

LOAD BALANCING ALGORITHM FOR EFFICIENT VM ALLOCATION IN HETEROGENEOUS CLOUD

Badshaha P Mulla¹, C. Rama Krishna² and Raj Kumar Tickoo³

^{1,2}Computer Science and Engineering Department, NITTTR, Chandigarh, India

³National Informatics Centre, Punjab State Unit, Chandigarh, India

ABSTRACT

Cloud computing is a growing service computing trend that offers users a range of on-demand variety of services from applications, processing capability, and storage based on the concept of the “Pay-As-Per-Use” model. Organizations from every sector are now realizing the benefits offered by cloud computing technology and moving towards the cloud. Cloud computing offers numerous advantages over conventional computing. However, it still faces a few challenges such as resource utilization in a cloud data centre and quality of service to the end-users due to improper workload balance among available resources. Heterogeneous cloud resources also impact cloud systems overall performance. We proposed an enhanced load balancing algorithm in this research paper for efficient VM allocation in a heterogeneous cloud. Our proposed algorithm allocates independent user tasks or requests to available virtual machines in cloud datacentre efficiently to manage proper load balancing. This algorithm is aimed at minimizing user request response time and the time required for data centre processing. The results obtained showed a significant reduction in user request response time and data centre processing time as compared to “Throttled” and “Round Robin (RR)” algorithms.

KEYWORDS

Load Balancing, Heterogeneous Cloud Environment, Response Time, Datacentre Processing Time, Independent Task.

1. INTRODUCTION

Over the past few years, the word cloud computing has received attention and enormous popularity in both academia and industry. The idea of cloud computing has its roots in the origins of utility computing.

The idea was coined and proposed publicly by John McCarthy in 1961 [1]:

“If computers of the kind I have advocated become the computers of the future, then computing may someday be organized as a public utility just as the telephone system is a public utility. ... The computer utility could become the basis of a new and important industry.”

The cloud computing definition composed by NIST has received industry-wide acceptance. The original definition is published in 2009, which is revised as per the industry inputs and published in 2011 [2]:

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

The term “cloud” in cloud computing refers to the two key concepts [3]:

Abstraction: The details regarding cloud system implementation are hidden from users and developers. The physical system which runs application is not specified and also the location where cloud data stored is unknown.

Virtualization: With virtualization, physical computing resources are represented and provided in the form of logical resources using hypervisors. Virtualization technology allows cloud service providers to reduce costs by pooling and sharing resources.

1.1. Motivation

Many organizations are attracted by the services and features offered by the cloud and started adopting it. The recent tremendous growth in the number of cloud users has resulted in heavy traffic at the cloud service provider. Dynamic nature of workload patterns and a huge number of service requests may lead to load imbalance on computing resources of cloud datacentre [4]. Due to improper load balance, the provider has to face the problem of performance degradation and resource wastage. Also, it degrades the overall Quality of Services (QoS) provided to the cloud users. The cloud datacentre processes the incoming user requests by allocating them to a Virtual Machine (VM) resource. The cloud datacentres may have a homogeneous or heterogeneous resource configuration. The heterogeneous VM configuration adds more difficulty to assign user requests and balancing the load among available VM's. Today load balancing has become a very complex task and considered as a key challenge in cloud computing [5-9].

1.2. Load Balancing

The load balancing mechanism dynamically distributes workload across all the computing nodes with proper workload balance on each node [10]. The situations like some computing resources are having a high workload while some are ideal or having little workload [11] are avoided using efficient load balancing mechanism. The workload examples in the cloud are memory-related load, processing load, storage load, network load, I/O load, etc. [12]. The cloud service provider aims at achieving maximum profit by providing the services, whereas the cloud users are expecting the high quality of service from the providers of cloud service. The expectations of both of these entities can be fulfilled by maximizing resource utilization through proper load balancing among the cloud resources.

The load balancing algorithm needs to consider the heterogeneous type of resources in cloud data centre along with its current state while deciding the allocation of user tasks to the resource.

1.3. Our Contributions

To overcome the challenges in a heterogeneous cloud environment and meet the good quality of service for cloud users, we propose an algorithm to manage workload in cloud heterogeneous datacentre efficiently. This paper's contributions are summarized in the following:

- A load-balancing algorithm is proposed for efficient VM allocation considering the heterogeneous configuration of virtual machines (VMs) in a cloud datacentre.
- Processing capacities of VMs are taken into consideration during the load balancing.
- The result obtained has demonstrated that the proposed algorithm reduced response as well as datacentre processing time significantly as compared to “Round Robin” and “Throttled” algorithm.

The paper is subdivided into five sections. Section 1 outlines the requirement of load balancing in a cloud. Section 2 presents some existing studies to this related work. Section 3 discusses our proposed algorithm. In Section 4, the simulation experiments with proposed and some traditional algorithm along with its results are analysed. The conclusions are given in Section 5.

