is still a problem, and that is the fact that it does not make use of those virtual machines that become available subsequently to the execution of the request. The below Fig. 2.3 represents the procedure of Throttled Load-balancing algorithm.

2.2. Related Work. Using a throttled load balancing method in a multiple data centers, the study [10] optimized response time by distributing workloads across virtual machines. After much deliberation, they settled on the throttled load balancing algorithm as the best option for the data center in terms of processing time for requests and total response time, with minimal processing expenses.

The authors of [11] presented their idea for the Advanced Throttled Load Balancing Algorithm in the year. A priority is given to every virtual machine. Depending of the capacity of the virtual machine (VM) as well as the number and size of tasks that have been given to the operation, the priority is determined. Among the several virtual machine (VM) sets that are accessible, the enhanced tuning scheduler chooses the VM that has the highest priority. In addition, a level of priority is established in order to prevent overloading. On the other

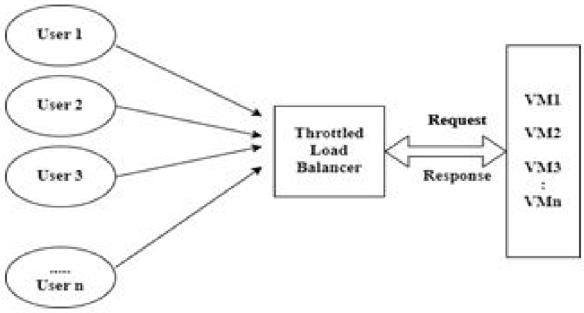

Fig. 2.3: Throttled Load-Balancing Algorithm Process

hand, if the virtual machine's priority is lower than the priority level, then the job will not be assigned to that virtual machine.

The paper [12] suggested an effective procedure for load balancing that is dependent on two distinct characteristics. The time it takes to respond to queries is the first distinguishing feature, and the second characteristic is the load distribution among the virtual machines that are already in existence. They contrasted the throttled variant with the Round Robin approach and suggested a tweaked version of the throttled algorithm.

Furthermore, it was discovered that the utilization of virtual machines (VMs) is more effective when using the Round Robin algorithm and the changed Throttled algorithm in comparison to the Throttled algorithm. In addition, out of the three algorithms tested, the Modified Throttled method yielded the best average reaction time.

The review study conducted in [13] demonstrates that load balancing is an essential technique of the cloud computation environment. It contributes to the improvement of load distribution and the efficient allocation of resources, particularly with regard to the enhancement of response time for cloud users. According to the article, there are a great deal of problems associated with LB. Some of these problems include the scheduling of activities, migration, and the utilization of resources, among other things. Research and studies on load balancing that have been conducted over the previous six years are surveyed and analyzed by the writers. This analysis's findings also show that intelligent methods, including AI and ML, have potential for cloud-based learning analytics. In particular, this study benefits researchers in identifying research topics connected to load balancing, particularly with regard to reducing the response time and avoiding breakdowns in the servers. Another benefit of this research that supports our idea is the availability of tools required for simulation and experimental contexts. This research focuses on the modelling resources.

The exceptional quality of Cloud Analyst and Cloud Sim as top-tier resources for this area of research is another thing that they demonstrate. This is the advantage of employing these tools.

A modified throttled balancing technique was proposed in [14], which they then implemented with the help of the CloudAnalyst tool of CloudSim. Following this, they checked it against other methods of load balancing and confirmed that a tweaked throttled algorithm outperformed the more traditional approaches.

The paper [15] suggested using a neural inference fuzzy system for load effective balancing. In this study, the authors also discuss the saftey of virtual machines (VMs) within cloud-hosted environments. There is a correlation between load balancing and the NP-hard optimization issue. According to Forbes, they are following the news regarding the implementation of the Protection of Personal Information Act. Security of the cloud continues to be a problem, and the current system makes use of a hybrid-based fuzzy LB, but this does not satisfy those requirements. In order to improve CPU utilization and turnaround time, the authors proposed their work, which they referred to as MANFIS (Modified Adaptive Neuro Fuzzy Inference System).

