International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Proposed Hybrid Load Balancing Algorithm to Reduce Response Time and Processing Time on Cloud Computing

Sarita Yogi[1], Ritesh Yadav[2]
[12]Department of Computer Application
SRK University, Bhopal, Madhya Pradesh, India

Abstract: A general name for the Internet-based hosting of services is "cloud computing." With cloud computing, scheduling seems to be a method for reducing the overall task response and processing time. Additionally, load balancing might benefit from a strong scheduling mechanism. In this work, we present a hybrid algorithm which attempts to enhance the system's typical response time with average processing time inside a cloud environment. The results demonstrate that the suggested approach gives better outcomes than the established algorithm currently in use. This study will move the field closer to more in-depth investigation by optimizing the current methods.

Keywords: Algorithm, Load Balancing, Processing Time, Response Time, Round-Robin, Scheduling.

INTRODUCTION

The fastest-evolving technology within the IT sector, cloud computing offers a new way to supply services that are paid for only as they are utilized. The supply of application software, framework, with cloud infrastructure through the Internet that is available from a search engine, along with software's and data that are stored mostly on servers for a pay-per-use model, are all examples of common conceptions of cloud computing. Internet-based services are provided via cloud computing. A combination of various computer resource systems and services known as "cloud computing" could be quickly and simply established and made accessible via the internet. The nature of cloud computing is scattered. The cloud does have a variety of features, including speed, security, as well as privacy.

Cloud computing has emerged as a famous technology that has been embraced by both business and academia, offering a versatile and effective way to store and retrieve materials[1]. The key challenge is scheduling incoming requests so that there is a minimal response time, efficient resource use, while also ensuring time, no underutilization of resource. Virtualization, a technology that increases power efficiency of datacentre and allows virtual machines to run onto a single server, is a key component of cloud computing systems.

An application run solely on a single or more linked servers instead of a localized computing device, like a PC's, tablets, or smartphones, inside the cloud computing models of network computing. One user can connect with a server to carry out a task, similar to the Conservative client-servers approach. The distinction of cloud computing being that it makes use of the virtualization concepts to allow the computing operation to execute on multiple linked devices. With virtualization, 1 or many physical servers could be set up and divided into numerous "virtual servers" that are not connected to each other but nevertheless appear to the customer to be a unified physical device[2].

A kind of dispersed computing known as "cloud computing" concentrates on providing many different users dispersed accessibility to virtualized hardware's and software architecture the internet. In the virtualization of dispersed computing, networking, internet services, as well as software are all involved. The concept of cloud computing has today's consumers interested in parallel, remote, and virtualized computing systems. It seems to be a common way to offer affordable and convenient access to shared IT resources[3].

Load Balancing: Load balancing is really the process of enhancing the performance of a paralleled and distributed systems, by redistributing the load across the processors. By minimizing execution time, decreasing communication delays, maximizing resource utilization, and maximizing throughput, load balancing aims to distribute the workload evenly amongst nodes.

Websites serve as a simple illustration of load balancing within regular living. Users could encounter delays, latencies, and even lengthy system replies without load balancing. In order to effectively distribute communication traffic and ensure website uptime, load balancing technologies frequently use redundant servers[4].

Resource sharing is indeed a highly desirable feature inside distributed computer systems environment, as defined in,

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

when two or more independent computers are linked by a communications system. A dispersed system's nodes can increase system efficiency in addition to sharing data as well as I/O devices by pooling their computing resources. Inside the distributed system, a process called load balancing allows jobs to migrate from 1 computer to the next. This results in quicker job service, such as reduced job response times and improved resource use. Numerous studies have demonstrated that load balancing amongst distributed systems node significantly boosts system performance as well as maximizes resource usage.

Problems Which Occur in Load Balancing and Scheduling- Due to the lack of centralized authority to divide the workload amongst numerous processors, the load balancing technique within distributed systems faces significant difficulties[5]. Following is a description of a few of the problems:

1. A solid load balancing strategy must be all-encompassing, reliable, scalable, as well as add a minimal amount of burden to system. These variables are linked.
2. Since processes may switch between one node towards another even while in the midst of execution to guarantee an equal workload, load balancing is essential.
3. Determining how to manage the load distribution amongst processors to ensure that the calculation is finished in the smallest amount of time is indeed a significant problem.
4. In order to avoid processes from becoming repeatedly circulated throughout the system without advancement, load balancing algorithms must use the assumption that data currently available at every node is valid.
5. When processes compete for processing time at different processors, load sharing tries to keep idle processors from entering the unshared mode.
6. Load balance is among the key components of scheduling issue. The necessity for equality and the need for location of the data frequently clash, which presents a difficulty for scheduling algorithms.
7. Since distributed operating processes are non-uniform as well as non-preemptive, meaning that processors might differ, load balancing including task scheduling are crucial components of total system performance.

Load Balancing Basics: Given that we have a common language, let's look at the fundamental load balancing operation. The load balancer is frequently placed among the user and the hosts which offer the services which customer wants to utilize, as shown in the image. This isn't a requirement with load balancing, as is the case with most situations, but rather a best practice in a normal deployment[6], [7]. Additionally, let's suppose that load balancer has been settled set up with one virtual server pointing to a network with 2 service points.

