



Static Round Robin Load Balancing Algorithm in Cloud Computing Environment

Mohit B. Rami^{1*}, Vivek. K. Shah^{2*}

M.Tech. Scholar, Computer Engineering, Sankalchand Patel University, Visnagar, India¹

Assistant Professor, Computer Engineering, Sankalchand Patel University, Visnagar, India²

Abstract: Recently, distributed computing has emerged as a buzzword within the cloud computing community, with many believing that it will transform the IT industry significantly. It provides us with the means to access applications through the internet. Additionally, it refers to the management, arrangement, and access of applications online. Cloud computing is a framework that enables widespread, convenient, on-demand network access to a shared pool of adjustable computing resources (such as networks, servers, storage, applications, and services) that can be quickly provisioned and deployed with minimal management effort or service provider interaction. Most enterprises today are increasingly relying on these services to lower infrastructure and maintenance expenses, leading to a gradual rise in cloud demand. Load balancing is one of the significant challenges that cloud computing faces today. It essentially means that there should be a strategy to ensure that no node becomes overloaded. Cloud computing represents a growing trend in service computing that provides users with a range of on-demand services, including applications, processing power, and storage, all based on the "Pay-As-You-Use" model. Organizations from various sectors are currently recognizing the benefits provided by cloud computing technology and are transitioning towards it. Cloud computing delivers several advantages compared to conventional computing methods. Nevertheless, it still encounters a few issues, such as resource utilization in a cloud data center and quality of service to end-users due to improper load balancing among available resources. Diverse cloud resources also affect the overall performance of cloud systems. In this research paper, we put forward an enhanced load balancing algorithm for effective virtual machine allocation in a heterogeneous cloud environment. Our proposed algorithm efficiently allocates free user requests to available virtual machines in the cloud data center to ensure proper load balancing. This algorithm aims to reduce the user request response time as well as the processing time required by the data center. The results obtained demonstrated a significant decrease in user request response time when using "Cooperative Effort (RR)" algorithms. In this paper, we introduce a modified cooperative resource allocation algorithm designed to meet user demands by minimizing response time.

Keywords: Cloud Computing, Load Balancing, Static Round Robin Load Balancing Algorithms, Cloud Computing, Cloud Model, Cloud Services, Cloudsim, Cloudanalyst.

I. INTRODUCTION

Distributed computing is gaining popularity in recent times. As technology continues to advance continuously, it is nearly impossible for a business to remain competitive without keeping its resources and technologies updated. These technological demands require significant investment in infrastructure, which may not be feasible for all businesses. Cloud computing addresses these challenges by providing services on an on-demand basis that can be accessed via the internet and paid for based on usage. Nowadays, most businesses, academia, and corporations have adopted it primarily due to the flexibility and ease of accessing services on demand for hardware or software. It has the capability to utilize all types of resources, whether they are available locally or remotely, in a cost-effective manner.

According to the NIST definition, cloud computing is a model that enables 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 quickly provisioned and released with minimal management effort or service provider interaction. Currently, cloud computing has become very popular among the community of cloud users by providing a variety of resources. Cloud computing platforms, such as those offered by Microsoft, Amazon, Google, IBM, and Hewlett-Packard, allow developers to deploy applications across computers managed by a central organization.

Developers benefit from a managed computing platform without having to allocate resources for designing, building, and maintaining the infrastructure. This paper is organized as follows: Sections 4, 5, and 6 focus on the related work regarding existing load balancing algorithms, while Section 7 presents the proposed work. Sections 8 and 9 discuss the Cloud Simulator, the methodology, and implementation detailed in Section 10, and Section 11 concludes with future work outlined in this paper.

II. CLOUD DEPLOYMENT MODEL

There are four primary types of cloud computing deployment models:

2.1 Public Cloud:

When cloud services are provided by a company for general public use, it is referred to as a public cloud. Public cloud providers leverage the internet to deliver resources such as storage and applications on a public cloud. In this case, the cloud infrastructure is managed by external service providers or organizations.