In addition, they focused on increasing the level of security that their work offered. Utilizing the Fire-fly Algorithm allows for the optimization of the parameters of the MANFIS system. Utilizing the Enhanced Elliptic Curve Cryptography allows for the implementation of security measures for user authentication. Users can be authenticated using this method, which does not require a password. Based on the findings, the authors inform

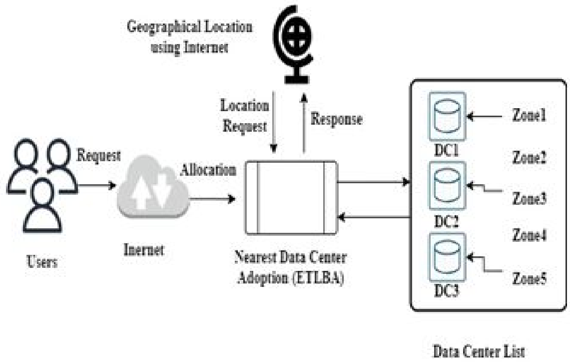

Fig. 3.1: Nearest Data Center Adoption (ETLBA)

us that cloud security has been enhanced and meets our expectations. Further, the experimental results reveal that they outperform the current system in terms of resource utilisation, cost, and execution time.

In their study, the authors of [4] introduced three algorithms for load balancing. These algorithms find use in cloud computation. Here are three algorithms: RR, AMLB, and TLB, which stand for "round-robin, active load-balancing monitoring, and throttled load-balancing", respectively. Following an analysis of the three load balancing algorithms' performance, it was determined that TLB outperformed the others in terms of data centre time and reaction time.

3. Proposed Method - Enhanced Thottled Load Balancing Algorithm (ETLBA). We propose an Enhanced Throttled Load Balancing Algorithm (ETLBA) as a solution to the problem that arises while utilizing the throttled load balancing algorithm. The proposed method consists of three major components:

1. Nearest data center adoption
2. Load Balancer with availability index
3. Usage index of Virtual Machines.

3.1. Nearest Data Center Adoption. The rapid development of cloud platforms has resulted in service providers maintaining many data centres spread out over the globe. The data centers that make up the cloud computing environment are dispersed throughout multiple regions.

The proposed work implements the nearest data center (DC) adoption such that to improvise the response time. The DC-1 server is hosted in zone 1, the DC-2 server is hosted in zone 3, and the DC-3 server is hosted in zone 5. The allocation of virtual machines in accordance with the proposed policy for the data center is done based on which are geographically nearest [16].

The procedures that make up the suggested algorithm for selecting the data center that is geographically nearest to the user, is listed below

1. The request is presented by the first user.
2. The database of Internet characteristics will be responsible for maintaining the table of region proximity
3. The request was forwarded to the data center that was found to be the closest.
4. In the event when multiple DC is experiencing the same network delay.
   In order to maintain a balanced load,
   a) Assign the DC in a random fashion.
   b) Give the DC with the lowest possible network latency
5. Make sure the data center policy is keeping up to date.

The process is depicted in the Fig. 3.1.

3.2. Load Balancer with availability index. The databases of virtual machines provide the basis of a throttled algorithm. Load balancers keep track of virtual machine identifiers and the status of their services, such as whether they are available or busy [17].

The following steps depict the role and process of load balancer with availability index in the proposed ETLBA.