2. EXISTING WORK

As cloud computing is gaining significant momentum, researchers are focusing on different key challenges in this field, and one of them is load balancing. In the following, some research studies related to load balancing are discussed.

Ramadhan et al. [13] have conducted experiments to determine the level of effectiveness of load balancing algorithms. The authors have pointed out that the incoming user requests for the cloud service are in vast numbers and also vary in nature of the job. The authors carried out simulation experiments using a “Throttled”, “Round Robin (RR)” and “Equally Spread Current Execution (ESEC)” algorithms [14-15]. The obtained result indicates that the Throttled algorithm has improved average response time. Domanal and Reddy [16] proposed “Modified Throttled Algorithm” for the distribution of incoming user tasks uniformly among the computing resources. The proposed algorithm maintains a table of VMs along with its state information similar to the Throttled algorithm. Initially, when the user request arrives, the algorithm selects VM from the starting index of VM index table but for the next subsequent request, the algorithm parses the table from the next VM which was previously assigned VM. In the existing “Throttled Algorithm”, the VM index table always scans from an initial index (starting index) whenever a new request arrives. The result shows improvement in response time for the users as compared with the “Throttled” and “Round Robin” Algorithm [14]. The processing capacity of each VM is not considered during allocation.

Phi et al. [17] have modified the Throttled Algorithm to minimize the cloud user response time and the datacentre processing time of cloud provider. This algorithm used two separate index tables to hold the VMs according to its status (Available Index & Busy Index). This algorithm provides more flexibility than the Throttled algorithm because it is easy to detect the VM in the size of the “Available Index” table only. The simulation results showed improvement in the processing time of datacentre and response time to the cloud user in comparison with “Round Robin” and “Throttled” Algorithm [14-15]. In this article, when the Available index table becomes empty, the load balancer returns a value of -1 to the Data Centre Controller (DCC). The DCC then arranges the request in the queue. This may increase the response time of such requests, and affect the overall response time.

Somani and Ojha [18] proposed a hybrid load balancing approach using “Throttled” and “Round Robin” Algorithm. The selected VM by the throttled algorithm is allocated in a round-robin manner. It effectively uses all the VMs selected by the throttled algorithm. This algorithm works well in a homogeneous cloud but performance may get degraded in case of heterogeneous cloud.

Kushwaha and Gupta [19] pointed out the problem of a heavy workload at the peak hours for accessing cloud services. This work has a primary focus on peak hour optimization problem. The performance analysis of “Throttled”, “Round Robin” and “Equally Spread Current Execution” algorithms [14-15] have done during the peak hours for internet application. The results showed that “Throttled” algorithm with “Performance Optimized Service Broker Policy” work best for handling peak hour workload. The authors have considered the homogeneous cloud for simulation, the heterogeneity of cloud resources may affect the performance in terms of response and processing time.

Elroub and Gherbi [20] discussed the classification technique for grouping of VMs and user tasks. The CPU and RAM utilization is considered for VM classification while the size of the user task and its associated information from log files are considered for task classification. Singh and Prakash [21] proposed enhancement over the “Active Monitoring Load Balancing (AMLB)” algorithm by assigning the weight to each virtual machine (VM) based on parameters like a no. of processors, the speed of the individual processor, RAM and network bandwidth. The primary aim of this work is to achieve effective VM utilization by allocating tasks to the VM based on the calculated weight factor. The weight factor calculation considers the only initial configuration of processing elements of VM. Makroo and Dahiya [22] proposed an efficient “Time Stamp Based Stateful Throttled VM Load Balancing Algorithm”. The state of userbase request allocation is preserved for determining the appropriate VM allocation to the userbase request in the future. The algorithm requires the extra overhead of preserving the state of userbase request allocation and may affect the overall performance.

In many previous research studies, the heterogeneous configuration of cloud datacentre resources and the actual utilization of processing elements have not been considered during the load balancing mechanism. The efficient load balancing algorithm needs to consider these issues for the effective utilization of all the available resources at cloud datacentre depending on their processing capabilities. The quality of services to the cloud users and datacentre processing time can be improved with these considerations. The Throttled Algorithm is considered as the best algorithm over the traditional load balancing algorithm in previous research studies.

This work focuses on all these issues and proposes an improvement over the throttled load balancing algorithm to achieve the improvements in response time and datacentre processing time in a heterogeneous cloud environment.

3. PROPOSED WORK

Our proposed work aims primarily at minimizing the response time of a user request and processing time in the heterogeneous cloud datacentre. The Datacentre Controller (DCC) consults with the proposed algorithm deployed at the VmLoadBalancer to identify the appropriate VM for the allocation of a user request. The performance analysis of the proposed work is compared with “Round Robin” and “Throttled” algorithms.

