..

Dynamic Resource Management: Load Balancing Algorithms for Heterogeneous Cloud Environments

Jasobanta Laha[1], Sabyasachi Pattnaik2, Chinmaya Kumar Nayak[3], Kumar Surjeet Chaudhury[4] and Subhra Swetanisha[5]

[1]Research Scholar, PG Department of Computer Sc., Fakir Mohan University, Balasore &
Assistant Professor, Faculty of Engineering & Technology (FET), Sri Sri University, Cuttack
MailID - yeshmcp@gmail.com
[2]Professor, PG Department of Computer Sc., Fakir Mohan University, Balasore
MailID - spattnaik1965@rediffmail.com
[3]Associate Professor, Faculty of Engineering & Technology (FET), Sri Sri University, Cuttack
MailID - cknayak85@gmail.com
[4]Assistant Professor, School of Computer Engineering KIIT Deemed to be University, Bhubaneswar
MailID – surjeet.chaudhuryfcs@kiit.ac.in

Abstract

The Pay as You Use (PaYU) model forms the foundation of cloud mobility, enabling customers to access diverse resources such as computing power, energy, and storage as per their demand. In today's world, the different companies across various industries are recognizing the advantages of cloud computing and transitioning to it. Cloud computing has many advantages over traditional computing. However, cloud storage still has some problems, such as inadequate allocation of resources and poor user service. The overall performance of cloud systems is also affected by heterogeneous cloud resources. In this work, we propose an improved load measurement method for deploying virtual machines in heterogeneous cloud. To maintain adequate equilibrium, our innovative method effectively allocates independent client requests for virtual machines within the data cloud. Compared to the Throttled and Round Robin algorithms, the findings indicate that the data center requires less time to respond to and handle user requests.

Keywords

Load Balancing, independent task, datacentre processing time, heterogeneous cloud environment, response time

Introduction

In the field of computer science, cloud computing stands out as the most cutting-edge and innovative technology currently accessible. Cloud denotes a network of IT resources, and computing involves utilizing these resources remotely to carry out tasks, with payment occurring only for the services utilized. It is a web-based technology that offers a variety of cloud services. These services are effective, reliable and affordable, and you can access them anytime, anywhere, on any device. It provides on-demand self-service and allows us to access (services) as needed without the cooperation of others. There are various cloud service providers available today, including Google Cloud, Amazon Web Services (AWS), and others [1].

In cloud computing, four distinct deployment options are available.

Private cloud: It is a cloud computing concept that offers an environment in which resources and applications are accessible exclusively by the designated customer.
Public cloud: This type of computing structure makes resources and applications available to anybody who wants to utilize them through a public internet connection.

Community Cloud: A community cloud is a collaborative endeavour that offers an infrastructure that is shared by several organizations, allowing for the sharing of applications between them.
Hybrid: A hybrid cloud combines elements of both public and private clouds.

The two crucial ideas in cloud computing are referred to by the name "cloud" [2]:
Abstraction: Users and developers are not made aware of the technical specifics for cloud systems. Both the physical equipment that executes the program and the location of the cloud's data storage are unknown.
Virtualization: Hypervisors are used to represent and deliver logical resources in place of physical computer resources. By pooling and sharing resources, virtualization technology enables cloud service providers to save prices.

Load Balancing

The government, business, and educational sectors are all gravitating toward the cloud these days. The amount of requests sent to the cloud server due to the widespread usage of cloud services is increased by altimetry. To deal with this request, load balancing is employed. In the Figure - 1 below, load balancing's fundamental operation is depicted:

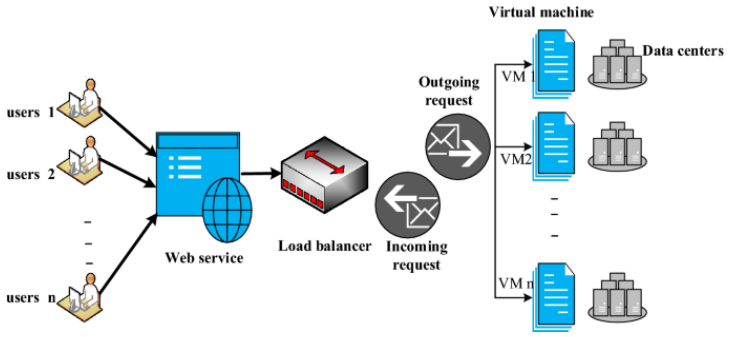


Figure - 1 The way load balancing operates on cloud servers is shown.

Enhancing resource usage, response times, and system performance is the main goal of virtual machine node balancing. Load balancing algorithms come in two different varieties. Static load balancing comes initially, followed by dynamic load balancing. Prior to task execution, assign a task to the server first. It is an assigned task. Second, attach a job to the server when it has been completed. It is not a mandated task. Response time, throughput, overhead, and other characteristics are used to analyze the load balancing algorithm's efficiency.

Literature Survey of Existing Algorithm

Load balancing is one of the major difficulties that experts are concentrating on as cloud computing gains substantial pace. The following section discusses a few load balancing research studies.