Public clouds contribute to the realization of features such as:

- An Adaptable and Scalable Environment
- Autonomy of Self-Service

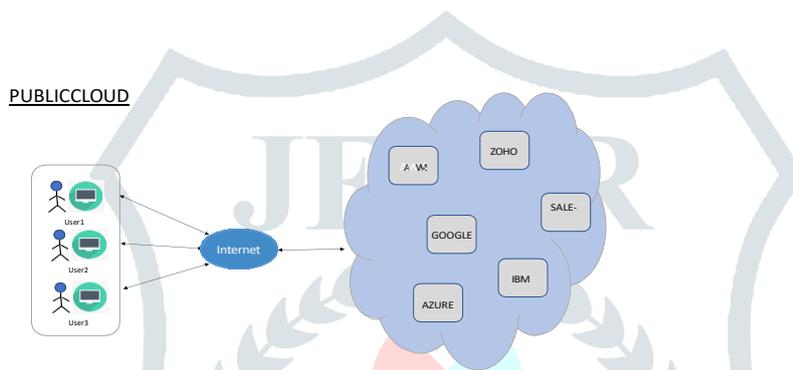


Fig.1PublicCloud

- Payforwhatyouuse
- AvailabilityandReliability

2.2 Private Cloud:

In a Private Cloud, the cloud infrastructure is maintained exclusively within a single organization. It is more secure due to its internal use, as only specified users within the organization can access the services offered by the cloud infrastructure. Private clouds can be costly as they generally lack the benefit of economies of scale.

Private clouds (illustrated in fig. 2) facilitate the realization of features such as:

- Improved Security Measures

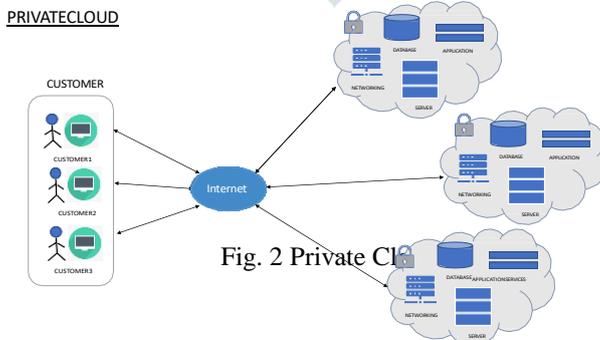


Fig. 2 Private Cloud

- Reserved Resources
- Increased Customization
- DedicatedResources
- GreaterCustomization

2.3 Hybrid Cloud:

A Hybrid Cloud consists of any combination of at least two distinct cloud infrastructures. It may include a private cloud alongside one or more public clouds. Resources should be shared between the different clouds.

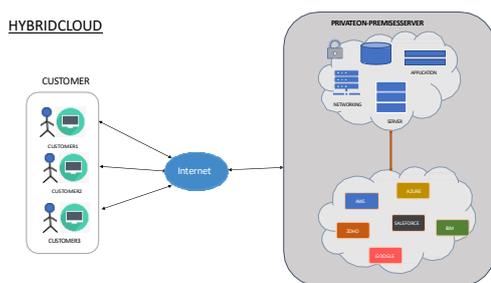


Fig.3.HybridCloud

Hybrid Clouds(as shown in fig 3) realize the characteristics:

- Optimal utilization
- Data centre consolidation

2.4 CommunityCloud:

In Community Cloud a few association mutually develop and share a similar cloud foundation just as arrangements, necessities, worth and concerns. The cloud framework cloud be facilitated by third get-together merchant or inside one of the organization locally.

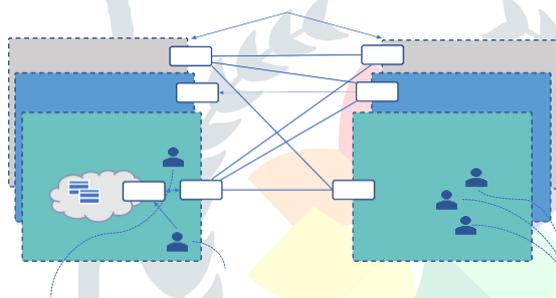


Fig.4CommunityCloud

IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service).

III. CLOUD SERVICE MODEL

Cloud computing providers deliver their services based on several essential models (as illustrated in fig.5):

3.1 Infrastructure as a Service (IaaS)

IaaS grants access to a variety of resources such as physical machines, virtual machines, and virtual storage. IaaS also offers virtual LAN (VLAN) network devices, IP addresses, software bundles, load balancers, and disk storage. By utilizing IaaS, cloud providers can effectively manage infrastructure over the internet in a cost-efficient way.