3.1. Traditional Algorithms

3.1.1. “Round Robin Algorithm (RR)”

This algorithm is a straight forward and simple algorithm widely used in cloud computing [23,24]. It assigns the users requests to each virtual machine in a circular fashion, without considering the processing capabilities of an individual virtual machine. This algorithm is highly efficient for datacentres where all the virtual machines have the same processing capabilities. In other words, it works well for the homogenous cloud environment.

3.1.2. “Throttled Algorithm”

The throttled algorithm is based on thresholding [14,17,25,26]. In this algorithm, a single VM can allocate only that many numbers of requests which is represented by its threshold value at any given time. In the case of more number of requests, the requests are queued until the state of VM becomes available. Due to this, it affects the overall user response time and datacentre processing time. For the heterogeneous cloud, the “Throttled” algorithm is not appropriate.

3.1.3. Our Proposed Algorithm

Our proposed algorithm considers each VMs actual processing power during the allocation. In this proposed algorithm, the indexes of VM are stored in two separate tables i.e. AVAILABLE Table and BUSY Table.

On receiving the user request, the AVAILABLE table will be scanned to find out the VM. In case of no VM found in the AVAILABLE table, the algorithm will search the VM with the suitable capacity to execute the current request from the BUSY table [17]. The defined threshold levels are used to avoid the extra overload on the BUSY VM.

In a heterogeneous cloud environment, the available VM resources may have different capabilities in terms of their processing speed, number of processors and memory. So, instead of putting a request in the waiting queue due to VM not being available; it can be executed on VM with sufficient resources from the BUSY table.

The working of the proposed algorithm is represented in Figure 1 and steps are described in Algorithm 1.

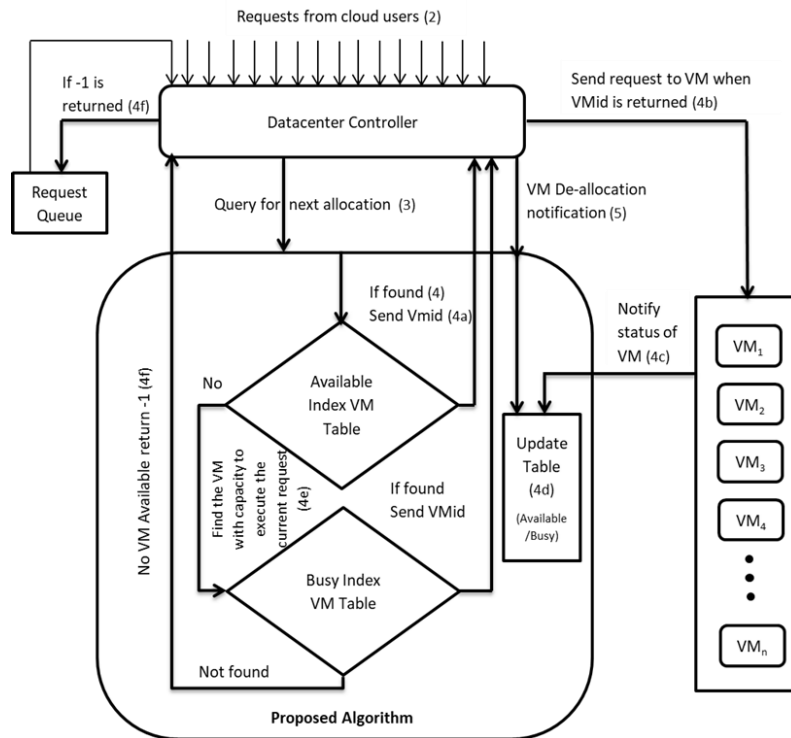


Figure 1. Proposed Load Balancing Algorithm for Efficient VM Allocation in Heterogeneous Cloud

In order to search for a suitable VM from the BUSY table, our proposed algorithm calculates the capacity of each VM. It also calculates the average capacity and selects only that VM which is having its capacity greater than or equal to the average capacity. The requests are allocated to these VMs in a round-robin manner. The algorithm uses two different threshold values to avoid extra overload on BUSY VM:

Threshold C (Cloudlets): This value represents the limit of maximum running cloudlets on each selected VM. It helps to avoid overload on the selected VM from BUSY VM.

Algorithm 1: Proposed VM Allocation based on their Processing Power

Input: Cloud User Requests, Available Virtual Machines (VMs)

Output: Virtual Machine Index Identifier

Step 1: Initialize

Two index tables are used to hold indices of AVAILABLE VMs and BUSY VMs

AVAILABLE: Holds indices of all available VMs

BUSY: Empty

Step 2: New cloud service request is arrived at the Datacentre Controller (DCC)

Step 3: DCC queries to the proposed algorithm for current request allocation on VM

Step 4: Algorithm scans the AVAILABLE index table of VMs :

If VM found then

a. The VM Index id is returned to a DCC.

b. The DCC forwards the current request to respective VM as per the received VM id.

c. The DCC informs this new allocation to the algorithm for updates in VM state.

d. Algorithm updates the status of VM and lists accordingly.

else

e. The algorithm parses the "BUSY" list of VMs and finds the VM with a capacity to execute the current request.