The basic transaction of load balancing is as follows-

- The client makes an attempt to connect to the load-balancing service.
- After deciding which host will receive the link, the load balancer obtains the connection then modifies the target IP (and perhaps port) to correspond with the services of the chosen host (note which the source IP of the customer is not touched).
- The host receives the connection then replies to the customer, the source of the request, using the load balancer as its default gateway.
- The load balancer detects the host's returning packet, modifies the source IP (as well as conceivably the port) to meet the IP with port of the virtual server, and then passes the message back to the customer.
- Once the client has received the return package and is convinced that it was sent by the virtual server, the procedure is continued.

Although this extremely straightforward example is quite up forward, there seem to be a few important details to pay attention to. Firstly, the client believes that communicating with the virtual server is as simple as sending packets there and waiting for a response. The NAT happens second. Now, the load balancer switches out the virtual server's target IP for the target IP of the hosts that it has decided to load balance the call. The final step of this procedure, which turns the NAT "bi-directional," is phase three. The client might be getting a packet by someone he didn't order one from and might simply drop it when the source IP of such return package from the hosts was left unchanged and the package was simply sent to the consumer. However, the load balancer fixes the packet by rewriting it with the virtual server's IP as the source IP while keeping track of the link.

Load Balancing Algorithm: Because to the independence of the processing units and also the inter-processor communication cost involved in the gathering of state data, communication delays, redistribution of workload, etc., load

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

balancing on several computers remains a challenge. The best situation for resolving and operating distributed and paralleled programme applications is one that utilises paralleled and distributed computing. A huge process or job is split up and then spread over several hosts enabling parallel computation in these types of applications. Inside a system with several hosts, Livny and Melman have noted that the likelihood of one host sitting idle while another host has numerous jobs stacked up could be extremely significant[8].

In this case, load balancing is most likely to enhance efficiency. These load imbalances imply that either work transfers from the overladen hosts to the lighter laden hosts or load distribution evenly/fairly amongst hosts can enhance efficiency. The aforementioned objective is accomplished with the aid of the load balancing algorithms(s).

The types and quantity of load and work information presumed to be acquired by the decision-making components has a direct bearing onto the load balancing algorithms used. Based on how the process is initiated, load balancing algorithms could be divided into three groups as follows:

a) Sender Initiated- In this kind, the sender initializes the load balancing algorithms. This kind of technique involves sending request messages to potential recipients until one is found that can take on the load.
b) Symmetric- It combines both sender- as well as receiver-initiated behaviors.
c) Receiver Initiated- In this kind, the receiver initiates the load balancing process. To discover a sender who can accept the loads, the receiver transmits request messages throughout this type of descriptive algorithm.

Static Load Balancers:

1. Round-Robin load Balancer
2. Min-Min
3. Max-Min

Dynamic Load Balancers:

1. Throttled load balancer
2. Honeybee foraging Algorithm
3. Biased random sampling
4. Round-robin technique
5. SJF technique
6. Active clustering
7. Join-idle queue
8. GP algorithm
9. Equally Spread Current Execution (ESCE) Algorithm

LITERATURE REVIEW

In their analysis of cloud computing's infrastructure, Jadeja, Y., and K. Modi also discuss a few of its most important uses as well as advantages and problems related to safety and privacy. Jing Yao and Ju-hou He talk on the structure of cloud computing, which is split into 2 parts the front end and the back end[9]. The web links the two together. Users can only see the front end, while the back end is for the cloud infrastructure. The customer's computer is really the "front end," and the back end provides the "cloud computing services," such as storage, machines, and so on. It also examines the Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) layers and services offered by the clouds computing architecture, as well as certain concerns with privacy, security, as well as reliability, among other things.

For a description and classification of load balancing, see Khiyaita et al[10]. Researchers discussed the various load balancing techniques utilised in the most popular distributed applications. The principal difficulties with load balancing within cloud computing were also addressed. An extensive analysis of the elements supporting cloud computing is given by Preeti Mishra et al[11], who also reviews several cloud deployments with service models. It lists the difficulties of cloud computing as being security, confidentiality, internet dependency, plus accessibility. The adaptability, increased resource utilisation, lower capital, and scale of cloud computing beyond conventional IT service environments are examined by the researcher as causes to transition towards a cloud computing environment. It views the technological difficulty of vertical scaling in cloud computing.

Bhathiya, Wickremasinghe had also documented in detail how the GUI-depended Cloud Analyst tool works[12]. This tool was created to model larger-scale Cloud apps so that researchers could examine how they behave in different deployment scenarios. Developers utilize Cloud Analyst to better understand how to distribute applications across Cloud infrastructures as well as employ value-added solutions like Service Brokers-enabled efficiency optimization of applications and inbound providers.

