Load Balancing Technique using Virtualization with WRR Algorithm in Cloud Computing

Foram Patel[1], Dr. Mohit Bhadla[2]
[1,2] Gandhinagar University

***Abstract*—Cloud Computing still grapples with enormous challenges in efficiently managing resource allocation, with load balancing being a core issue affecting system performance, Quality of Service (QoS), and compliance with Service Level Agreements (SLAs). This study advances an innovative load balancing scheme that incorporates virtualization and dynamic task migration for efficient utilization of resources in cloud computing environments. Through the use of virtualization, the model dynamically generates and maintains virtual machine (VM) copies in order to avoid node overload and reduce execution latency. The designed methodology applies a Weighted Round Robin (WRR) technique for task allocation according to server capacities, which guarantees efficient and equitable load distribution. CloudSim simulation results reveal that the model registers significant enhancements in system response time, resource utilization, and operating cost savings. By strategically creating and destroying VM replicas according to individual module requirements, the framework promotes scalability, minimizes storage overhead, and provides guaranteed service availability at all times, providing a realistic and space-saving solution to the challenges of contemporary cloud infrastructures.**

***Index Terms*—Cloud computing, Load Balancing Algorithm, Virtualization, Virtual Machine, Weighted Round Robin, WRR.**

I. INTRODUCTION

Cloud computing is internet-based technology for providing IT related services and resources by public or private network as per the user demands. In a typical Cloud Computing environment, there are two components: the frontend side and the backend side. The frontend is on the user side where it is accessible through connections over the Internet, whereas for the backend side, it deals with cloud service models It is a concept of network computing which offers computing services of both hardware and software platforms and testing tools over the web itself [5]. It provides platform for deployment, easy and convenient access to web services, storage and so on to client to purchase services as per requirements. It allows consumers to connect many configurable assets like processor, memory, networks, application etc. to provide facilities to consumers as pay-per-use rates [8]. In this structure user access the services based on user requirements regardless of cloud or user geographical location. This make it much faster and convenient to develop, resources to be more flexible and cost to be more manageable because instead of installing their application on individuals PCs, user can receive them from the cloud provider in the form application and installed it on set of network servers virtually for hassle free usage.

Load Balancing is a key aspect for distributing work load among the available nodes in network for processing task efficiently and to avoid the situation of overload, underutilized or idle in environment. It is a common issue in cloud computing which affects the performance of the application, Quality of Service (QOS) measures and Service Level Agreement (SLA) document of cloud providers [3]. It ensures equitable and dynamic task distributions and allotment process for effective resource utilizing for processing client requests. It is the efficient and fair assignment of work to computing resources to maintain the load satisfaction (i.e., processor load, the used memory, delays, or the network load) of users and increase the rate of resource productivity [1].

Using the concept of resource virtualization, load balancing in the context of cloud computing refers to the effective and optimal distribution of traffic load among the available nodes. In order to maintain user load satisfaction (i.e., processor load, memory usage, delays, or network load) and boost resource productivity, load balancing is the effective and equitable distribution of work across computer resources [1]. The efficiency of the system is decreased when any virtual machines (VMs) in the network have higher overhead volumes than the others. As a result, the cloud would be restrictive and affect the amount of time needed to finish certain tasks. Furthermore, the cloud system must distribute its resources among itself when it encounters numerous and substantial requests which resultant into rise in productivity, convenience and resource availability for the users [9]. Thus, the proper selection of resources by considering the characteristics of tasks will enhance a system's efficiency; therefore, a mechanism is required to select appropriate resources in responding to user demands [1].

II. LITERATURE REVIEW

Sefati, S.,[1] This research article discusses a very important problem known as load balancing in cloud computing environments. Load balancing is important to increase IT efficiency by managing resources in an optimal way, which ultimately enhances the experience and productivity of users in distributed systems. To address this problem, this paper proposes a solution utilizing a new load balancing approach based on aspects of the Grey Wolf Optimization (GWO) algorithm that exists in nature. The load balancing method proposed in this paper based on the GWO algorithm is validated through simulations and the results indicate substantial improvements in operational costs and response times over other techniques. There is basis to believe that this proposed method could be a viable means to gain much more cost-effective and efficient load balancing in cloud computing infrastructures.