The efficacy of the load balancing method has been assessed, according to Ramadhan et al. [3]. The researchers observed that there are a large number of different types of delivery being requested by users in need of cloud services. The authors utilized the Throttled, Round Robin (RR) and Equally Spread Current Execution (ESEC) algorithms in their simulation investigations [4-5].The results indicate an increase in the average reaction time of the Throttled algorithm. Domanal and Reddy [6] proposed the Modified Throttled Algorithm to evenly distribute incoming user workloads among computer resources. This approach, similar to the Throttled method, keeps a database of virtual machines (VMs) and their respective states. The procedure starts at the supplied index and chooses a virtual machine (VM) from the VM index database in response to user requests. Subsequent requests prompt the algorithm to scan the table from the previously allocated VM onwards. However, the present Throttled Algorithm is used to systematically scan the table starting from the beginning index with each new

request. In comparison to the Throttled and Round Robin Algorithms, this approach shows an enhancement in user response time [4]. The allocation of resources does not consider the processing power of each VM.

Somani and Ojha [7] introduced a hybrid load balancing approach that combines the "Throttled" and Round Robin algorithms. The throttled algorithm chooses one virtual machine (VM) and uses all of the VMs it has selected to distribute it in a round-robin fashion. While this method demonstrates strong performance in a homogeneous cloud environment, it may face challenges in maintaining performance in a heterogeneous cloud.

In their publication, Kushwaha and Gupta [8] addressed the challenge of high workload during peak times in utilizing cloud services. The primary focus of their work is the optimization challenge during peak hours. They conducted a performance evaluation of the Throttled, Round Robin, and Equally Spread Current Execution algorithms [4–5] during high internet application usage. The results indicated that the "Performance Optimized Service Broker Policy" combined with the "Throttled" algorithm effectively managed the peak hour workload. While the diversity of cloud resources may influence response and processing times, the authors considered homogeneous clouds for their simulations.

The classification technique for categorizing virtual machines and user workloads has been explored by Elroub and Gherbi [9]. While CPU and RAM use are taken into account for virtual machine classification, task categorization takes into account the size of the user work and any accompanying log file data. The "Active Monitoring Load Balancing (AMLB)" technique was improved by Singh and Prakash [10] by giving each virtual machine (VM) a weight based on factors including RAM, CPU speed, number of processors, and network bandwidth. This study's primary objective is to make effective use of virtual machines by allocating jobs to them based on a weight factor that has been established. For computing the weight factor, only the initial configuration of the virtual machine's processing components is considered. An effective "Time Stamp Based Stateful Throttled VM Load Balancing Algorithm" was proposed by Makroo and Dahiya [11]. The status of the userbase request allocation is kept for further usage as soon as the appropriate virtual machine allocation is selected. Due to the extra overhead needed to keep the user base request allocation in a consistent state, the strategy might affect overall performance.

The load balancing approach has not taken into consideration the various cloud datacenter resource designs and the actual processing element usage in many prior study investigations. These elements must be considered by the effective load balancing algorithm in order to properly utilize all of the cloud datacenter's resources based on their processing capacities. By keeping these things in mind, cloud users' experiences and datacenter processing times could be improved. It has been demonstrated in earlier research that the throttled algorithm outperforms the conventional load balancing method.

In order to enhance response times and datacenter processing times in a heterogeneous cloud environment, this study addresses all of these concerns and suggests an alternative to the throttled load balancing technique.

Traditional Algorithms

Round Robin Algorithm

This technique is straightforward and easy to understand and is popular in cloud computing [12,13]. Without taking into account each virtual machine's processing power, it distributes the user requests among them in a circular pattern. For datacenters with identical processing power across all virtual machines, this technique performs quite well. In other words, it performs effectively in a cloud environment that is uniform.

Throttled Algorithm

The thresholding principle is the fundamental building block of the throttled algorithm [14,15,16,17]. A single virtual machine (VM) is limited in the number of requests it can assign during this process to the value indicated by its threshold. The time taken for the virtual machine's state to become available increases with the number of requests. Consequently, it affects the datacenter processing time as well as the overall user response time. Diverse clouds are not suitable for Throttle algorithms.

A Novel Approach

Our Novel Approach research's main objective is to reduce client request response and processing times in heterogeneous cloud datacenters. To ascertain which virtual machine (VM) is the most appropriate for a user request, the Datacenter Controller (DCC) validates the proposed algorithm put into place at the VmLoadBalancer. The suggested work's performance analysis is contrasted with those of the "Round Robin" and "Throttled" algorithms.

During the allocation process, the suggested innovative method considers each VM's actual processing power. The AVAILABLE Table and the BUSY Table are two separate tables that it keeps up to date.

The first step in the process is to search the "Available VMs Table" for a virtual machine (VM) that is appropriate to execute in response to a user request. If the algorithm finds no suitable virtual machine (VM) in the "Available VMs Table," it then searches the "Busy VMs Table" for a VM with the necessary capacity. The purpose of enforcing predefined threshold levels is to prevent overloading the "Busy VMs" and guarantee proper resource allocation.

The number of processors, memory size, and processing speed of virtual machine resources can vary significantly in a heterogeneous cloud environment. Upon VMs' unavailability, our approach doesn't queue requests; instead, it prioritizes execution on VMs with enough resources straight from the "Busy VMs Table."

This updated algorithm offers improved resource utilization and ensures that user requests are efficiently executed while considering the diverse capabilities of available VMs in the cloud environment.

Figure - 2 illustrates **A Novel Approach algorithm's operation**, and Algorithm - I describes the stages.