S. Mohapatra et al. addressed a comparison of the performance for several virtual machine plus policy load balancing strategies[13]. The effectiveness of 4 well-recognized load balancing algorithms, including First Comes First Service, Execution Loads, Round Robin, & Throttled

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Loads Balancing Algorithms, has been examined in this research cantered on the typical response time, typical datacentre demand servicing time, as well as overall cost. The Clouds Analyst simulator's findings indicate that round robin offers the greatest integration performance.

Various scheduling strategies are covered by Isam Azawi Mohialdeen[14]. The author compares scheduling algorithms used in cloud computing also explains how they are used in this setting. The round robin algorithm alternative developed by Dash et al increases CPU effectiveness in real-time as well as time-sharing operating systems[15]. The writer's proposed technique eliminates all the drawbacks of a straightforward round robin structure. Additionally, he compared the suggested technique to the standard round robin's scheduling algorithm. The suggested approach lowers the performance factors to a suitable level, increasing systems throughput and resolving the issue with easy round robin design.

A comparison of the Round Robin and Throttled virtual machine load balancing methods has been presented by V. Behal and A. Kumar[16]. In order to calculate cumulative response time, datacentre's hourly estimated completion times, response times as per zone, datacentres demand servicing times, consumer base every hour response times, and full expenses, that has a major impact on performance, all these algorithms are utilized with optimised response times service broker strategy and simulation is executed. The simulation findings show that in a diverse cloud computing environment, the suggested technique of throttled and optimised response time services broker policy performs better over round robin load balancing method.

The performance of 3 load balancing algorithms was discussed sarkar et al[17], who also looked into their limitations and investigated the reasons why it is impossible to have centralised scheduling policy under cloud conditions. The Honeybee Foraging Behaviours algorithm, Randomized Sampling method, and Active Clustering algorithm offered for load balancing were examined by the author as three potential options.

In order to load balance nodes, K Nishant et al devised an algorithm which is a modified version of ant colonies optimization[18]. This technique has been implemented from the perspective of cloud network systems. It differs from the original strategy in which every ant first constructs its own result set before eventually constructing an entire result. The ants, on the other hand, modify a single result collection as opposed to each one of their own result sets. This method identifies nodes that are overloaded as well as underloaded and then executes operations depended on the discovered nodes. Every ant's job is specific rather than all-encompassing, and it relies on the kind of initial node it found and if it was over-loaded or underloaded.

A study on load balancing algorithms in cloud computing was put up by Ghomi et al[19]. The study classified task scheduling as well as load balancing algorithms into seven categories, including hadoop-map decrease load balancing, agents-based load balancing, natural phenomenon-based load balancing, applications-oriented load balancing, generalized load balancing, networks aware load balancing, as well as workflow-depended load balancing, that belong to the writings fall under 2 domains depended on system condition and who initialised the procedure. The various algorithms out of each class are gathered together, and both their benefits and drawbacks are listed.

In the meantime, Milani et al examined the load balancing methods that had been identified through the study[20]. The authors divided the available algorithms into three primary categories: static, dynamic, as well as hybrid. The writers addressed major concerns concerning importance, expectations degree of metrics, responsibility, and obstacles experienced in load balancing. They also formalised pertinent questions regarding load balancing. Also, with aid of Boolean processes in search terms and the use of a Quality Assessment Checklists (QAC) as the selection criterion, a good search procedure was carried out in the search term to extract the most pertinent content from various publishing sources.

The Response time, Build, Scalability, Resource Usage, Migration Time, Throughput, and Energy Conservation were the only QoS metrics which were investigated in the 2 studies, leaving out other crucial QoS metrics such as migration expense, service level infringements, extent of balance, work rejection ratio, etc. This survey closes the gap in metric choices for analysis.

The Ant Colony Optimizations (ACO), Genetic Algorithm (GA), Particle Swarm Optimizations (PSO), League Championship Algorithms (LCA), and BAT algorithms were 5 basic meta-heuristic methodologies that Kalra and Singh took into consideration when comparing different scheduling algorithms for cloud as well as grid computing[21]. The methods are also thoroughly compared, although their research is restricted to metaheuristic method scheduling algorithms exclusively. Additionally, the survey focuses exclusively on evolutionary algorithms and lacked comprehensive classification.

Mesbahi and Rahmani analysed the fundamental needs and elements in developing and deploying a preferred load

balancer on cloud provider as well as divided load balancing algorithms under 3 groups: general algorithms based, architecture based, and artificial intelligence depended[22]. This research, like other earlier studies, views static with dynamic categorisation as a broad arrangement. However, writers identified several significant difficulties in developing load balancing algorithms. Additionally, writers formed a conclusion based on research that the ideal algorithms are those that are dynamic, distributed, and non-cooperative.

According to a categorization paper by Kanakala et al, load balancing methods that are now in use can be divided into static as well as dynamic algorithms, similar to those addressed in earlier research[23]. They also discovered difficulties in solving the load balance issue. The literature has long mentioned a number of difficulties, including nodes' geographic spread, migration periods, system performance, power management, and safety. In reality, the authors evaluated many QoS parameters, including throughput, velocity, response time, migrating time, etc., to conventional load balancing algorithms. The study came to the conclusion that metrics are traded off. The writer's main drawback is that, out of a large number of load balancing techniques, only 8 are compared.