1. Load Balancer creates and updates an table of index for each virtual machine. Additionally, it monitors the availability and busyness of each virtual machine. When the programme began running, every single virtual computer was online.
2. User initiates the request and that will be allocated to nearest Data Centre as mentioned in part 1. Now Data center Controller is then assigned a new user request.
3. The Data Centre Controller will call the Load Balancer in order to allocate the virtual machine to the next available slot.
4. After that, when all available virtual machines are ready, the Efficient throttled load balancer will begin to build a new map.
5. Then, if the length of the available VM map is larger than zero, the efficient throttled load balancer will deconstruct the map and retrieve the first VM ID from it. After that, the process will be taken up by part 3 of the proposed method ETLBA

3.3. Usage Index of Virtual Machine. In this part, the proposed ETLBA maintains the usage list of all available VMs. The most recent task's total processing time is used to compute usage. The overall cost of a datacenter is determined by adding up the costs of data transfer and virtual machine rental.

1. The efficient load balancer monitors the usage of the VM who's ID was retrieved. If the selected one is with the lowest usage, it retrieves its ID from the "Available Index" table and returns it to the datacenter controller (DCC).
2. Otherwise the available virtual machine with lowest usage is searched again in the index table and retrieved, further for forwarding to the data center controller.
3. The Data Centre Controller receives the VM id from the Enhanced Throttled Load Balancer.
4. Based on that identifier, the Data Centre Controller will pass the call on to the appropriate virtual machine.
5. After the new VM_ ID has been allocated, the data center controller informs the enhanced throttle load balancer and removes the corresponding entry from the available virtual machine map. And keeps the status as 'Busy'.
6. After the Enhanced Throttled Load Balancer receives the request from the Controller of Data Centre, it upgrades the VM map of the available virtual machines accordingly.
7. The following procedures are executed in the event that the virtual machine (VM) in Fig. 3.2requested does not appear in the VM Map:
   a) The Enhanced Throttled Load Balancer returns 1.
   b) The database administrator will then add the request to a queue.
   c) The Enhanced Throttled Load Balancer receives a alert from the Data Centre Controller when the processing requests of each virtual machine are finished and the response has been received. It then de-allots the respective virtual machine and adds its ID to the available VM Map.
   d) In the instant after a VM is de-allocated, the Data Centre Controller examines the request queue. If there are any calls in the pending queue, processing will start at the third phase and continue thereafter.

3.4. Procedure - ETLBA.

Procedure Enhanced Throttled Load Balancing (ETLBA)_Part 1.

Step 1: Users sends requests from different regions for resource allocation.
Step 2: Database of Internet characteristics will maintain the table of region proximity
Step 3: While(New request are received by the Nearest Data Center Adoption Method)
Step 4: Check (Nearest Data Centre =available)
Step 5: Allocate the user request to the nearest Data Centre with low latency
Step 6: In case of multiple nearest data centre with same latency
Step 7: Allot the nearest data center on random basis
end procedure

Fig. 3.2: Procedure of ETLBA

Procedure Algorithm Enhanced Throttled Load Balancing (ETLBA)_ Part 2.

Step 1: Make sure that all virtual machines are first assigned the state of "AVAILABLE" in the VM State list; otherwise, set it to "BUSY."
Step 2: While(New requests are received by the data center Controller)
Step 3: If (available VMArray () >0)
Step 4: If 'Yes' :
Step 5: Check (Usage_VM.Selected =least)
Step 6: Return the VMID to the data centre
Step 7: else : Search for the next least usage available VM
Step 8: If (available VMArray () =0) Data Centre Controller queues the request
Step 9: Repeat Step 3

end procedure

Procedure Enhanced Throttled Load Balancing (ETLBA) _ Part 3.

procedure Enhanced Load Balancer_Usage Index

Begin

Step 1: VM_Id = 1; Min = 0; i=0;
Step 2: For each VM_selected in VMList
Step 3: usage = vm.getUsage();
Step 4: vmId = vm.getId();
Step 5: If i== 0 then
Step 6: min = usage;
Step 7: Else

Step 8: If min > usage then
Step 9: min = usage;
Step 10: End If
Step 11: End If
Step 12: i++;
Step 13: End For
Step 14: Return VM_Id;
Step 15: End

The proposed method "ETLBA" allocates resources efficiently as mentioned in the above procedure. In detail the above procedure works as: User sends request from different regions for resource allocation. All these requests are maintained region wise by the internet database. If the nearest data center is free, then it is allocated to the user request based on priority. All the user requests are queued by the data center controller. If more than one nearest data center is free, then the nearest one is allotted on random basis. All the data centers are tagged with index of availability and busy status. The data center with index available are chosen for allocation in response to user request. If two or more datacenters nearer to the user region are available with availability index then their usage is checked to choose for allocation. Once data center is allocated to the user, then its status is marked as busy and if freed, as available by data center controller.