3.2 Platform as a Service (PaaS)

PaaS provides runtime application environments for applications. Essentially, it offers a platform for application development. One of its key benefits is the reduced costs since there is no need to buy hardware and software. It offers a browser-based environment for developing applications. It is a secure, scalable web service that simplifies the workflow approval process.

3.3 Software as a Service (SaaS)

SaaS delivers software applications to users as a service. It is commonly utilized in CRM applications similar to Salesforce. SaaS is also applied in helpdesk solutions, HR software, billing, and invoicing. It makes software available online while the vendor manages maintenance.

IV. LITERATURE REVIEW

T. Dillon et al. [1] This study examined the challenges and issues related to Cloud management. We discussed the connections between Cloud computing, Service-Oriented Computing, and Grid processing. Several difficulties in the journey towards achieving Cloud computing were outlined. The interoperability challenge was emphasized, and various solutions for different cloud management models were explored subsequently.

P. Kumar et al. [2] Load-balancing strategies are essential for energy conservation and task management, necessitating the development of new algorithms. It has become crucial to review both recently developed and traditional load-balancing approaches across various classifications based on test systems that rely on different load-balancing metrics to assess the viability of these algorithms before deploying them in a real cloud environment.

A. Hota et al. [3] The Cloud computing framework has been widely accepted by both industry and academia. Nonetheless, numerous issues persist in this arena, such as load balancing. This paper presents various load-balancing algorithms, highlighting their advantages and disadvantages. The review offers ample opportunities for researchers to devise effective load-balancing algorithms suitable for the cloud environment.

S. Shaw et al. [4] In this paper, we analyze different algorithms for load balancing in Cloud Computing. Load balancing can mean two things: first, it involves managing numerous concurrent accesses or data traffic directed to multiple nodes to minimize response time for users; second, it reallocates resources from a heavily loaded node to various nodes to enhance each node's resource utilization.

N. Shah et al. [5] Load balancing is crucial in distributed computing to ensure effective and maximum utilization of resources. This paper discusses current static load-balancing schemes available in distributed computing, while also identifying gaps in existing static load-balancing algorithms.

P. Samal et al. [6] This paper addresses the resource allocation problem across various nodes of a distributed system to enhance both resource utilization and job response time by evaluating variations in the Round Robin algorithm. The issues of overloading and underutilization are mitigated. Consequently, load balancing ensures that all processors in the system or each node in the network performs approximately the same amount of work at any given time. The proposed algorithm demonstrates improved response time when compared to other algorithms.

S. Patel et al. [7] This paper discusses Cloud Computing along with the research challenges in load balancing. It also focuses on the pros and cons of distributed computing. Significant emphasis is placed on analyzing load-balancing algorithms, followed by a comparative study of the previously mentioned algorithms in distributed computing with respect to reliability, resource utilization, whether they are static or dynamic, supportive or non-supportive, and process migration.

S. Patidar et al. [11] In this paper, the enhanced algorithm takes into account the limitations of each Virtual Machine (VM) and the job duration of each requested task to allocate workloads to the most appropriate VMs.

D. Chitra Devi et al. [8] In this paper, the overall completion time of all participating tasks across different algorithms was considered. Ultimately, future improvements could focus on the total duration of each job, which can be examined within various scheduling and load-balancing algorithms. The algorithms could be further refined to achieve better predictive outcomes across various aspects. Ultimately, assessment results should be gathered for distinct job scheduling designs across all three different scheduling and load-balancing algorithms.

H. Shoja et al.[9] Managing variations in workload is a critical challenge in distributed computing. It is important to distribute the workload fairly among all the nodes to maximize user satisfaction and resource utilization, ensuring that each processing resource is allocated effectively and appropriately.

Sajjan R.S et al. [12] highlight that load balancing is arguably the most critical challenge in distributed computing.

It is a mechanism that evenly distributes the workload across all nodes within the cloud environment. This can lead to increased user satisfaction and better resource utilization, ultimately enhancing the overall performance and resource efficiency of the system.

S. Mayur et al. [14] note that load balancing remains a major challenge in distributed computing today. This paper reviews various previously discussed load balancing techniques, which primarily focus on reducing overhead, improving system performance, enhancing resource usage, and optimizing user response time and throughput through cooperative effort and weighted utility function.