Shah et al. discussion of the load balancing algorithm's includes a thorough review[24]. According to the system status, the various load balancing techniques were categorised as both static and dynamic, homogeneous as well as heterogeneous, and VM kind homogeneity. The load balancing techniques were categorised using performance measures as well. The benefits and drawbacks of every algorithm were also covered. The literature is not systematically discussed in the work.

Load balancing methods within software-defined networks were reviewed by Neghabi et al. and were broadly categorised into deterministic and non-deterministic methods, in addition to associated metrics that were thoroughly examined[25]. The study asks some significant issues and attempts to provide answers in terms of their importance, metric analysis, function, and difficulties encountered in software depended network load balancing. The research done by the authors outlines the benefits and restrictions of the body of literature that already exists within communication networks. Even though it does not specifically address the subject of cloud computing, the article has a robust foundation and high correlation across load balancing indicators. Additionally, rather than using a hierarchical categorization system, the study employs one level classification.

The solution put forward by Hu et al. in 2010 used the evolutionary algorithm of load balancing among virtual machines, while it also looked at system shifts over time as well as the system's present state[26]. This approach also computes in advance the results of installing virtual machines onto host machines. This technique reduces the dynamic migration for virtual machines while achieving load balancing.

A load balancing technique was put forth in 2011 by Jiandun and colleagues to optimise resource usage in a cloud setting. The algorithm is a technique for managing dynamic resources[27]. The objective of this approach is to efficiently divide the workload over all available virtual machines which aren't within the higher or lower bound. The outcomes of the experiment demonstrate how this method enhances resource usage and speeds up response.

In order to facilitate load balancing, Razali et al. established a taxonomy of virtual machines. Virtual machines moved in this manner to 2 distinct resource classes: higher-power hosts and lower-power hosts based on MIPS (Million Instructions per Second). The movement of virtual machines is dependent onto the steady state CPU use. The quantity of migrations is reduced and energy is conserved during idle state by using this strategy[28].

PROPOSED ALGORRITHM

As a result, the following is how our proposed algorithms differ from other algorithms: Include within the cloudlet job's, queue job and the anticipated finish time of every resource (virtual machine). Depending on this variable, the algorithm chooses the VM for operation allocation that has the shortest projected completion time as well as the least usage rate. The queue records (Cloudlets) of all systems, the queue listing submitted for every virtual machine, the proportion of utilisation of every virtual machine (conveyed via the amount of virtual demands submitted by every virtual machine as well as the estimated cost of accomplishing that queues), and the anticipated completion time whenever a demand is received by every virtual machine are all contained within our proposed algorithm. The Figure 1 portrays the Architectural Flowchart of Proposed Algorithm.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