Find Efficient VM from BUSY Index Table (Algorithm 2)

If found repeat Step **a** to **d**.

f. Returns negative value (-1) to the DCC. On receipts of such negative value, DCC puts the current request in request queue.

Step 5: After VM completes processing the request, it sends the response to DCC, and DCC further informs the algorithm of VM deallocation.

Step 6: The DCC checks for any pending requests available in the request queue. If found, it will continue from step number 3.

Step 7: Repeat from Step 2 as and when a new request arrives.

Threshold B (Burst): This value represents a maximum number of requests handled for allocation on BUSY VM. This allows a particular number of the request to be processed based on the value of this threshold. After reaching this maximum limit of threshold, the algorithm does not allocate or process the subsequent request. For such subsequent request, it returns -1, and notifies the data centre to queue up the request, it also starts decrementing the threshold value. On reaching zero value, the algorithm resets the threshold value to its original value. It simply provides a way to avoid a burst of requests and helps in avoiding overload on the BUSY VM. The sub-algorithm for finding efficient VM from the BUSY list is described in Algorithm 2.

Our proposed algorithm helps to identify the appropriate VM from the BUSY index table to allocate the request when the AVAILABLE index table is empty. This helps in the reduction of user response time and also the datacentre processing time.

Algorithm 2: Proposed method to find Efficient VM from BUSY Index Table

Input: BUSY Virtual Machines (VMs) List

Output: Virtual Machine Index Identifier

Step 1: Calculate each VM's current capacity present in BUSY Index Table.

Step 2: Calculate the average capacity all BUSY VMs.

Step 3: Select VM in Round Robin manner for processing the current request

If Capacity of VM \leq Average Capacity **AND**

Cloudlets on VM \geq Threshold C

then

Return Index of VM to DCC.

else

Return -1 to DCC.

Step 4:

if Threshold B \geq Max Limit **then**

Repeat **Step 3** for the subsequent request

else

Put the request in waiting queue and return -1 to DCC.

Decrement the Threshold B by one unit.

Step 5: If Threshold becomes ZERO, reset to its original value and then continue from Step 3.

4. SIMULATION SETUP AND RESULT DISCUSSION

We have used the simulator to perform the experiments because real tests limit experiments on the scale of the infrastructure. It also makes the repetition of experiments very difficult [27]. The performance measurement of the system using of real cloud environment is a time-consuming process. In addition, payments are required to access the infrastructure of a real cloud [28].

4.1. Simulation Setup

For the simulation, we have considered Facebook (FB) a Cloud-based social networking application on the Internet. As per the report published in [29] the approximate Facebook user's distribution across the main six regions (continents) is given in Table 1.

Table 1. Facebook Users Distribution across the regions (Approx.)

Region with Cloud Analyst Region Id	Users (Approx in Millions)
North America(0)	260
South America(1)	270
Europe(2)	340
Asia(3)	810
Africa(4)	170
Ocenia(5)	20

We have assumed a similar system but on the normalized scale (1/200)th. We have defined 6 User Bases which represent the users from the above 6 regions. The user base characteristics are represented in Table 2.

Table 2. User Base Characteristics.

User Base and Region	Time Zone (GMT)	Peak Hours	Average Peak Users	Average Off-Peak Users
UB-0	GMT - 6.00	13.00 – 15.00	130000	13000
UB-1	GMT - 4.00	15.00 – 17.00	135000	13500
UB-2	GMT + 1.00	20.00 – 22.00	170000	17000
UB-3	GMT + 6.00	01.00 – 03.00	400000	40000
UB-4	GMT + 2.00	21.00 – 23.00	85000	8500
UB-5	GMT + 10.00	09.00 – 11.00	10000	1000

For simplicity, we have assumed only 10% of users are active during off-peak hours. Also, we have assumed that when a user is online it generates a new request for every 5 minutes. The peak hours are assumed by considering that, many users generally access the application in the evening time after the work for around 2 hours [30-31]. Other parameters used are given in Table 3.

The virtual machines (VMs) are hosted on physical machines available in a datacentre. For creating a heterogeneous cloud environment, the physical machines are having different capacity of RAM (2GB/4GB), number of CPU (2/4/6/8) and processing power in terms of MIPS.