VI. RELATED WORK

The fundamentals of load balancing:

Load balancing is utilized to distribute workloads across multiple nodes to optimize the overall performance of a system. The existing load balancing algorithms in the cloud computing environment are not very effective. Load balancing has become a highly complex task today due to the unpredictable nature of user requests arriving at the server. Recently, there has been significant research conducted on load balancing. Generally, load balancing is classified into two categories: static load balancing and dynamic load balancing algorithms. Static Round Robin load balancing algorithms are mostly defined during the design or implementation of the system. In load balancing, the capability to distribute the incoming load from user requests among virtual machines is essential.

Researchers have proposed several load balancing algorithms, such as Round Robin, Throttled Load Balancing, and Active Monitoring Load Balancing. Here, it is proposed to implement the current algorithms and compare them based on response time to determine which algorithm performs better. To assess the performance of load balancing in a cloud environment, cloud simulators play a crucial role as it is challenging to conduct experiments in a real cloud setup. The algorithms are implemented and simulated in CloudAnalyst, which is used for analyzing large-scale applications running in the cloud and offers a GUI for user simplicity.

VI. AIM

Our objective is to implement the Static Round Robin algorithm using CloudSim 3.0. We will also implement load balancing algorithms such as Round Robin with the goal of reducing response time. This outline focuses on conducting a qualitative performance analysis of existing VM load balancing algorithms, followed by implementation in CloudSim and Java programming language.

VII. EXISTING SYSTEM

Ramadhan et al. [17] conducted experiments to determine the effectiveness of load balancing algorithms. The authors have highlighted that the number of user requests for cloud services is substantial and the nature of the workloads also varies. They performed simulation tests using a "Cooperative" (RR) algorithm [18-19]. Initially, when a user request comes in, the algorithm selects a VM from the initial VM list table; however, for subsequent requests, it processes from the next VM that was previously assigned. The VM list table continuously scans from an initial list each time a new request is received. The results indicate an improvement in response time.

In many previous studies, the heterogeneous arrangement of resources in cloud data centers and the actual use of processing elements have not been taken into account in the load balancing strategy. An efficient load balancing algorithm must address these issues for effective utilization of all the available resources in the cloud data center, based on their processing capabilities. By considering these factors, the quality of services provided to cloud users and the processing time at data centers can be enhanced. The Round Robin Algorithm is regarded as superior to traditional load balancing algorithms in earlier research studies. This work focuses on all these issues and suggests an enhancement over the cooperative load balancing algorithm to achieve improvements in response time and data center processing time in a cloud environment.

VIII. PROPOSE WORK

The aim of our proposed work is primarily to reduce the response time for a user request and the processing time in the heterogeneous cloud data center. The Data Center Controller (DCC) collaborates with the proposed algorithm deployed at the VmLoadBalancer to determine the appropriate VM for the allocation of a user request. The performance analysis of the proposed work in relation to "Cooperative" algorithms.

8.1 Round Robin Algorithm:

Cooperative is the most basic and straightforward scheduling algorithm. It operates based on a time quantum, with each processor requiring this time quantum for processing. It functions in a circular manner. In this scheme, the data center controller assigns requests to VMs on a rotational basis. If the process does not complete within the allotted time, it waits for the final process and moves to the end of the line. A significant challenge is that it cannot finish the cycles because some processes may become overloaded and thus must remain in the waiting line.

IX. CLOUD SIMULATOR

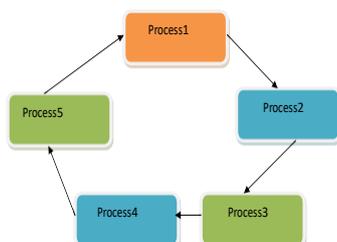


Fig.6RoundRobinAlgorithm

Cloud simulators play a crucial role in executing large-scale applications that operate on the cloud. This is a toolkit that supports both modeling and simulation of cloud-based systems and application provisioning. 9.1 CloudSim: CloudSim is a simulation tool that enables developers to evaluate the performance of their provisioning strategies at no cost. Additionally, it serves as a library for simulating cloud environments. The primary goal of CloudSim is to offer a general, extensible simulation framework that facilitates

modeling, simulation, and experimentation of cloud computing systems and application services. It is designed for researchers who are working on various sets of problems to enhance cloud computing technology.