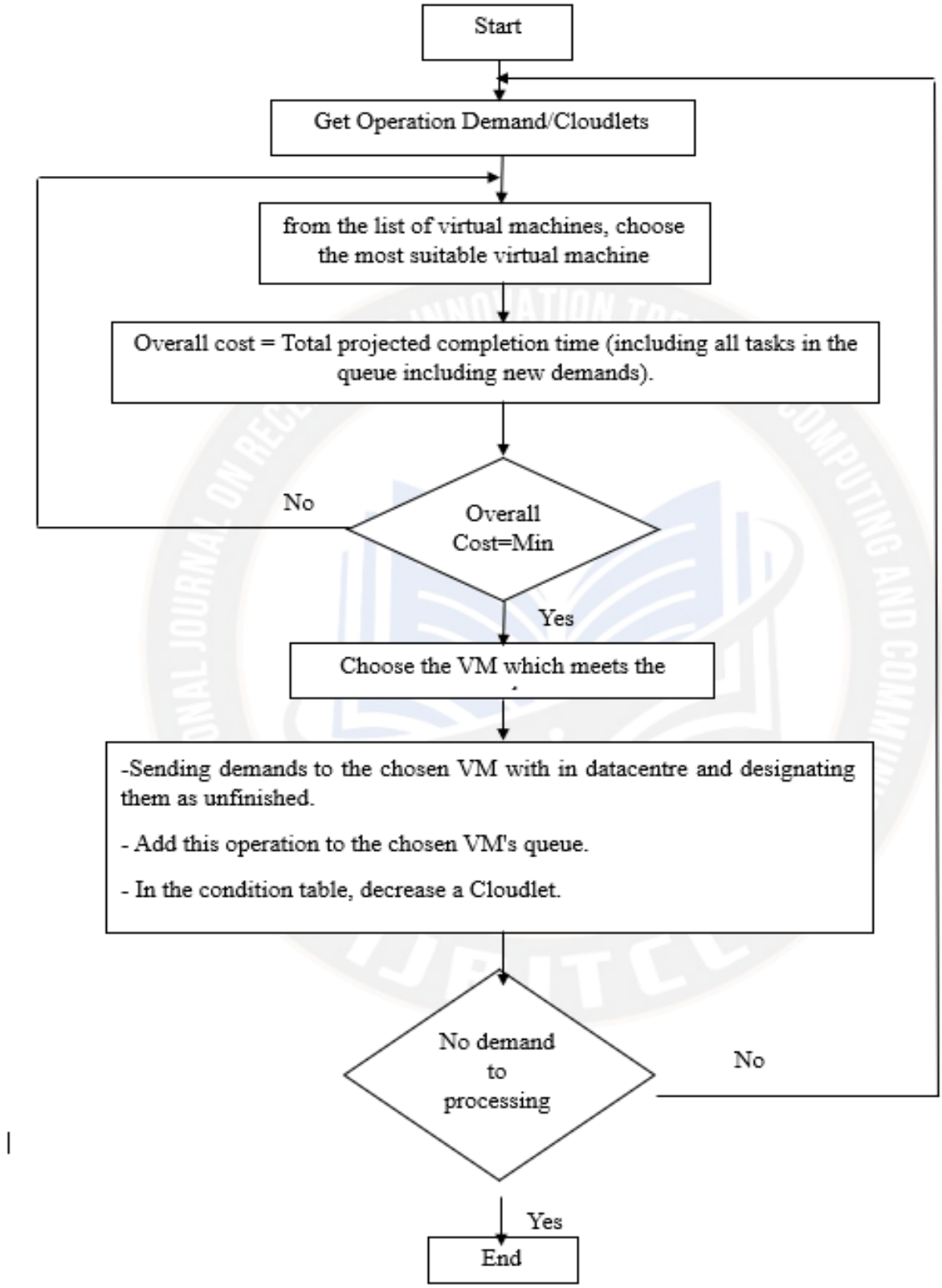
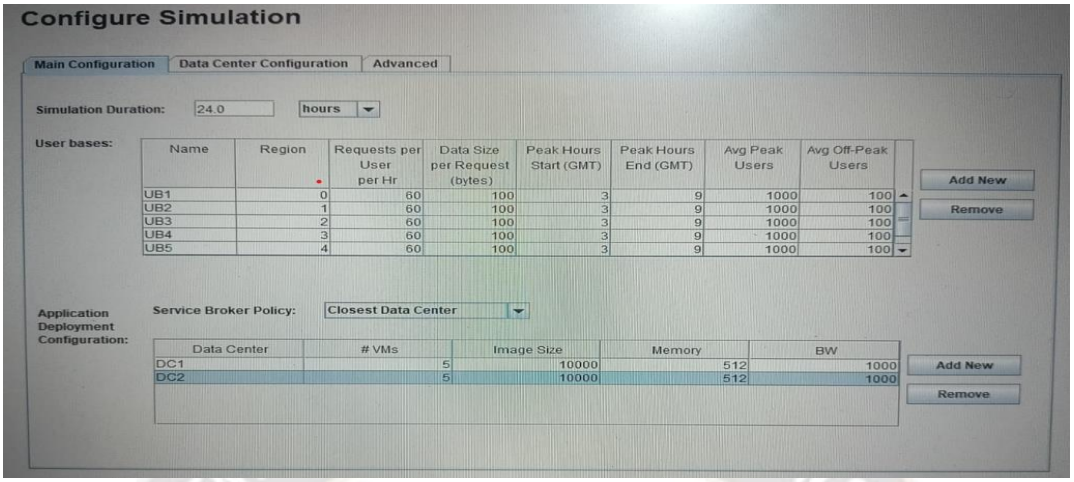
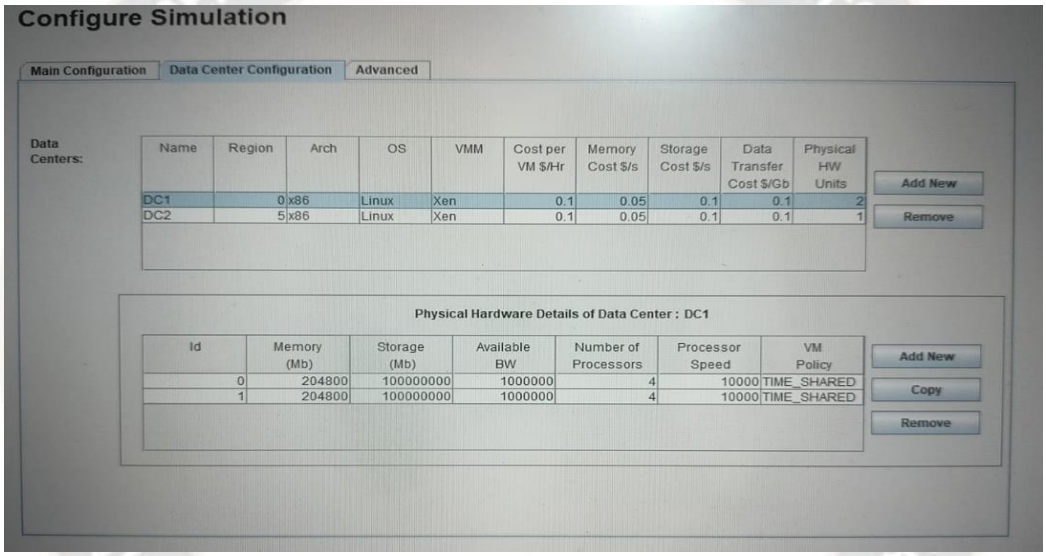