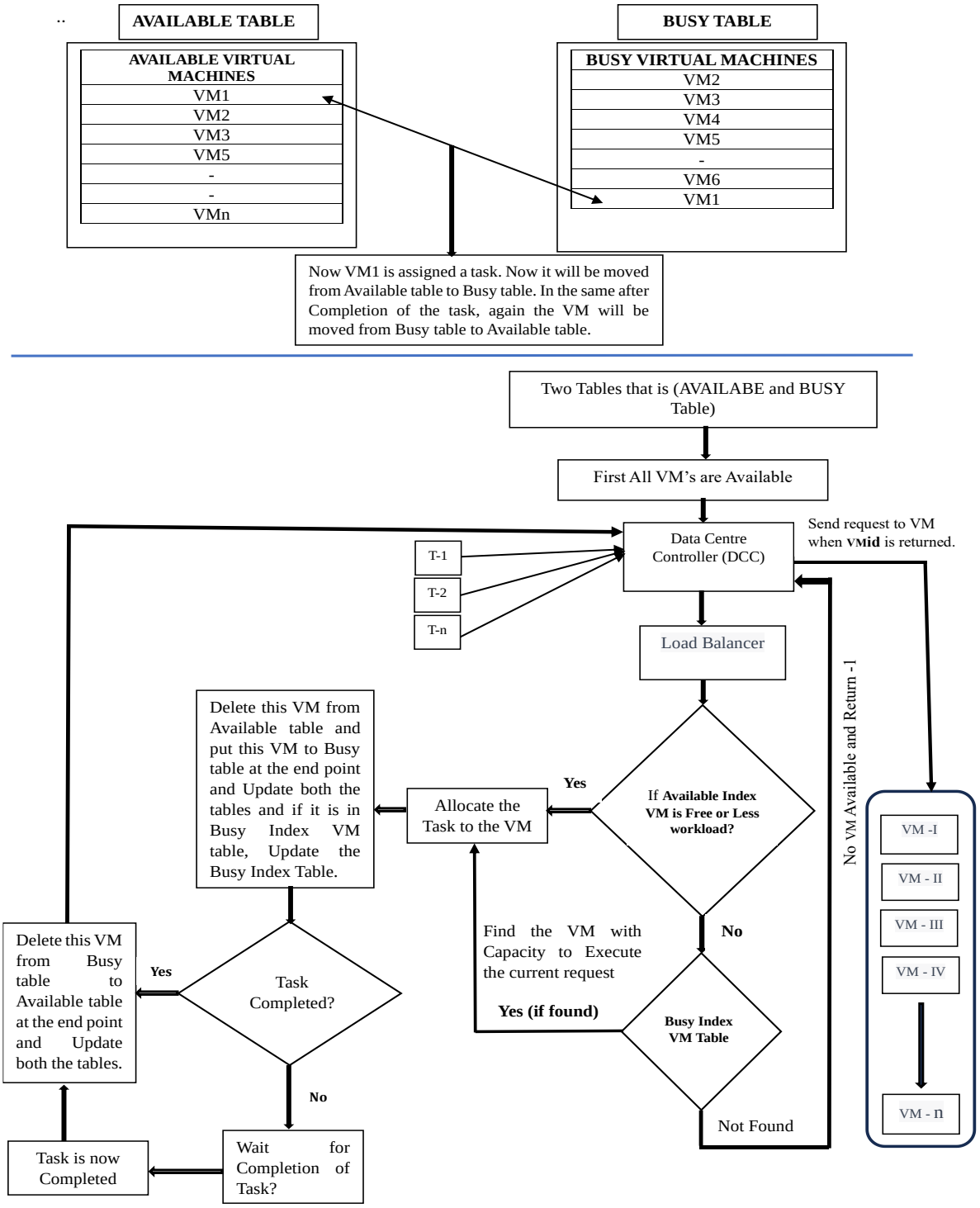


Figure – 2: This demonstrates the recommended load balancing approach for efficient virtual machine (VM) allocation in diverse cloud environments.

Our new approach first ascertains the capacity of each virtual machine (VM) before searching the BUSY table for an appropriate one. Additionally, it determines the mean capacity and only chooses virtual machines with capacities greater than or equal to the mean. Round robin is how these virtual machines receive the requests. In order to stop more overload on BUSY VM, the method uses two different threshold values:

Threshold T1 (Cloudlets): This number indicates the most cloudlets that can be running simultaneously on each selected virtual machine (VM). It assists in preventing overload from BUSY VM on the chosen VM.

Algorithm – I:

Based on processing power, VM allocation is suggested.

Input:

- Requests from Cloud Users
- Virtual Machines (VMs) that Are Available

Output:

- Virtual Machine Index Identifier

Initialization:

- Set the initial values for the two index tables, BUSY (empty) and AVAILABLE (holds indices of all VMs available).

S - 1: Initialize

- AVAILABLE ← Listings of every VM that is available
- BUSY ← Empty

S - 2: The Datacentre Controller (DCC) receives a new cloud service request.

S - 3: DCC queries the algorithm for current request allocation on a VM

S - 4: Algorithm scans the AVAILABLE index table of VMs:

- If a VM is available:
  - **a.** VM Index id is returned to DCC.
  - **b.** DCC forwards the current request to the respective VM based on the received VM id.
  - **c.** DCC informs the algorithm of this new allocation for VM state updates.
  - **d.** Algorithm updates the status of VM and lists accordingly.
- Else (If no available VM is found):
  - **e.** The "BUSY" list of virtual machines is parsed by the algorithm to identify a virtual machine that can handle the current request.
    Locate an efficient Virtual Machine from the BUSY Index Table using Algorithm II.
    If found, again repeat Step **a** to **d**.
  - **f**. returns to DCC a negative number (-1). When DCC receives a negative value, it adds the current request to the queue.