9.2 CloudAnalyst: CloudAnalyst represents a comprehensive version of CloudSim, showcasing unique systems and advanced visualization capabilities. It is utilized for analyzing large-scale applications that operate within the cloud. The software provides a user-friendly graphical interface, making it easy for users to comprehend simulation experiments. It allows for the creation of simulations with a high degree of configurability and adaptability, additionally enabling researchers to execute simulations repeatedly and conduct a series of exploratory simulations by making minor adjustments promptly and easily. CloudAnalyst is the cloud simulator designated for straightforward setup during simulation.

X. METHODOLOGY

Load balancing is employed to distribute the workload from larger processing nodes to smaller processing nodes to enhance the overall system performance. In a cloud computing environment, load balancing is essential to distribute the workload evenly across all nodes. This facilitates a fair allocation of computing resources to achieve high user satisfaction and efficient resource utilization. Optimal resource utilization and effective load balancing contribute to minimizing resource consumption. This approach aids in execution failures, scalability, and the avoidance of bottlenecks. All these functions are achieved by employing load balancing algorithms within CloudAnalyst.

10.1 Implemented Algorithms:

The load balancing algorithms utilized in this research are as follows:

RoundRobin Algorithm

- a) The Round Robin VM load balancer keeps track of an index of virtual machines (VMs) and their states (busy/available). Initially, all VMs have zero allocation.
- b) The datacenter controller receives user requests/cloudlets and records the arrival time and burst time of these requests. The allocations to VMs are made based on their statuses determined from the VM queue. The Round Robin VM load balancer is responsible for assigning the time quantum for the execution of user requests.
- c) The Round Robin VM load balancer evaluates the turnaround time for each process, calculating the response time and average waiting time for user requests, thereby determining the scheduling order.
- d) After the execution of cloudlets, the Round Robin VM Load Balancer deallocates the VMs.
- e) The datacenter controller then checks for any new, pending, or waiting requests in the queue.
- f) The process continues from step 2.

10.2 Implementation Details

The implemented load balancing algorithms have been tested within CloudAnalyst. CloudAnalyst incorporates the CloudSim library, which is a Java-based code library. This library can be directly utilized by integrating it with the JDK to compile and execute the code.

Execution steps:

- Download and install Eclipse, then import CloudAnalyst into Eclipse.
- Launch CloudAnalyst, which will automatically open the GUI for the application.
- Implement the algorithms into CloudAnalyst and run the simulation.
- Set all parameters necessary for running the simulation.

Simulation Setup:

- Parameter Value
- Size of VM Image 10000 byte
- Memory Size 512 MB
- Bandwidth 1000 byte
- Architecture x86
- Operating System “Linux”
- VMM/Hypervisor “Xen”



Fig.7 GUI of Cloud Analyst

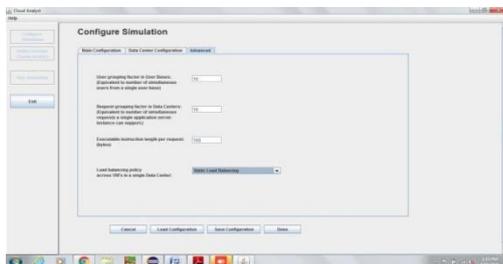


Fig.8 Adding algorithm in Cloud Analyst

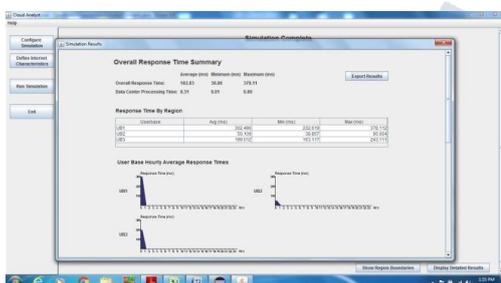


Fig.9 Overall IResponse Time Summary

10.3. Results and Discussion: Two distinct scenarios are analyzed to evaluate the performance of our proposed algorithm. The number of data centers (DCs), virtual machines (VMs), and the arrangement of representatives vary for each scenario. The simulation is repeated for "Cooperative effort" and our proposed algorithm to assess the outcomes. The performance metrics are:

a. Average Response Time

"This represents the duration it takes for the system to react to a user request." A quicker response is preferable to ensure a satisfactory quality of service for cloud clients.