Fig.1: The Figure Demonstrates the Architectural Flowchart of Proposed Algorithm

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

The images below the various settings for the configuration simulation-





Delay Matrix

The transmission delay between regions. Units in milliseconds

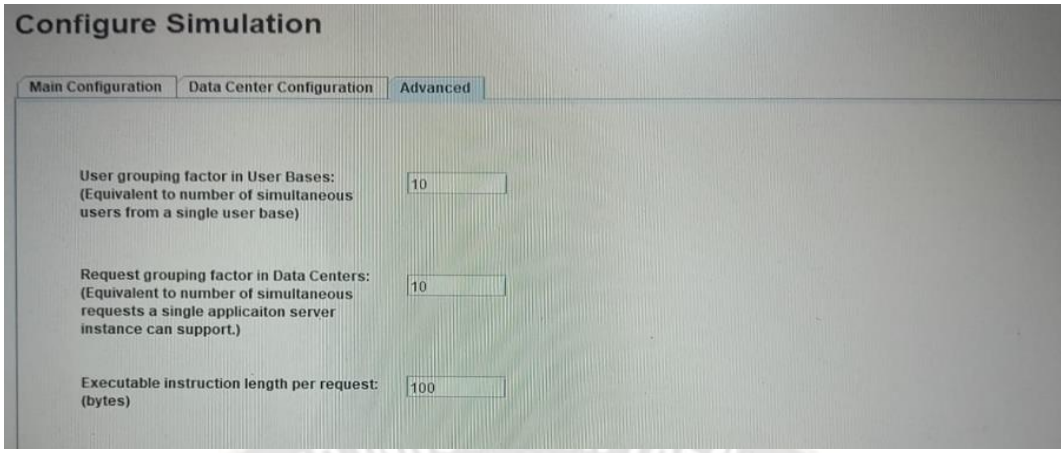
Region\Region	0	1	2	3	4	5
0	25	100	150	250	250	100
1	100	25	250	500	350	200
2	150	250	25	150	150	200
3	250	500	150	25	500	500
4	250	350	150	500	25	500
5	100	200	200	500	500	25

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

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*



RESULTS

The Snapshots Portrays the Results of the Simulation Configuration

Overall Response Time Summary

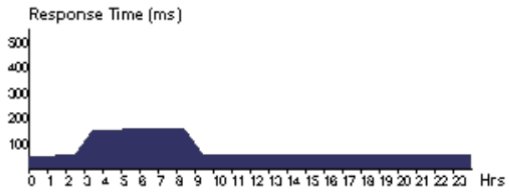
	Avg (ms)	Min (ms)	Max (ms)
Overall response time:	340.87	37.65	763.91
Data Center processing time:	73.66	0.02	313.58

Response Time by Region

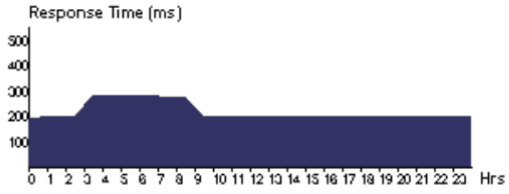
Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	134.30	37.65	254.56
UB2	263.83	146.46	429.14
UB3	353.77	218.87	529.94
UB4	533.22	363.79	758.87
UB5	533.13	361.26	763.91
UB6	223.21	39.53	368.63