Table 3. Other Parameters

Parameter	Value
Virtual Machine (VM)	
Size of VM Image	10000 byte
Memory Size	512 MB
Bandwidth	1000 byte
Architecture	x86
Physical Machine at Datacentre	
Operating System	“Linux”
VMM/Hypervisor	“Xen”
Main Memory	2048 / 4096 MB
VM Scheduling Policy	“Time Shared”

4.2. Results and Discussion

The three different scenarios are considered for analyzing the behaviour of our proposed algorithm. The number of datacentres (DCs), VMs and broker policy are different for each scenario. The simulation is repeated for “Round Robin”, “Throttled” and our proposed algorithm for analyzing the results. The performance parameters are:

i. Average Response Time

“It is the time taken by the system to respond to the user request.” It is desirable to respond more quickly to provide acceptable quality of services to the cloud users [4].

$$\text{Response Time} = \text{Finish}_t - \text{Arrival}_t + \text{Transmission}_{\text{Delay}}$$

Where,

Finish_t = Completion time of request processing

Arrival_t = Time of request arrival

Transmission_{Delay} = Network Delay between source and destination for providing the response to the user

ii. Average Datacentre Processing Time

“It is the time taken by the datacentre to process all the received requests from the userbases.” It is calculated using following formula [4],

$$\text{Processing Time} = L_{\text{VMs}} / C_{\text{VMs}}$$

Where,

L_{VMs} = Sum of each VM load in a datacentre

C_{VMs} = Sum of capacity of each VM in a datacentre

Case 1: Single Datacentre with 50 VMs

In this scenario, only a single datacentre is used and hosted with 50 VMs. Figure 2 shows that the “Round Robin (RR)” and “Throttled” algorithm takes more time for the considered performance parameters. In the “Round Robin” algorithm, the VMs are allocated in a circular manner without considering its processing capabilities such as processing power, number of processor, RAM etc.

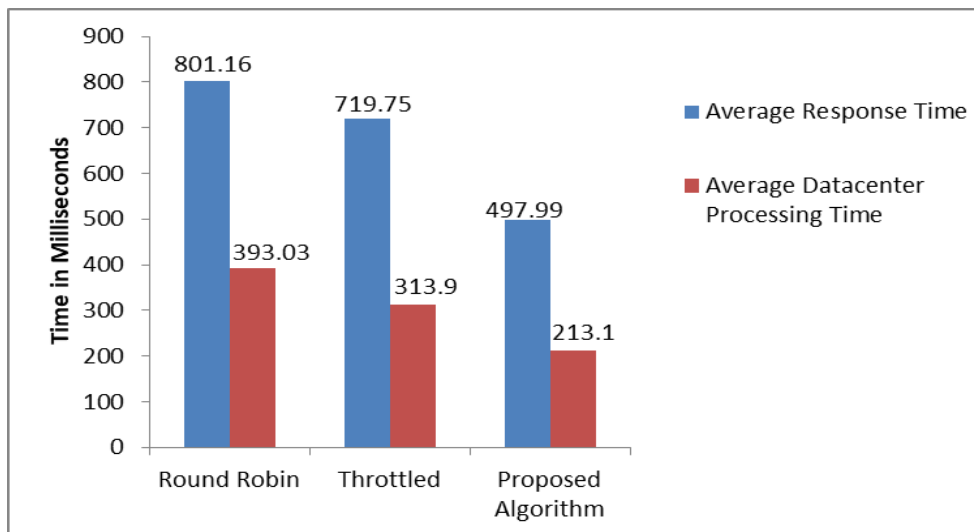


Figure 2. One Datacentre with 50 VMs

In the “Throttled” algorithm, a single VM can allocate only a specified number (represented by threshold value) of user requests at any given time. In case of additional requests, such requests

are queued in waiting queue until the availability of next VM. Throttled algorithm also does not consider the processing capability of VMs. Our proposed algorithm has considered the capabilities of each VM and allocated the requests accordingly. The result of our proposed algorithm showed a 30% reduction in user request response time and 32% reduction in processing time of datacentre.

Case 2: Two Datacentre with 25 VMs

This case represents the scenario where application grows in popularity. The provider deploys its application in few more locations. Assuming this, we have used two datacentres with 25 VMs each in this case. Also, we used the Closest Datacentre (“Service Proximity Based”) and “Performance Optimized Routing” service broker policies. Figure 3 and 4 shows simulation results.