4. Experimental Setup. Data center load balancing techniques can be cost-effectively tested using simulation. A simulated data center environment can be modelled using a variety of tool sets. Cloud Analyst simulation makes it easier and faster to examine the data. Cloud Analyst Simulator makes use of a number of technical phrases, so let's review them.

4.1. Simulator - Cloud Analyst . The cloud computing environment 'simulator' that is most often used is Cloud Analyst [18]. It is an event driven simulator [19].

The major considerations are:

1. There are six distinct regions on the Earth map in the Cloud Simulator. Data centers and user bases can be located in any of the six zones.
2. The internet as it exists in the real world serves as the model in Cloud Analyst Simulation. The transmission delay and latency are caused by the traffic on the internet.
3. In the Cloud Simulator, a collection of users is referred to as a User Base. Hundreds of users could make up a single user base at times. One tool for producing load is the user base.
4. An Digital cloud-let is a user's collection of requests. Execution commands, input files, and output files are all carried over the Internet cloud
5. Cloud Analyst Simulator revolves around its controller. It will oversee operations in the data-center setting, like the creation and deletion of virtual machines.
6. The Controller determines which virtual machine (VM) should execute the cloud task by using a virtual machine load balancer (VmLoadBalancer).

4.2. Simulation Settings. Zone1, Zone2, Zone3, Zone4, Zone5, and Zone 6 were labelled on the global map [20]. The social media platform X, which boasts over 300 million active users globally, is worth considering. The same is depicted in Fig. 4.1 [20] and Table 4.1.

For cloud usage and load balancing simulation, 5% of 'X' users are used. Additional assumptions include 10% of users being online at the busiest times and 10% being inactive during peak hours. We will define six in total, user databases for the six available zones, using the specifications in Table 4.2.

Fig. 4.2 displays the sample settings of the primary configuration section, which includes the following: user bases, application deployment configuration, and simulation duration.

The user requests are generated by each userbase and delivered to the processing facility for data storage. The various configurations of data centers are detailed in Table 4.3, which includes information such as the operating system, hardware type, and the cost of individual operations.

5. Results and Performance Analysis. After running simulations with the aforementioned parameters, we compared the results based on Data processing time, Overall response time, Data Center request servicing time, and Data Centre cost for the following load balancing algorithms: Round Robin, AMLBA - Active

Fig. 4.1: DC1, DC2, DC3 cloud analyst simulation

Table 4.1: X Users

Zonal Name	Zonal-Id	Registered -U sers	Sample-data (5%)
North America	1	95 Million	35,00,000
South America	2	25 Million	15,00,000
Europe	3	75 Million	25,00,000
Asia	4	35 Million	18,50,000
Africa	5	6 Million	3,50,000
Australia	6	10 Million	6,00,000

Monitoring Load Balancing Algorithm, TLBA-Throttled Load Balancing Algorithm, and the proposed method, ETLBA-Enhanced Throttled Load Balancing Algorithm.

In terms of the performance parameters mentioned earlier, the results based on simulation show that the suggested method Enhanced Throttled Load Balancing Algorithm (ETLBA) performed better than the existing methods Round Robin, Active Monitoring Load Balancing Algorithm (AMLBA), and Throttled Load Balancing Algorithm (TLBA). Below, we'll go over the same.