User Base Hourly Response Times

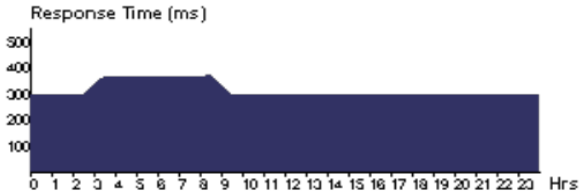
UB1



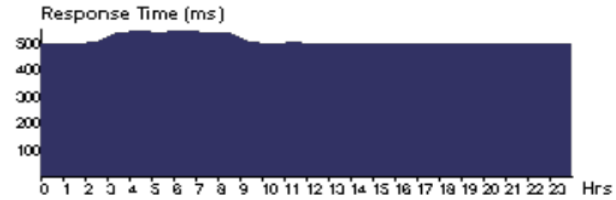
UB2



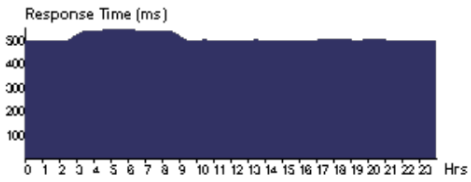
UB3



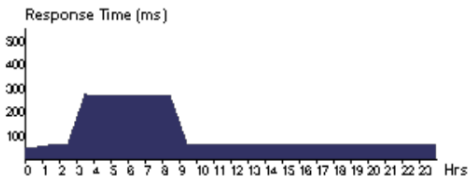
UB4



UB5



UB6

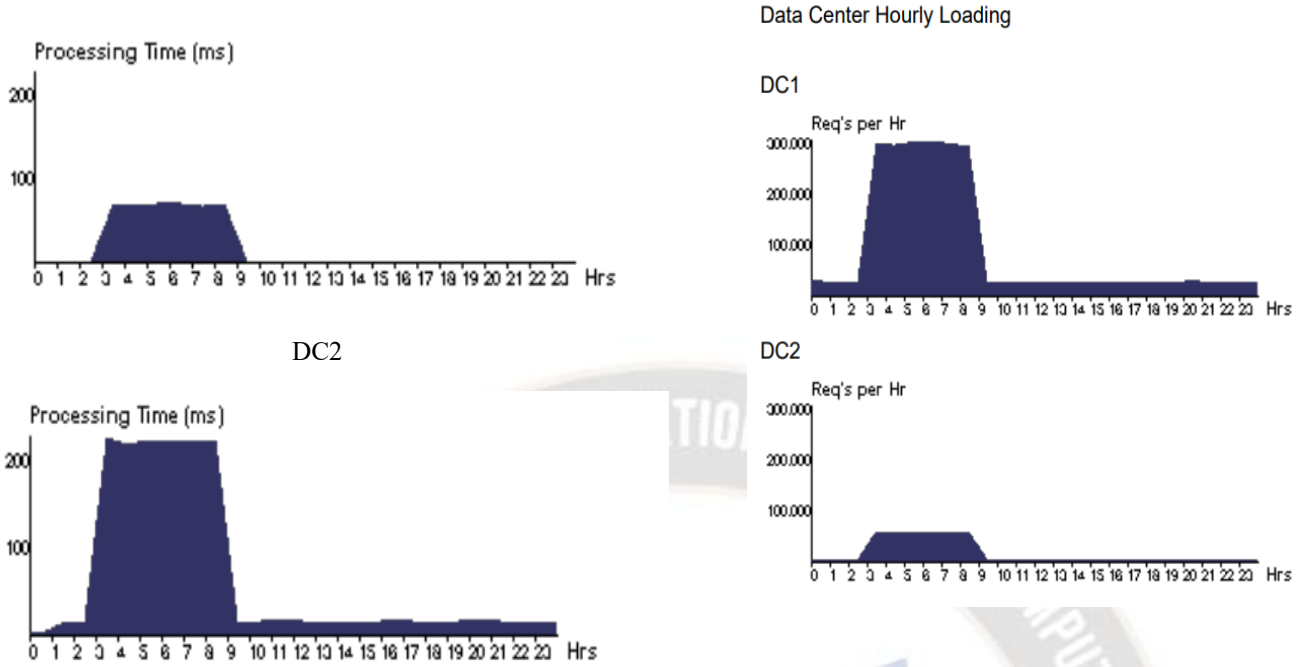


Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	53.76	0.02	199.81
DC2	173.56	0.04	313.58

DC1

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*



CONCLUSION

To improve the performance of cloud computing, we suggested a hybrid algorithm. With the suggested algorithm, the study is capable of completing the tasks and needs of virtual machine while also concentrating upon the load. Because of the diverse load distribution amongst virtual machines with in cloud system, the processing time expenses for every virtual machine may vary. Select the virtual machines that require the lowest processing time to distribute jobs for effective load balancing. The proposed hybrid algorithm suggested for the utilisation and put to the test with in CloudSim cloud computing system. In this post, we employ the same scheduling with virtual computers and tasks like Space-shared - Timeshared. Result shows that the algorithm has hugely enhanced response time as well as overall processing time.

The suggested hybrid algorithm portrays the overall response time 340.87 ms (min 37.65 and max 763.91) and the average datacentre processing time 73.66 ms (min 0.02 and max 313.58), which portrays that the proposed algorithm shows excellent results and lower the response and processing time of request that are clearly shown in the simulation configurations results tables and graphs.

We will test this algorithm with in actual world over time for enhance efficiency, and we'll also take other factors into account for effective resource management, like cost, failure, etc. We will modify and improve the method to address the load balancing issue in a situation of surge workload. Future research will focus on ways to enhance the algorithm's performance. Additionally, because the actual cloud system will produce issues with response times, that will be corrected more logically and effectively by building up algorithms, this will enable us to do study in greater extent and detail.

References

[1] "Cloud computing: methods and practical approaches," Choice Rev. Online, 2014, doi: 10.5860/choice.51-3290.

[2] M. Walterbusch, B. Martens, and F. Teuteberg, "Evaluating cloud computing services from a total cost of ownership perspective," Manag. Res. Rev., 2013, doi: 10.1108/01409171311325769.

[3] M. Rahimi, N. Jafari Navimipour, M. Hosseinzadeh, M. H. Moattar, and A. Darwesh, "Toward the efficient service selection approaches in cloud computing," Kybernetes, 2022, doi: 10.1108/K-02-2021-0129.

[4] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," Journal of King Saud University - Computer and Information Sciences. 2020. doi: 10.1016/j.jksuci.2018.01.003.