Shahid, M. [2] This research paper evaluates different load balancing algorithms in cloud computing environments. There is a focus on several important performance metrics, most notably response time measured by users and processing costs incurred by the cloud provider. The paper further examined different service broker policies for distributing existing workloads across the available resources. There is somewhat of an emphasis on load balancing in evaluating the actual workloads being processed. Overall, the findings of this research indicate the importance of having an efficient distribution of the resources in cloud environments to help optimize their performance and reduce costs.

Laha, J. [6] This research paper addresses the important issue of load balancing in a cloud computing environment in order to make effective use of resources and minimize application response time. In this case, we examine the Throttled Load Balancer [2], with the intent of distributing requests evenly amongst VMs based on available resources. The principal contribution of the paper indicates that the Throttled Load Balancer improves overall system performance by maintaining an index table of the state of each VM, enabling the authors to make better assignment of the tasks. The main contribution of the paper is a method that either extends or revises the Throttled Load Balancer and shows an increased effectiveness when distributing workload across cloud-based resources than the previous load balancing approaches.

Ghoomi, E.J [4] This research paper offers a complete survey of the key issues raised by load balancing in cloud computing environments. The research focuses on inherent issues of scalability, efficiency, and performance of the cloud systems. To provide an organized understanding of the current landscape, the paper presents an extensive taxonomy classifying (and providing examples) of load balancing algorithms in cloud infrastructures. The paper also proceeds to explore many solutions and approaches to effectively distribute workloads over cloud resources to improve the Quality of Service (QoS) of cloud users.

Selvan, M.A. [11] This paper discussed the challenge of dynamic load balancing in new data-intensive networks. The contribution of this paper is an optimization approach that combines optimization techniques, Genetic Algorithms and Particle Swarm Optimization with multipath routing protocols. By combining these approaches, the paths can reactively respond to the current network conditions and changes in link capacities and traffic demands. In addition, simulation results throughout the paper showed significant improvements in data throughput and total network latency, in light of the growing demands in network environments today.

Elrotub, M. [13] This paper proposed a new task allocation approach exclusively for a cloud computing environment. The major aim of the research was to improve resource usage in a cloud environment as well as lower costs. The implementational approach is classification for VMs (Virtual Machines) as a classification scheme that would simplify the total task allocation process. By adding classification, the expected benefits would be to decrease the search time to find appropriate VMs for incoming tasks and consequently the total process time. The authors believe that by classifying VMs according to their attributes and capabilities they would have a better match on allocating tasks to resources and provide better overall system performance and cost efficiency. Areas of future research were discussed which consisted of evaluating the performance of the proposed approach with all workloads and investigating the use of task prediction methods which would allow for more efficient and proactive task allocation in cloud computing.

III. LOAD BALANCING MODEL

A load balancing model is an approach or system created to allocate tasks uniformly among various computing resources, including servers, network connections, or CPUs. The main objective is to enhance resource use, increase throughput, reduce response time, and avoid any one resource from becoming overwhelmed, thus boosting overall system efficiency and dependability. These models utilize different algorithms to allocate incoming requests or tasks, varying from straightforward static techniques that pre-assign resources to more advanced dynamic methods that adjust based on real-time system circumstances and resource accessibility. Efficient load balancing guarantees high availability, fault tolerance, and a uniform user experience by allocating the processing load and avoiding bottlenecks.