5.1. Overall Response Time. The amount of time it takes for the desired process to finish processing is known as response time. Round robin has an average reaction time of 179.34 milliseconds, AMLBA is 177.43 milliseconds, throttled is 171.61 milliseconds and the proposed ETLBA is 164.53 milliseconds. Based on our findings, the load balancing algorithm is not well-suited to Round Robin, AMLBA, TLBA, due to the increased response time it requires. Consequently, the proposed Enhanced throttled load balancing algorithm ETLBA outperform the other three existing in terms of response time. Fig. 5.1 graph express the performance of proposed method ETLBA in terms of Avg. Overall Response Time.

5.2. Data Center Processing Time. A load balancer's processing time in a data center is the amount of time it takes to handle all of the necessary requests. The proposed method ETLBA consumes 16.32 milliseconds, TBLA takes 23.89 milliseconds, AMLBA takes 41.12 milliseconds, and Round-robin takes 43.35 milliseconds. When compared to the RR, AMLBA and TBLA algorithms with respect to processing time in the data center, ETLBA is appropriate. The same is shown in the below graph in Fig. 5.2.

5.3. Data Center Request Servicing Time. Service time for request is also determined by the datacenter's serving time, which does not include the time it takes to transport data to the user. The proposed method ETLBA consumes 216.28 milliseconds, TBLA takes 37.17 milliseconds, AMLBA takes 45.36 milliseconds, and Round-robin takes 52.25 milliseconds. When compared to the RR, AMLBA and TBLA algorithms with respect

Table 4.2: User Database Configuration

User Database	Zonal Id	Requests per user per Hx	Data Size per request (byte)	Peak Hour (GMT)	Avg. Users in peakbss	Avg. Users nonpeals hrs
UB 1	1	60	100	13: 00 15: 00	5,00,000	50,000
UB 2	2	60	100	15: 00 17: 00	2,00,000	20,000
UB 3	3	60	100	20: 00 22: 00	3,00,000	30,000
UB 4	4	60	100	01: 00 03: 00	2,35,000	23,500
UB 5	5	60	100	21: 00 23: 00	35,000	3,500
UB 6	6	60	100	09: 00 11: 00	80,000	8,000

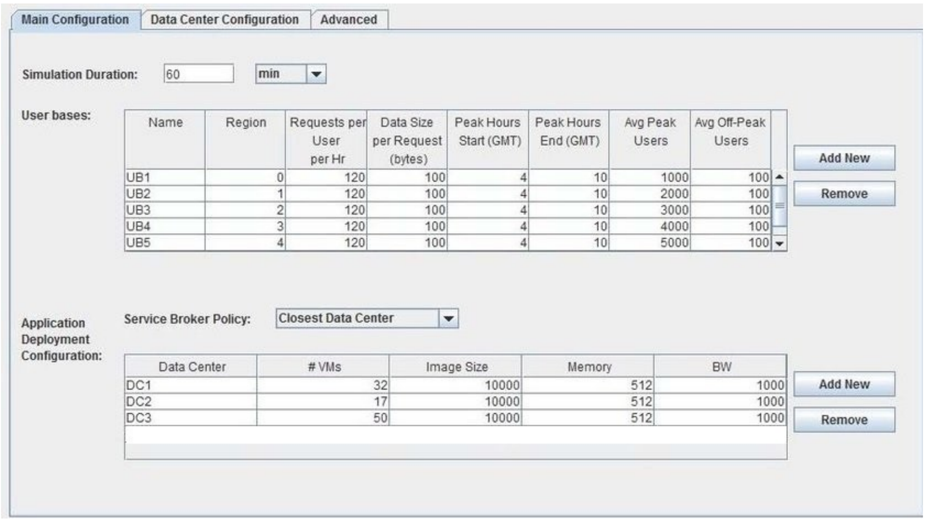

Fig. 4.2: Sample Settings in Cloud Simulator

Table 4.3: User Database Configuration

Name	Data Center1	Data Center2	Data Center3
Region-ID	1	3	S
Configuration	X86	X86	X86
Operating System	Linux	Linux	Linux
Virtual Machine	Xen	Xen	Xen
Cost of VM	0.2 $ / hr	0.2 $ / hr	0.2 $ / hr
Cost in terms of Mernory	0.10 $ / sec	0.10 $ / sec	0.10 $ / sec
Cost of Data Transfer	0.2 $ / GB	0.2 $ / GB	0.2 $ / GB