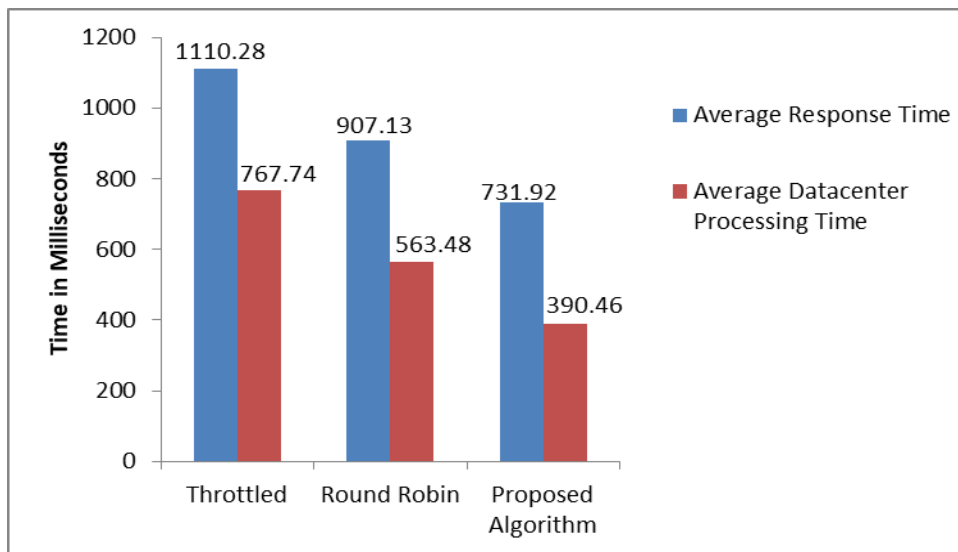


Figure 3. Two Datacentres with 25 VMs and Closest Datacentre Service Broker Policy

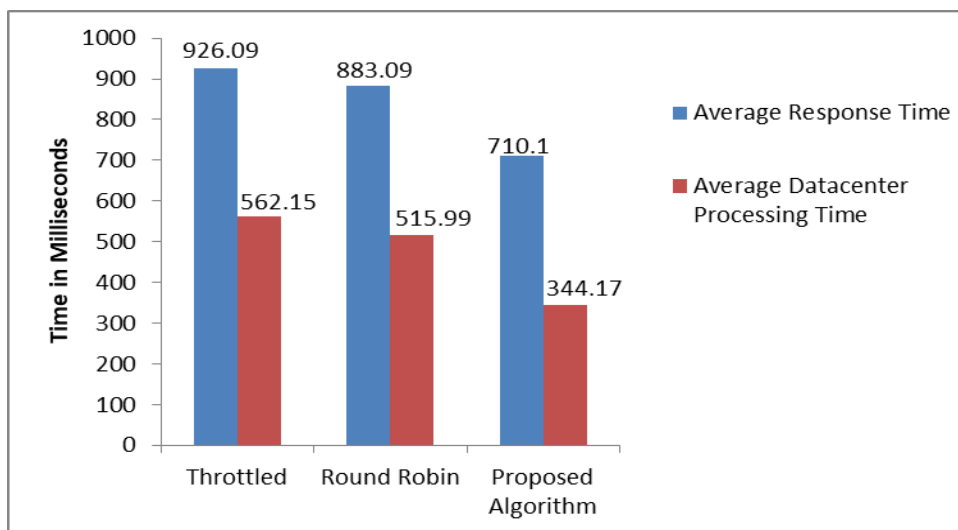


Figure 4. Two Datacentres with 25 VMs and “Performance Optimized Routing Service Broker Policy”

From Figure 3 and 4, it is clear that the “Performance Optimized Routing” service broker policy provides better results. The reason behind this is the closest datacentre service broker policy selects the nearest datacentre without considering the response time whereas the performance optimized service broker policy calculates an estimation of response time from each datacentre and selects the best datacentre. In both cases, our proposed algorithm has reduced response time significantly by 20% and datacentre processing time by 30%.

Case 3: Two Datacentres with 50 VMs

In this scenario, the VMs in a datacentre are increased to analyze the effect on response and datacentre processing time. In Case 2 each datacentre hosts 25 VMs; here we have two datacentres with each 50 VMs. We used the “Performance Optimized Routing” service broker policy. The results are shown in Figure 5.

The result showed an improvement over Case 2 in terms of both performance parameters due to the efficient sharing of peak hour load. In Case 2, we have used 25 VMs at each datacentre and they get overloaded in peak hours. By increasing the VMs the peak hour load is efficiently shared among the VMs. The proposed algorithm is performing better than the “RoundRobin” and “Throttled” Algorithm. The response time is reduced considerably by 28% and datacentre processing time is reduced by 44%.