S - 5: After Virtual Machine (VM) finishes processing the request:

- The Virtual Machine sends the response to the Datacentre Controller (DCC).
- The Datacentre Controller (DCC) notifies the algorithm of Virtual Machine deallocation.

S - 6: The Datacentre Controller (DCC) examines the request queue for any pending requests. If found, it proceeds from S - 3.

S - 7: Iterate from S - 2 whenever a new request arrives.

Threshold T2 (Burst): According to this configuration, a major number of requests can be processed for allocation on a BUSY Virtual Machine. As a result, depending on the value of this threshold, a certain number of requests may be handled. The algorithm does not allocate resources for or process more requests after it reaches this maximum level. It instructs the data center to enqueue the request, initiates a reduction in the threshold value, and returns -1 for a comparable request in subsequent instances. The threshold value is reset to its initial value by the algorithm when it reaches zero. It simply offers a means of avoiding a spike in requests and aids in preventing overload on the BUSYVM. This describes the sub-algorithm of Algorithm II that finds the effective virtual machine (VM) from the BUSY list.

In cases where the AVAILABLE index table is empty, our proposed approach aids in selecting the appropriate Virtual Machine from the BUSY index table for request allocation. Both the processing time in the datacenter and the time it takes for users to respond are reduced as a result.

Algorithm II: Proposed technique for identifying an Efficient Virtual Machine from the BUSY Index Table

Input: List of BUSY Virtual Machines (VMs)
Output: Identifier of Virtual Machine Index
St - 1: Compute the current capacity of each Virtual Machine (VM) in the BUSY Index Table.
St - 2: Compute the average capacity of all BUSY VMs.
St - 3: Choose a VM in a Round-Robin fashion to process the current request.

- **If** (the capacity of the selected VM <= Average capacity of all BUSY VMs And
  the number of cloudlets (tasks or workloads) on the VM >= to a predefined threshold (Threshold T1**)**
  Then
  Send back the index of this Virtual Machine to the Datacentre Controller (DCC).
  Else
  Return -1 to the DCC.

St - 4: Load Balancing and Threshold Handling

- Check if Threshold T2 is greater than or equal to a maximum limit:
  - If true, repeat St - 3 for the subsequent request using the round-robin method.
  - If false:
    - Put the current request in a waiting queue.
    - Return -1 to the DCC.
    - Decrement Threshold T2 by one unit.

St - 5: Threshold Management

- Check if Threshold T2 has become ZERO:
  - If true, reset Threshold T2 to its original value.
  - Continue from St - 3 (repeating Steps **a** to **d**) if a VM was found in the previous steps.

In essence, this improved algorithm uses the capacity and quantity of cloudlets that busy virtual machines (VMs) are already processing to assign incoming requests to VMs. Additionally, it provides Threshold T2 for request queuing and load balancing, ensuring that the system is responsive even under heavy loads. To ensure the algorithm remains responsive, threshold T2 is reset when it approaches zero.

Configuration for Simulation:

Since most users use the app in the evening for two hours after work, we created six UserBbases that correspond to six zones with a fixed time zone. that each user submits a fresh internet request every four minutes:

Here,

- Peak Hour: peak access period.
- Number of concurrent users visiting during peak hours: The number of users online at the same time.
- Concurrent Online Users During Off-Peak Hours: the number of users who log on at a slower time.

These settings are made on the Configure Simulation class' Main Configuration tab. The virtual machine (VM) setup may be found on this tab as well.

Table - 2: Different Features of the User Base

Region	Name	Time Zone (GMT)	Requests Per User Per Hour	Data size per Request (bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Average Peak Users	Average Off - Peak Users
0	UB1	GMT - 6.00	60	100	13	15	350000	35000
1	UB2	GMT - 4.00	60	100	15	17	100000	10000
2	UB3	GMT + 1.00	60	100	20	22	280000	28000
3	UB4	GMT + 6.00	60	100	1	3	145000	14500
4	UB5	GMT + 2.00	60	100	21	23	40000	4000
5	UB6	GMT + 10.00	60	100	9	11	75000	7500

We presume that only ten percent of users are active during non-peak times. Additionally, our assumption is that every four minutes, a new request is submitted on the internet. Peak hours are calculated by considering the fact that many users usually visit the application in the evening after work, over a period of approximately two hours. Table – 3, provides specifics on other variables.

The physical machines present in a datacenter serve as hosts for the virtual machines (VMs). The actual computers used to build a heterogeneous cloud environment have varying Random Access Memory(Capacity of RAM - 2GB to 4GB), CPU (number of CPU may be 2 or 4 or 6 or 8) and processing power capacities in terms of MIPS .