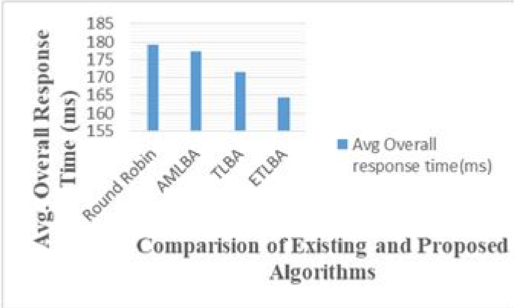

Fig. 5.1: Avg. Overall Response Time Comparison

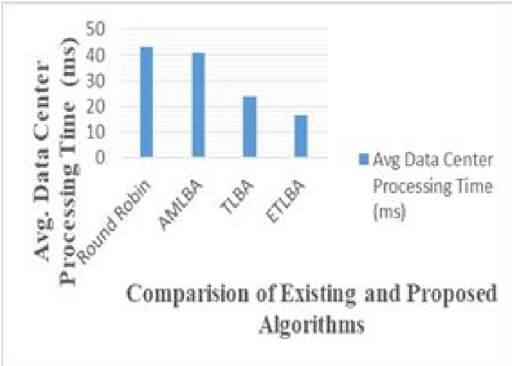

Fig. 5.2: Avg. Data Center Processing Time Comparison

Table 5.1: Overall Response Time

Algorithm	Overall resp onsetime(ms)		
	Avg(ms)	Min(ms)	Max (ms)
Round Robin	179.34	46.43	671.56
AMLBA	177.43	45.24	669.46
TLBA	171.67	43.63	663.45
ETLBA	**164.53**	**41.28**	**651.43**

to requests processing time in the data center, ETLBA outperforms the existing algorithms and the same is depicted in the below graph in Fig. 5.3.

5.4. Data Center Request Servicing Time. The datacenter cost is determined by adding up all the works performed by the virtual machines. These works can include diverse tasks including ALU operations, CU operations, storage operations, and more. Round robin method has incurred cost of 29.03 $, while 26.82 $, 23.93 $ for AMLBA and TLBA respectively, whereas the proposed ETLBA is 21.78 $. Based on simulation findings, the Round Robin, AMLBA, TLBA, are not cost effective, compared with the proposed Enhanced throttled load balancing algorithm ETLBA. The graph in Fig. 5.4 shows the data center costs of all the existing and proposed algorithms.