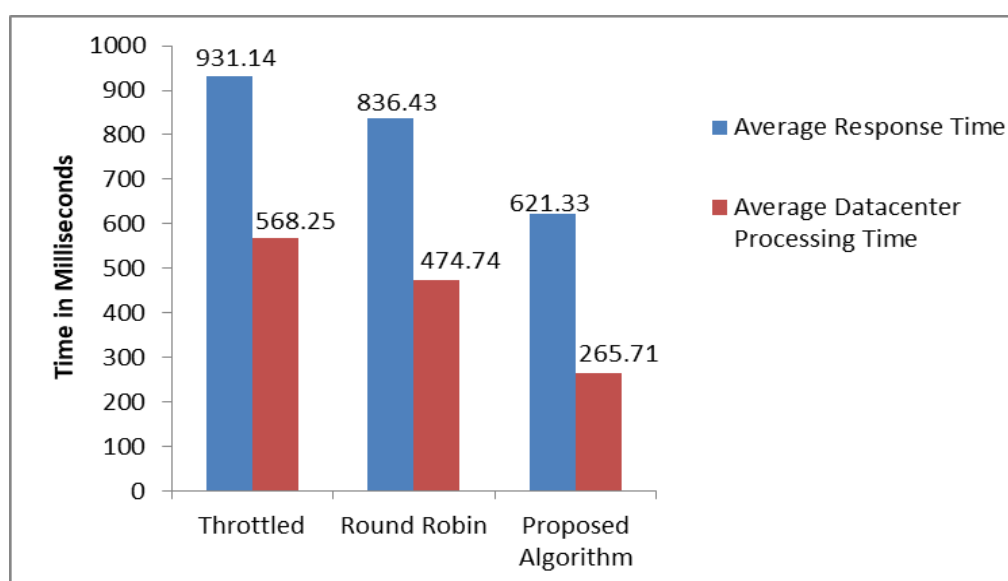


Figure 5. Two Data Centres each with 50 VMs.

For each of these 3 simulation cases, the datacentre consists of machines with the varying processing power in terms of MIPS (500/1000/1500/2000/4000/5000), varying numbers of CPU (2/4/6/8) and RAM (2GB/4GB/8GB). The virtual machines (VMs) are hosted on physical machines available in a datacentre.

5. CONCLUSIONS

Cloud computing’s growing importance has challenged to find effective ways to improve cloud services. The proper load balancing can enhance cloud services. Many algorithms have been proposed for load balancing in literature however, they have not considered the heterogeneous configuration of resources at cloud datacentre. Also, the current status of processing elements of

resources during its use has not been considered. This work considered these issues and proposed an algorithm for efficient VM allocation in the heterogeneous cloud. The primary objective of the proposed work is to reduce response time and datacentre processing time. The proposed algorithm calculates the current capacity of each VM and an average capacity of all VMs. Based upon these capacities and pre-defined threshold values for avoiding overload, the algorithm allocates the requests to a suitable VM. The analysis of results showed significantly reduction in both considered performance parameters compared to “Round Robin (RR)” and “Throttled” algorithm.

REFERENCES

- [1] R. P. Erl, Thomas, and Zaigham Mahmood (2013), “Cloud Computing Concepts, Technology & Architecture”, Prentice Hall.
- [2] P. Mell and T. Grance (2011), “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology”, Nist Special Publication.800-145, pp. 1-7.
- [3] B. Sosinsky(2010), “Cloud Computing Bible”, Vol. 762, Indiana, Wiley Publishing, 2010.
- [4] A. Thakur and M. S. Goraya (2017), “A Taxonomic Survey on Load Balancing in Cloud”, Journal of Network and Computer Applications, Vol. 98, pp. 43–57.
- [5] M. Mesbahi and A. Masoud Rahmani (2016), “Load Balancing in Cloud Computing: A State of the Art Survey”, International. Journal of Modern Education and Computer Science, Vol. 8, No. 3, pp. 64-78.
- [6] J. Duan and Y. Yang (2017), “A Load Balancing and Multi-Tenancy Oriented Data Center Virtualization Framework”, IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 8, pp. 2131–2144.
- [7] D Chitra Devi and V Rhymend Uthariaraj (2016), “Load Balancing in Cloud Computing Environment using Improved Weighted Round Robin Algorithm for Non preemptive Dependent Tasks”, The scientific world journal, Vol. 2016, pp. 1-14.
- [8] EinollahJafarnejadGhomi, Amir MasoudRahmani, and Nooruldeen Nasih Qader (2017), “Load-balancing Algorithms in Cloud Computing: A Survey”, Journal of Network and Computer Applications, Vol. 88, pp. 50–71.
- [9] Mala Kalra, and Singh S (2015), “A Review of Metaheuristic Scheduling Techniques in Cloud Computing”, Egyptian Informatics Journal, Vol. 16, No.3, pp. 275–295.
- [10] Minxian Xu, Wenhong Tian, and Rajkumar Buyya (2017), “A Survey on Load Balancing Algorithms for Virtual Machines Placement in Cloud Computing”, Concurrency and Computation: Practice and Experience, Vol. 29, No. 12, pp. 1–16.
- [11] MA Alsaih, R Latip, A Abdullah, and SK Subramaniam (2013), “A Taxonomy of Load Balancing Techniques in Cloud Computing”, International Review on Computers &Software, Vol. 8, No. 1, pp. 64–76.
- [12] Ali M Alakeel (2010), “A Guide to Dynamic Load Balancing in Distributed Computer Systems”,International Journal of Computer Science and Information Security, Vol. 10, No. 6, pp. 153–160.
- [13] 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.
- [14] BhatiyaWickrema singhe and Rajkumar Buyya (2009), “Cloudbanalyst: A Cloudsim-Based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments”,MEDC project report, Vol. 22, No. 6, pp. 433–659.
- [15] BhatiyaWickremasinghe, Rodrigo N Calheiros, and Rajkumar Buyya (2010), “Cloudbanalyst:ACloudsim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications”, IEEE international conference on advanced information networking and applications, pp. 446–452.
- [16] Shridhar G Domanal and G Ram Mohana Reddy (2013), “Load Balancing in Cloud Computing using Modified Throttled Algorithm”, IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1–5.
- [17] N.X Phi, C.T. Tin, K Thu, and Hung T.C. (2018), “Proposed Load Balancing Algorithm to Reduce Response Time and Processing Time on Cloud Computing”, International Journal of Computer Networks and Communication, Vol. 10, No. 3, pp. 87–98.