Table – 3 Different Parameters

Parameters	Values
Virtual Machine (VM)	
Size of Image	10000 Bytes
Size of Memory	512 MB
Capacity of Bandwidth	1000 Bytes
System Architecture	x86
Physical System at Datacentre	
Type of Operating System	Linux
Type of VMM/Hypervisor	Xen
Size of Main Memory	2048 / 4096 MB
The Policy of VM Scheduling	Time Shared

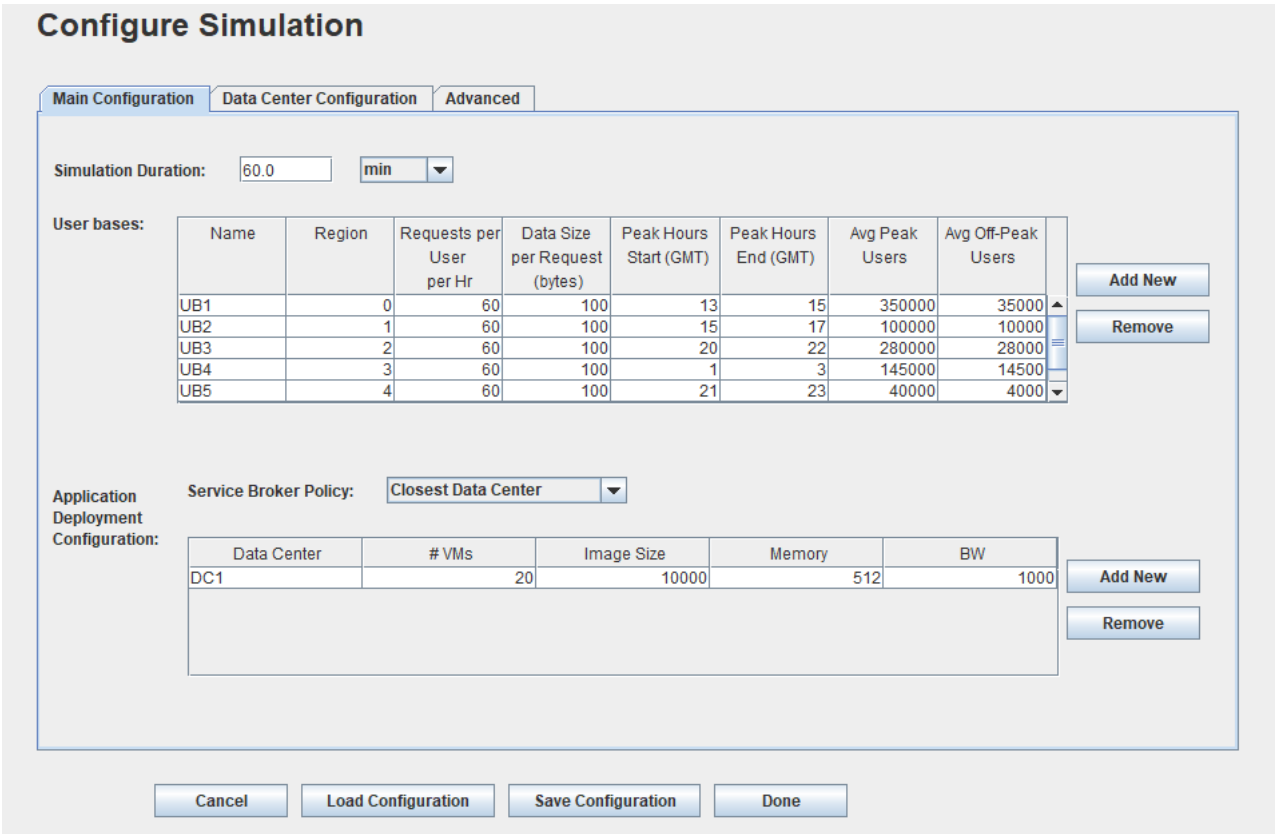


Figure – 3 User and VMs configuration settings

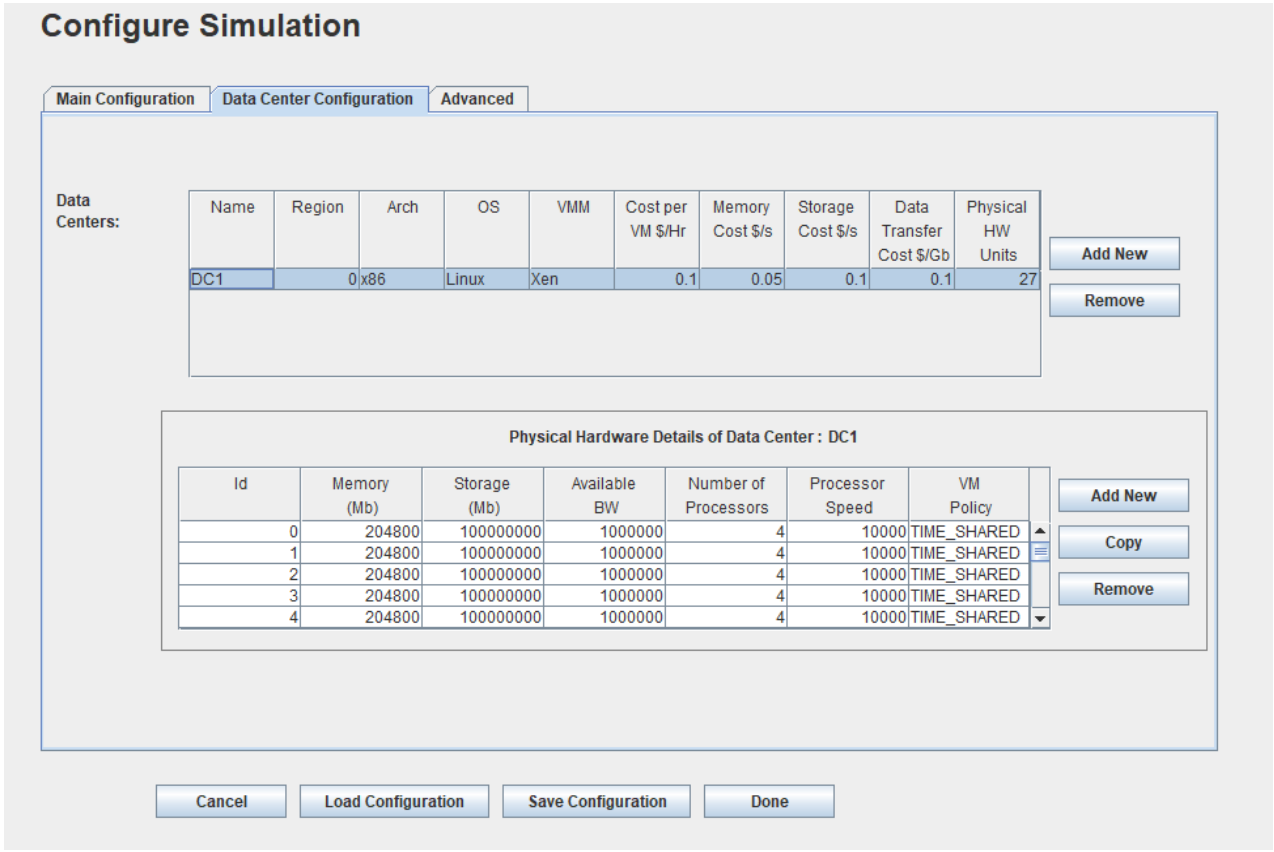
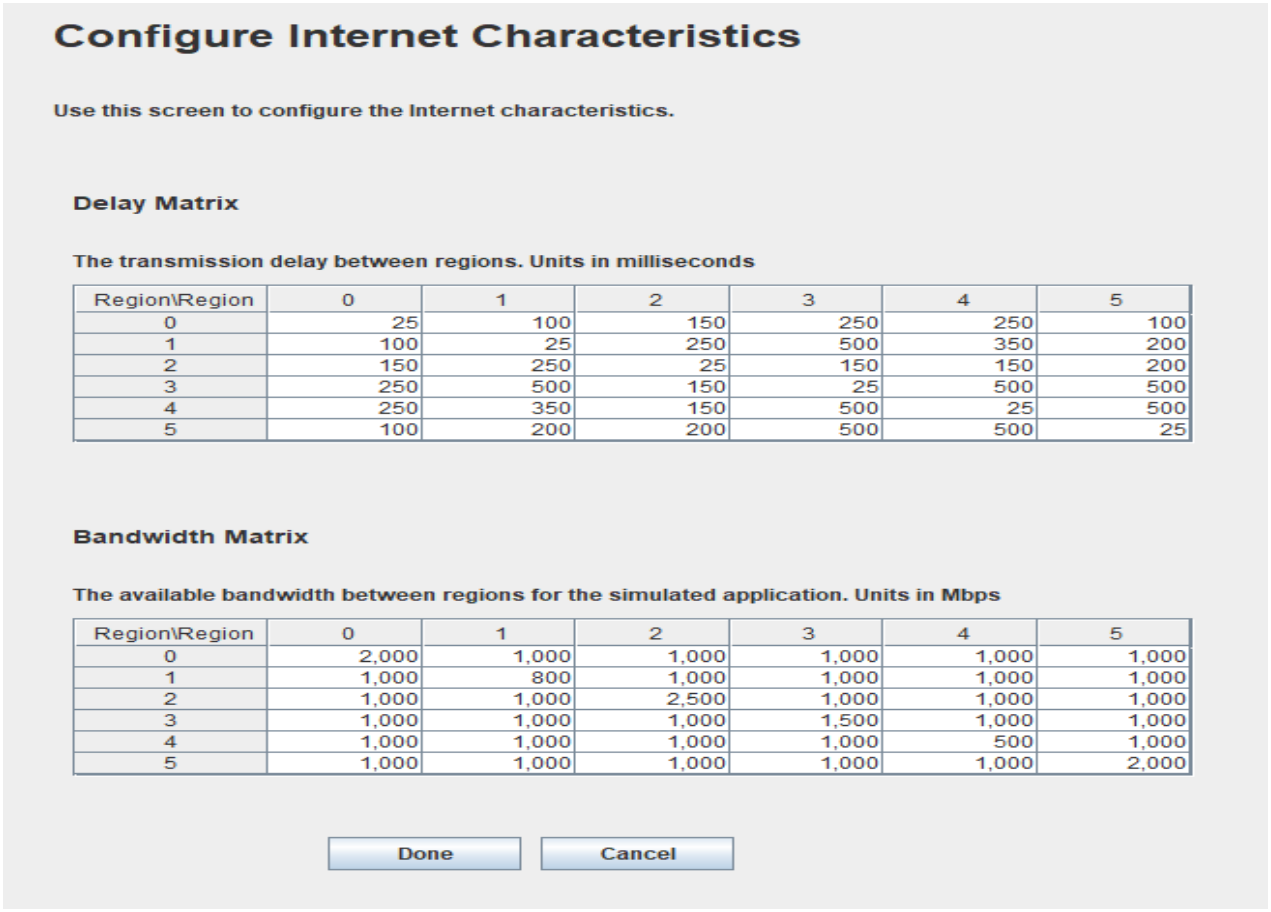


Figure – 4 Datacentre configuration parameters



Configure Internet Characteristics

Use this screen to configure the Internet characteristics.