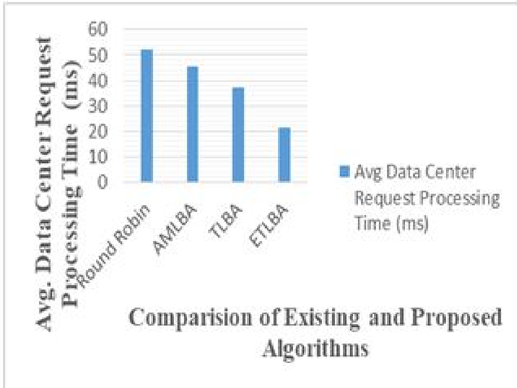

Fig. 5.3: Avg. Data Center Request Processing Time comparison

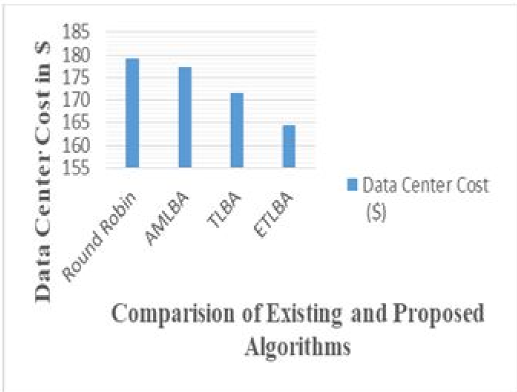

Fig. 5.4: Data Center Cost comparison

Table 5.2: Data Center Processing Time

Algorithm	Data Center Processing Time		
	Avg(ms)	Min(ms)	Max (ms)
Round Robin	43.35	8.89	251.33
AMLBA	41.12	8.89	252.15
TLBA	23.89	7.92	242.28
ETLBA	16.32	7.14	233.64

5.5. Analysis and Discussion. Table 5.1, 5.2, 5.3, 5.4 shows simulation results of all the four algorithms, Round robin, AMLBA, TLBA and the proposed ETLBA in terms of Overall response time, Data Center Request servicing time, Data Center Cost, and Data processing time. Out of the four methods under comparison, the ETLBA algorithm consistently produces the lowest level of outcomes. Since the four performance scenarios chosen are significantly distinct from one another, we can observe that ETLBA remains stable despite the variable inputs and performs well in all cases.

Table 5.3: Data Center Request Processing Time

Algorithm	Data Center Request Processing Time		
	Avg(ms)	Min(ms)	Max (ms)
Round Robin	52.25	10.19	263.27
AMLBA	45.36	10.09	243.27
TLBA	37.17	9.14	232.58
ETLBA	21.28	8.26	223.24

Table 5.4: Data Center Cost

Algorithm	Data Center Cost		
	VM Cost ($)	Data Transfer Cost ($)	Total ($)
Round Robin	179.34	46.43	671.56
AMLBA	177.43	45.24	669.46
TLBA	171.67	43.63	663.45
ETLBA	**164.53**	**41.28**	**651.43**

6. Conclusion. A novel method 'Enhanced Throttled Load Balancing Algorithm" for effective load balancing is provided in this article. This method implements the strategy of allocating the resources to the priority user requests based on nearest data center, index and usage of the datacenter and its virtual machines It delves into the strategy of enhancing load balancing to boost cloud computing performance in terms of response time, data center costs. Various load balancing approaches are simulated along with the proposed method for showcasing the performance, using the Cloud Analyst tool. These techniques include Round Robin, Active Monitoring AMLBA, TLBA (Throttle Load Balancing Algorithm), and the proposed ETLBA (Enhanced Throttled Load Balancing algorithm). Better time response, less resource starvation, more powerful virtual machines to handle more requests, and cost savings are all achieved by using the suggested algorithm (ETLBA). Consistently decreasing datacenter costs with this approach demonstrate the practical applicability and future development prospects of the ETLBA algorithm. ETLBA has the possibility of using and implementing the results of future research and testing. ETLBA can further enhanced by including the region proximity based on the data centers. Real-world testing and research can be conducted with the proposed ETLBA in a datacentre.

REFERENCES

[1] Singh, Archana, and Rakesh Kumar. , *Performance evaluation of load balancing algorithms using cloud analyst.* 10th International Conference on Cloud Computing, Data Science and Engineering (Confluence). IEEE, 2020.

[2] Narale, Snehal A., and P. K. Butey. , *"Throttled load balancing scheduling policy assist to reduce grand total cost and data center processing time in cloud environment using cloud analyst."* 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2018.

[3] Roy, S., Hossain, D. M. A., Sen, S. K., Hossain, N., & Al Asif, M. R. , *"Measuring the performance on load balancing algorithms."* Global Journal of Computer Science and Technology, 19, 41-49, 2019.

[4] Volkova, V. N., Chemenkaya, L. V., Desyatirikova, E. N., Hajali, M., Khodar, A., & Osama, A. , *"Load balancing in cloud computing."* 2018 IEEE conference of russian young researchers in electrical and electronic engineering (EIConRus). IEEE, 2018.