[5] B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm with Reinforcement Learning," IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3149955.

[6] E. Gures, I. Shayea, M. Ergen, M. H. Azmi, and A. A. El-Saleh, "Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey," IEEE Access. 2022. doi: 10.1109/ACCESS.2022.3161511.

[7] S. K. Maurya and G. Sinha, "Load balancing in cloud computing: an analytical review and proposal," Indonesian Journal of Electrical Engineering and Computer Science. 2022. doi: 10.11591/ijeecs.v26.i3.pp1530-1537.

[8] M. Livny and M. Melman, "Load balancing in homogeneous broadcast distributed systems," ACM SIGMETRICS Perform. Eval. Rev., 1982, doi: 10.1145/1010631.801689.

[9] Y. Jadeja and K. Modi, "Cloud computing - Concepts, architecture and challenges," in 2012 International Conference on Computing, Electronics and Electrical Technologies, ICCEET 2012, 2012. doi: 10.1109/ICCEET.2012.6203873.

[10] A. Khiyaita, H. El Bakkali, M. Zbakh, and D. El Kettani, "Load balancing cloud computing: State of art," in Proceedings of the 2nd National Days of Network Security and Systems, JNS2 2012, 2012. doi: 10.1109/JNS2.2012.6249253.

[11] P. Mishra, E. S. Pilli, V. Varadharajan, and U. Tupakula, "Intrusion detection techniques in cloud environment: A survey," Journal of Network and Computer Applications. 2017. doi: 10.1016/j.jnca.2016.10.015.

[12] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 2010. doi: 10.1109/AINA.2010.32.

[13] S. Mohapatra, K. Smruti Rekha, and S. Mohanty, "A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing," Int. J. Comput. Appl., 2013, doi: 10.5120/11586-6922.

[14] I. A. Mohialdeen, "Comparative study of scheduling al-gorithms in cloud computing environment," J. Comput. Sci., 2013, doi: 10.3844/jcssp.2013.252.263.

[15] A. R. Dash, S. kumar Sahu, and S. K. Samantra, "An Optimized Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum," Int. J. Comput. Sci. Eng. Inf. Technol., 2015, doi: 10.5121/ijcseit.2015.5102.

[16] V. Behal and A. Kumar, "Cloud computing: Performance analysis of load balancing algorithms in cloud heterogeneous environment," in Proceedings of the 5th International Conference on Confluence 2014: The Next Generation Information Technology Summit, 2014. doi: 10.1109/CONFLUENCE.2014.6949291.

[17] A. S. Mondal and S. Chattopadhyay, "Comparative Analysis of Load Balancing Algorithms in Cloud Computing," 2022. doi: 10.1007/978-981-16-5207-3_28.

[18] K. Nishant et al., "Load balancing of nodes in cloud using ant colony optimization," in Proceedings - 2012 14th International Conference on Modelling and Simulation, UKSim 2012, 2012. doi: 10.1109/UKSim.2012.11.

[19] E. Jafarnejad Ghomi, A. Masoud Rahmani, and N. Nasih Qader, "Load-balancing algorithms in cloud computing: A survey," Journal of Network and Computer Applications. 2017. doi: 10.1016/j.jnca.2017.04.007.

[20] A. S. Milani and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," Journal of Network and Computer Applications. 2016. doi: 10.1016/j.jnca.2016.06.003.

[21] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," Egyptian Informatics Journal. 2015. doi: 10.1016/j.eij.2015.07.001.

[22] M. Mesbahi and A. Masoud Rahmani, "Load Balancing in Cloud Computing: A State of the Art Survey," Int. J. Mod. Educ. Comput. Sci., 2016, doi: 10.5815/ijmecs.2016.03.08.

[23] R. Kanakala and V. K. Reddy, "Performance Analysis of Load Balancing Techniques in Cloud Computing Environment," TELKOMNIKA Indones. J. Electr. Eng., 2015, doi: 10.11591/telkomnika.v13i3.7237.

[24] J. M. Shah, K. Kotecha, S. Pandya, D. B. Choksi, and N. Joshi, "Load balancing in cloud computing: Methodological survey on different types of algorithm," in Proceedings - International Conference on Trends in Electronics and Informatics, ICEI 2017, 2018. doi: 10.1109/ICOEI.2017.8300865.

[25] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature," IEEE Access. 2018. doi: 10.1109/ACCESS.2018.2805842.

[26] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in Proceedings - 3rd International Symposium on Parallel Architectures, Algorithms and Programming, PAAP 2010, 2010. doi: 10.1109/PAAP.2010.65.

[27] J. Li, J. Peng, Z. Lei, and W. Zhang, "An energy-efficient scheduling approach based on private clouds," J. Inf. Comput. Sci., 2011.

[28] R. A. M. Razali, R. A. B. Rahman, N. Zaini, and M. Samad, "Virtual machine migration implementation in load balancing for Cloud computing," in 2014 5th International Conference on Intelligent and Advanced Systems: Technological Convergence for Sustainable Future, ICIAS 2014 - Proceedings, 2014. doi: 10.1109/ICIAS.2014.6869540.