- [18] Rajkumar Somani and Jyotsana Ojha (2014), “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.
- [19] 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.
- [20] 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.
- [21] Aditya Narayan Singh and Shiva Prakash (2018), “WAMLB: Weighted Active Monitoring Load Balancing in Cloud Computing”, Big Data Analytics, pp. 677–685.
- [22] 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.
- [23] Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma (2008), “Performance Analysis of Load Balancing Algorithms”, World Academy of Science, Engineering and Technology, Vol. 38, No. 3 pp. 269–272.
- [24] KomalMahajan, AnsuyiaMakroo, and Deepak Dahiya (2013), “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.
- [25] VibhoreTyagi and Tarun Kumar (2015), “ORT Broker Policy: Reduce Cost and Response Time using Throttled Load Balancing Algorithm”, Procedia Computer Science, Vol. 48, pp. 217–221.
- [26] Mohammad Reza Mesbahi, Mahnaz Hashemi, and Amir Masoud Rahmani (2016), “Performance Evaluation and Analysis of Load Balancing Algorithms in Cloud Computing Environments”, Second International Conference on Web Research (ICWR), pp. 145–151.
- [27] AtyafDhari and Khaldun I Arif (2017), “An Efficient Load Balancing Scheme for Cloud Computing”, Indian Journal of Science and Technology, Vol. 10, No. 11, pp. 1–8.
- [28] Keith R Jackson, LavanyaRamakrishnan, Krishna Muriki, Shane Canon, ShreyasCholia, John Shalf, Harvey J Wasserman, and Nicholas J Wright (2010), “Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud”, IEEE second international conference on cloud computing technology and science, pp. 159–168.
- [29] Internet World Stats, “Facebook Stats”, 2017. [Online]. Available: <http://www.internetworldstats.com>.
- [30] Evan Asano, “How much time do people spend on social media?”, 2017. [Online]. Available: <https://www.socialmediatoday.com/marketing/how-much-time-do-people-spend-social-media-infographic>.
- [31] Elizabeth Arens, “Best times to post on social media for 2019”, 2019.[Online]. Available<https://sproutsocial.com/insights/best-times-to-post-on-social-media>.

AUTHORS

Badshaha P. Mulla is perusing M.E. (Computer Science and Engineering) from the Department of Computer Science and Engineering (CSE) at National Institute of Technical Teachers Training and Research (NITTTR), which is funded by Ministry of Human Resource Development (MHRD), the Government of India, and affiliated to Panjab University, Chandigarh. He received the B.E. (Computer Science and Engineering) from Shivaji University, Kolhapur (Maharashtra) in India, 2007. He has more than 7 years of teaching experience. His main research areas are load balancing, performance optimization in cloud computing, and computer networking.



C. Rama Krishna received his B.Tech from JNTU, Hyderabad, M.Tech from Cochin University of Science & Technology, Cochin, and Ph.D. from IIT, Kharagpur. He is Senior Member, IEEE, USA. Since 1996, he is working with the Department of Computer Science and Engineering, National Institute of Technical Teachers Training and Research (NITTTR), Chandigarh, and currently holding the position of Professor & Head. His areas of research interest include Computer Networks, Wireless Networks, Cryptography & Cyber Security, and Cloud Computing. To his credit, he has more than 80 research publications in referred International and National Journals and Conferences. He acted as Associate Editor for International Journal of Technology, Knowledge and Society. He is a member of Advisory/Technical committees for many national and international journals and conferences and also chaired many technical sessions. He is a reviewer of various journals for the IEEE, ACM, Elsevier, and Springer publishers. He has more than 22 years of experience in organizing more than 100 training programs in the upcoming areas of CSE and IT for the faculty of engineering colleges, polytechnics and industry professionals. He is an expert member of National Board of Accreditation (NBA) for CSE and IT disciplines and NIRF, Govt. of India.



Er. Raj Kumar Tickoo M.E. (Computer Science and Engineering) has been working as Sr. Technical Director (Scientist 'F') in National Informatics Centre, Ministry of Electronics & Information Technology, Govt. of India. He has worked as Oracle DBA for seven years and has experience in implementation and execution of large projects of various ministries. He is interested in the latest technologies like AI, IOT, and Data Analytics.