[5] Maithili, K., V. Vinothkumar, and P. Latha. , *"Analyzing the security mechanisms to prevent unauthorized access in cloud and network security"* Journal of Computational and Theoretical Nanoscience, 2018.

[6] Singh, Archana, and Rakesh Kumar. , *"Performance evaluation of load balancing algorithms using cloud analyst."* 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2020.

[7] Hota, Arunima, Subasish Mohapatra, and Subhadarshini Mohanty., *"Survey of different load balancing approach-based algorithms in cloud computing: a comprehensive review."* Computational Intelligence in Data Mining: Proceedings of the International Conference on CIDM 2017. Springer Singapore, 2019.

[8] Mohapatra, Subasish, Subhadarshini Mohanty, and K. Smruti Rekha., *"Analysis of different variants in round robin*

algorithms for load balancing in cloud computing." International Journal of Computer Applications 69.22, 17-21, 2013.
[9] Gao, G., Hu, H., Wen, Y., & Westphal, C. , *"Resource provisioning and profit maximization for transcoding in clouds: A two-timescale approach."* IEEE Transactions on Multimedia 19.4, 836-848, 2016.
[10] Karthick Raghunath, K. M., Manjula Sanjay Koti, R. Sivakami, V. Vinoth Kumar, Grande NagaJyothi, and V. Muthukumaran, *"Utilization of IoT-assisted computational strategies in wireless sensor networks for smart infrastructure management"* International Journal of System Assurance Engineering and Management, 1-7, 2022.
[11] Phi, Nguyen Xuan, and Tran Cong Hung. , *"Load balancing algorithm to improve response time on cloud computing."* International Journal on Cloud Computing: Services and Architecture 7.6, 1-12, 2017.
[12] Narale, Snehal A., and P. K. Butey. , *"Throttled load balancing scheduling policy assist to reduce grand total cost and data center processing time in cloud environment using cloud analyst."* 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). IEEE, 2018.
[13] Shafiq, Dalia Abdulkareem, N. Z. Jhanjhi, and Azween Abdullah. , *"Load balancing techniques in cloud computing environment: A review."* Journal of King Saud University-Computer and Information Sciences 34.7, 3910-3933, 2022.
[14] Bhandari, Anmol, and Kiranbir Kaur., *An enhanced post-migration algorithm for dynamic load balancing in cloud computing environment.* Proceedings of International Ethical Hacking Conference 2018: eHaCON 2018, Kolkata, India. Singapore: Springer Singapore, 2018.
[15] Le, Hieu N., and Hung C. Tran., *Ita: The Improved Throttled Algorithm of Load Balancing On Cloud Computing.* International Journal of Computer Networks & Communications (IJCNC), 14, 2022.
[16] Patel, Hetal, and Ritesh Patel , *Design and Evaluation of Wi-Fi Offloading Mechanism in Heterogeneous Networks*, International Journal of e-Collaboration, 2021, 60-70.
[17] Vinoth Kumar, V., S. Ramamoorthy, V. Dhilip Kumar, M. Prabu, and J. M. Balajee , *Cloud analyst: an insight of service broker policy.*, International Journal of Advanced Research in Computer and Communication Engineering, 2015, 122-127.
[18] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya, *"CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms."* Software: Practice and Experience, 23-50, 2011.
[19] Buyya, Rajkumar, and Manzur Murshed. , *"Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing."* Concurrency and computation: practice and experience 14.13-15, 1175-1220 2020.
[20] Muthukumaran, V., V. Vinoth Kumar, Rose Bindu Joseph, Meram Munirathnam, I. S. Beschi, and V. R. Niveditha , *"Efficient Authenticated Key Agreement Protocol for Cloud-Based Internet of Things."* In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2022, 365-373. Singapore: Springer Nature Singapore, 2022.

Edited by: Polinpapilinho F. Katina
Special issue on: Scalable Computing in Online and Blended Learning Environments: Challenges and Solutions
Received: Jan 7, 2023
Accepted: Mar 23, 2024