Response Time = Finisht – Arrivalt + TransmissionDelay Where,

Finisht = Completion time of request processing Arrivalt = Time at which the request arrives

TransmissionDelay = Network Delay between the source and the destination for delivering the response to the user

b. Average Datacenter Processing Time

"This indicates the amount of time the datacenter takes to process all incoming requests from the user bases." It is calculated using the following formula.

Processing Time = LVMs/CVMs Where,

LVMs = Total load of all VMs in a datacenter CVMs = Total capacity of all VMs in a datacenter Case 1: One Datacenter with 50 VMs In this scenario, only a single datacenter is employed, equipped with 50 VMs. Figure 11 illustrates that both the "Cooperative effort (RR)" and our proposed algorithm require more time for the specified performance metrics. In the "Cooperative effort" algorithm, VMs are assigned in a circular manner without taking into account their processing capabilities such as processing power, number of processors, RAM, etc. The results from our proposed algorithm indicated a 35% reduction in user request response time and a 38% reduction in datacenter processing time.



Fig.10Output screen of simulation

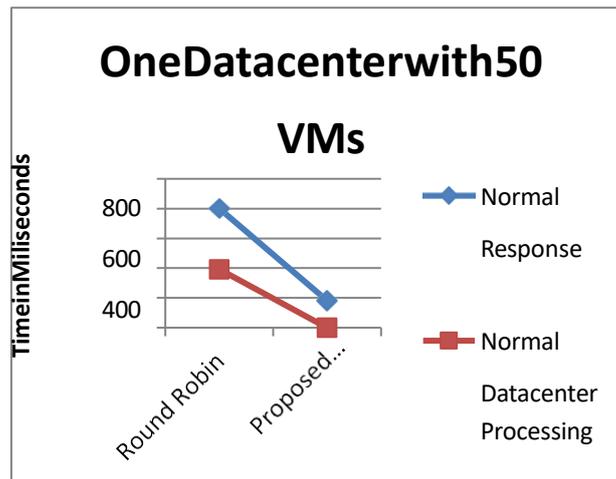


Fig.11 One Data centre with 50 VMs

Case 2: Two Data Centers with 50 Virtual Machines

In this scenario, the virtual machines (VMs) in a data center are scaled up to analyze the effects on response times and data center processing duration. Unlike Case 1, which involved a single data center with 50 VMs, this case features two data centers, each housing 50 VMs. We applied the "Optimized Response Time" management strategy.

The results, illustrated in Figure 12, indicate an enhancement over Case 2 in terms of both performance metrics due to the effective distribution of peak load. In Case 1, the single data center with 50 VMs experienced overload during peak hours. By increasing the number of VMs, the peak load can be effectively spread across the VMs. The proposed algorithm demonstrates superior performance compared to the "Round Robin" method. The response time has been significantly reduced by 33%, and the processing time at the data center has decreased by 39%.

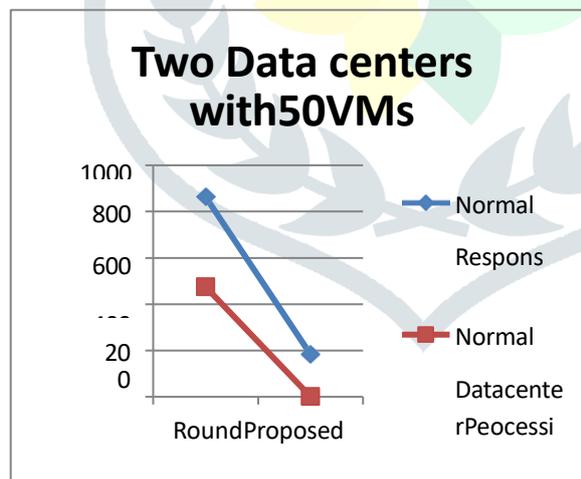


Fig.12 Two Data centres with 50 VMs

XI. CONCLUSION

Load balancing distributes the overall workload across all individual nodes to enhance resource utilization and response speed. This approach prevents scenarios where some nodes are overloaded while others remain idle or perform minimal tasks. In this context, the existing algorithms were implemented, and performance was evaluated using the CloudAnalyst simulator. Performance was assessed based on response time. The Round Robin Static Load Balancing algorithm demonstrated improved response times in both CloudAnalyst and CloudSim with Java. Future research could consider various metrics related to response time for the proposed algorithm. This algorithm aims to integrate load balancing techniques, both cooperative existing systems and proposed methods, to achieve higher resource utilization and improved response times. These algorithms operate within CloudAnalyst and CloudSim with Java, allowing comparisons based on response times.