Delay Matrix

The transmission delay between regions. Units in milliseconds

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

Bandwidth Matrix

The available bandwidth between regions for the simulated application. Units in Mbps

Region\Region	0	1	2	3	4	5
0	2,000	1,000	1,000	1,000	1,000	1,000
1	1,000	800	1,000	1,000	1,000	1,000
2	1,000	1,000	2,500	1,000	1,000	1,000
3	1,000	1,000	1,000	1,500	1,000	1,000
4	1,000	1,000	1,000	1,000	500	1,000
5	1,000	1,000	1,000	1,000	1,000	2,000

Done Cancel

Figure – 5 Internet feature configuration

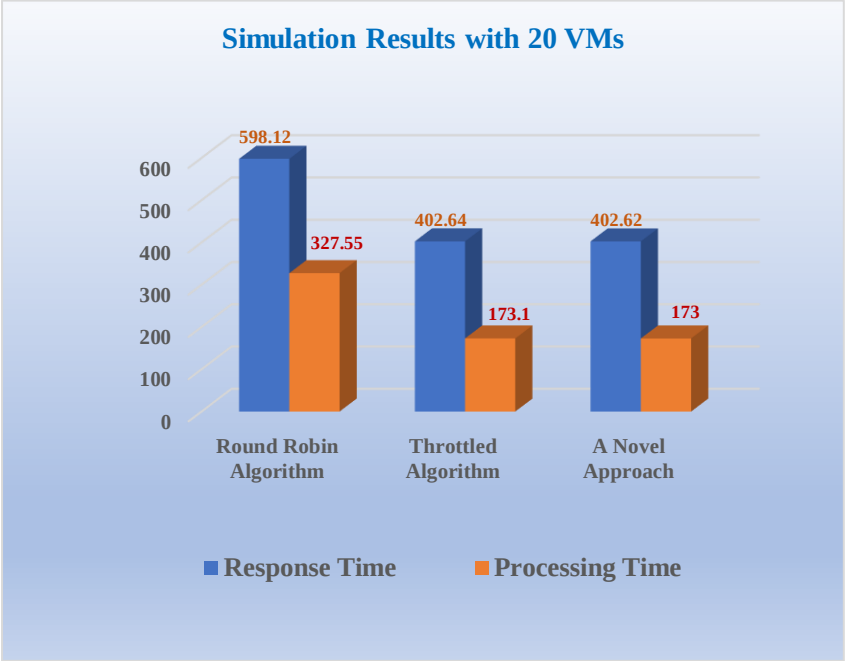


Figure – 6: outcomes of the 20-VM simulation

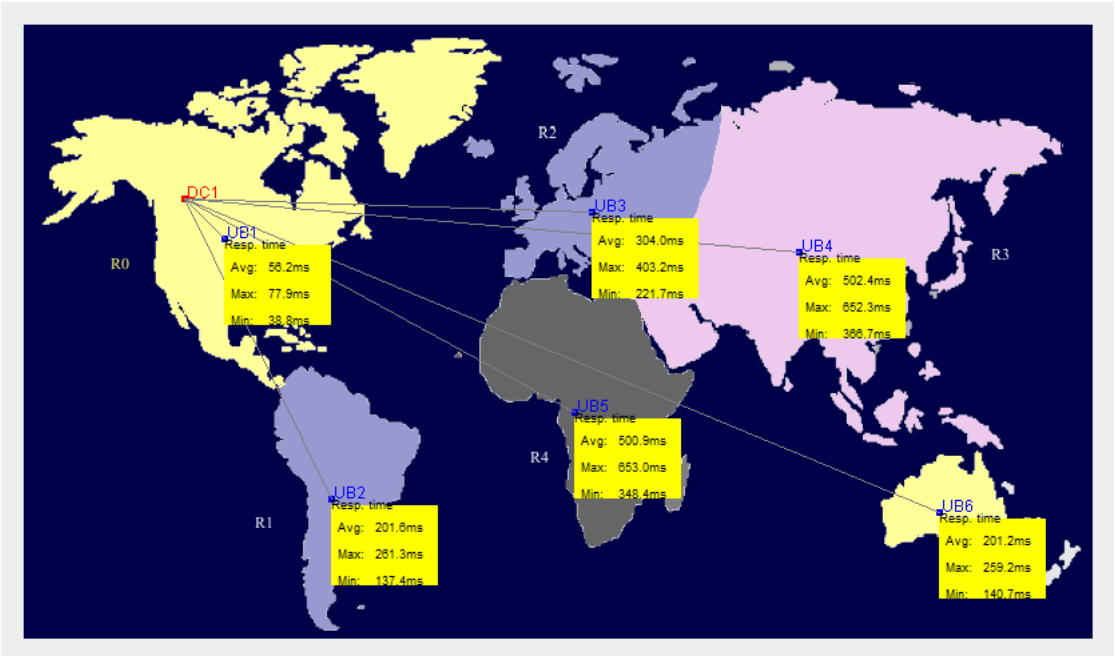


Figure – 7: An image of the regions used in the simulation

Analysis Results:

Use twenty virtual machines (VMs) to simulate:

There is no need to allocate queues because the Round Robin algorithm distributes the requests across the VMs evenly. When a significant number of virtual machines (VMs) were found in the state index table utilizing a detection technique that went from the beginning to the end of the table, the throttled algorithm would cause the status of requests to queue. The system simply needs to distribute requests to VMs in the Available index table using the A Novel Approach since it uses two status index tables (Available Index and Busy Index) and does not require searching for them. Eliminating the need for the system to be queued up increases processing time in the datacenter.