REFERENCES

- [1] Tharam Dillon, Chen Wu, and Elizabeth Chang, "Cloud Computing: Issues and Challenges," 2010 24th IEEE International Conference on Advanced Information Networking and Application.
- [2] Pawan Kumar, Rakesh Kumar, "Issues and Challenges of Load Balancing Techniques in Cloud Computing: A Survey," ACM comput 2022.
- [3] Arunima Hota, Subasish Mohapatra, and Subhadarshini Mohanty, "Survey of Different Load Balancing Approach-Based Algorithms in Cloud Computing: A Comprehensive Review," Springer 2023.
- [4] Subhadra Bose Shaw, Dr. A.K. Singh, "A Survey on Scheduling and Load Balancing Techniques in Cloud Computing Environment," 2024 5th International Conference on Computer and Communication Technology (ICCT).
- [5] Nadeem Shah, Mohammed Farik, "Static Load Balancing Algorithms In Cloud Computing: Challenges & Solutions," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 4, ISSUE 10, OCTOBER 2023.
- [6] Pooja Samal, Pranati Mishra, "Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing," International Journal of Computer Science and Information Technologies, Vol. 4 (3), 2017, 416-419.
- [7] Shweta Patel, Prof. Mayank Bhatt, "Implementation of Load Balancing in Cloud Computing through Round Robin & Priority using CloudSim," International Journal for Rapid Research in Engineering Technology & Applied Science Vol 3 Issue 11 November 2019.
- [8] D. Chitra Devi and V. RhymendUthariaraj, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non-preemptive Dependent Tasks," Ramanujam Computing Centre, Anna University, Chennai 600 025, India, Received 29 November 2015; Accepted 21 December 2021.
- [9] Hamid Shoja, Hossein Nahid, Reza Azizi, "A Comparative Survey on Load Balancing Algorithms in Cloud Computing," IEEE November 2018.
- [10] Manpreet Kaur (Mtech Scholar), Dr. BK Verma (Professor), "A Review on Various Load Balancing Algorithms with Merits - Demerits in Cloud Computing," International Journal of Advance Engineering and Research Development Volume 5, Issue 05, May - 2021.
- [11] Srushti Patidar, Assistant Prof. Amit Saraf, "Load Balancing in Cloud Computing Using Modified Round Robin Algorithm," International Journal of Scientific Research & Engineering Trends Volume 6, Issue 4, July-August 2022, ISSN (Online): 2395-566X.
- [12] Sajjan R.S, Biradar Rekha Yashwantrao, "Load Balancing and its Algorithms in Cloud Computing: A Survey," International Journal of Computer Sciences and Engineering 2016.
- [13] Shivangi Mayur, Nidhi Chaudhary, "Enhanced Weighted Round Robin Load Balancing Algorithm in Cloud Computing," International Journal of Innovative Technology and Exploring Engineerin (IJITEE), ISSN: 2278-3075, Volume-8, Issue- 9S2, July 2019.
- [14] Aanje Mani Tripathi, Sarvpal Singh, "A Literature Review on Algorithms for Load Balancing in Cloud Computing Environments and Their Future Trends," COMPUTER MODELLING & NEW TECHNOLOGIES 2019 21(1) 64-73.
- [15] P. Mell and T. Grance (2019), "The Definition of Cloud Computing by NIST," NIST Special Publication 800-145, Department of Commerce, USA, pages: 2-3.
- [16] Gema Ramadhan, Tito Waluyo Purboyo, and Roswan Latuconsina (2018), "A Model for Load Balancing in Cloud Computing Utilizing the Throttled Algorithm," International Journal of Applied Engineering Research, Vol. 13, No. 2, pp. 1139-1143.