The required distribution to the virtual machine (VM) rotates in a circle using the Round Robin technique, as shown in **Figure 6**, without taking the VM's state into account. The datacenter's processing and response time are delayed as a result. The system's user ratio is significantly greater than the other two techniques when compared to UserBase. Our Novel Approach algorithm's system response time and datacenter processing time are slightly slower than those of the other two throttled algorithms.

Conclusion

Finding practical solutions to enhance cloud services has become more difficult as cloud computing's prominence grows. Cloud services may be improved with the right load balancing. Cloud services may be improved with the right load balancing. Although a number of load balancing strategies have been put out in research, none of them take into account the different resource configurations seen in cloud datacenters. Additionally, the way resources are processed at the moment of consumption has not been taken into account. In order to provide a strategy for efficient virtual machine (VM) allocation in heterogeneous clouds, our study considered these issues. Accelerating response times and datacenter processing is the primary objective of our proposed novel work. The suggested algorithm determines both the individual VMs' present capacities as well as their combined average capacities. The algorithm distributes the requests to an appropriate VM based on these capabilities and threshold values that have been pre-established to prevent overload. When Round Robin (RR) and the Throttled algorithm were compared in the study of the findings, both performance metrics were dramatically reduced.

References

[1] Jaimeel M Shah, Dr Sharnil Pandya, Dr Narayan Joshi, Dr Ketan Kotecha and Dr D.B.Choksi. "Load balancing in cloud computing: Methodological Survey on different types of algorithm." 2017 IEEE.

[2] B. Sosinsky(2010), "Cloud Computing Bible", Vol. 762, Indiana, Wiley Publishing, 2010.

[3] Gema Ramadhan, Tito Waluyo Purboyo, and Roswan Latuconsina (2018), "Experimental Model for Load Balancing in Cloud Computing using Throttled Algorithm",International Journal of Applied Engineering Research, Vol. 13, No. 2, pp. 1139–1143.

[4] BhathiyaWickrema singhe and Rajkumar Buyya (2021), "Cloudanalyst: A Cloudsim-Based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments",MEDC project report, Vol. 22, No. 6, pp. 433–659.

[5] BhathiyaWickrema singhe, Rodrigo N Calheiros, and Rajkumar Buyya (2022), "Cloudanalyst: ACloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications", IEEE international conference on advanced information networking and applications, pp. 446–452.

[6] Shridhar G Domanal and G Ram Mohana Reddy (2021), "Load Balancing in Cloud Computing using Modified Throttled Algorithm", IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1–5.

[7] Rajkumar Somani and Jyotsana Ojha (2022), "A Hybrid Approach for VM Load Balancing in Cloud using Cloudsim", International Journal of Science, Engineering and Technology Research, Vol. 3, No. 6, pp. 1734–1739.

[8] Monika Kushwaha and Saurabh Gupta (2015), "Response Time Reduction and Performance Analysis of Load Balancing Algorithms at Peak Hours in Cloud Computing", International Journal of Computer Applications, Vol. 128, No. 17, pp. 26-31.

[9] MousaElrotub and Abdel Gherbi (2018), “Virtual Machine Classification-Based Approach to Enhanced Workload Balancing for Cloud Computing Applications”, Procedia Computer Science, Vol. 130, pp. 683–688.

[10] Aditya Narayan Singh and Shiva Prakash (2018), “WAMLB: Weighted Active Monitoring Load Balancing in Cloud Computing”, Big Data Analytics, pp. 677–685.

[11] A. Makroo and D. Dahiya (2016), “Time Stamp Based Stateful Throttled VM Load Balancing Algorithm for The Cloud”, In. Journal of Science and Technology, Vol. 9, No. 48, pp. 1–8.

[12] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma (2021), “Performance Analysis of Load Balancing Algorithms”, World Academy of Science, Engineering and Technology, Vol 38, No. 3 pp. 269 – 272.

[13] KomalMahajan, AnsuyiaMakroo and Deepak Dahiya (2022), “Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure”, Journal of Information Processing systems, Vol. 9, No. 3, pp. 379 394.

[14] Mulat, Worku Wondimu, et al. "Improving Throttled Load Balancing Algorithm in Cloud Computing." Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2021. Singapore: Springer Nature Singapore, 2023.

[15] Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2023). A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques. Sensors, 22(3), 1242.

[16] Badshaha P Mulla, C. Rama Krishna and Raj Kumar Tickoo, “Load Balancing Algorithm for Efficient VM Allocation in Heterogeneous Cloud”, International Journal of Computer Networks & Communications (IJCNC) Vol.12, No.1, January 2023

[17] Jasobanta Laha, Sabyasachi Pattnaik and Kumar Surjeet Chaudhury, “Optimizing Resource Utilization in Cloud Environments: A Novel Dynamic Load Balancing Algorithm”, “Seybold Report Journal”, Vol. 18. No. 9. 2023.