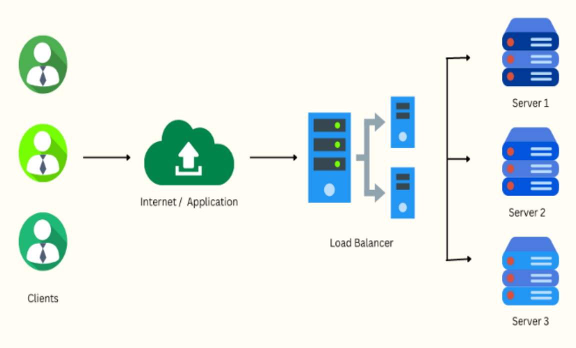

Fig 1 Load Balancing Model

The taxonomy of load balancing algorithm will help to find depth analysis of different load balancing algorithms in cloud computing. Static algorithm uses the current state of node. It will not bother about the previous state of node. Dynamic algorithms use previous as well as current state of node to distribute the load. Research have shown that static and dynamic have not been fully efficient for load balancing. This gave room to hybrid algorithms. Hybrid methods inherit the properties from both static and dynamic load balancing techniques and attempts at overcoming the limitation of both algorithms. Proposed framework indicates the resolving method of overloading using virtualization to avail VM for request implementation which is free or less loaded which reduce throughput time and increase efficiency of execution phase.

Virtualization in load balancing

A load balancing employing virtualization takes advantage of the adaptable and dynamic characteristics of virtual machines (VMs) to effectively allocate workload across computing resources. This method usually entails setting up several VMs that run the application or service. A load balancer smartly allocates incoming requests to these VMs according to established algorithms and the current health and capacity of each VM. Virtualization allows for on-demand scaling by facilitating the quick launch of new VMs to manage heightened load and the removal of VMs when demand falls, enhancing resource efficiency and cost-effectiveness. This model improves application availability and responsiveness by stopping any individual physical server from being overwhelmed and offering redundancy in the event of VM failures. The capability to swiftly allocate and oversee virtual resources renders virtualization a robust basis for creating scalable and resilient load balancing solutions in contemporary computing settings, encompassing cloud and on-site infrastructures. virtualization aims to optimize the use of resources and can have numerous meanings for different people according to its type of usage. Such uses can include the following items such as Server, Storage, Network and Service virtualization [1].

IV. PROPOSED MODEL

This section explains the framework for improvising load balancing using virtualization in cloud computing field. The sole goal of proposed approach is to provide availability of processing node to user for executing task. It avoids system failure due to overloading and optimizing storage issue in cloud using virtualization and migration technique. It creates virtual instances of system for creating replica of a system for the end user which can migrate task to one node to another due to any inconvenience without letting it know to the client.

Top layer deals with requests generated by multiple clients. Data centre which is storage of a cloud which receives request generated by client and pass them to the load balancer. In bottom layer load balancer controller direct user request to available node present in network which is not much pre occupied by prior tasks. If the primary allocated VM is overloaded in that case migration technique comes to rescue which transfer request to another VM. This will conclude that primary VM is overloaded thus it is unavailable for further execution requests from the load balancer controller. The allocation list is also been updated as per the status of VM.

The proposed Methodology uses the replication concepts of virtual Machines which will provide stand by VM or a replica of a VM for a particular module for handling overloading situation due to which controller can divert the traffic of particular module to its dedicated VM node. Which is beneficiary when the management for handling overloading is priorly setup and it also optimized storage usage because only required module replica is only created for stand by rather than whole VM so when the requirement of same is over the replica of a module is turn off or destroy to free the storage for further usage and the rest of the requests for module having no dedicated VM will be assigned to the VM using Weighted Round Robin (WRR) Algorithm for allotment of task to particular node by using weight value of respected incoming request task for optimized solution and faster execution.

Weighted Round Robin Algorithm

Weighted Round Robin (WRR) is a widely used load balancing algorithm in cloud computing that extends the basic Round Robin approach by assigning a weight to each server (or virtual machine, VM) based on its capacity, such as CPU power, memory, or network bandwidth. In WRR, servers with higher weights are selected more frequently to handle incoming requests, proportionally to their capabilities [12]. The algorithm works by maintaining a list of available servers and their associated weights, distributing client requests in a cyclic manner but giving preference to servers with higher weights. For example, if one server has a weight of 3 and another has a weight of 1, the first server will receive three times as many requests as the second. This ensures a more efficient and fair utilization of resources, preventing weaker servers from being overloaded while allowing stronger servers to handle a greater share of the workload [13]. WRR is particularly useful in cloud environments where VMs often have varying resource capacities, and it can be adapted to dynamic environments where server weights may change over time based on performance monitoring or scaling operations. Its simplicity, fairness, and efficiency make WRR a preferred choice for managing load balancing in scalable cloud applications.

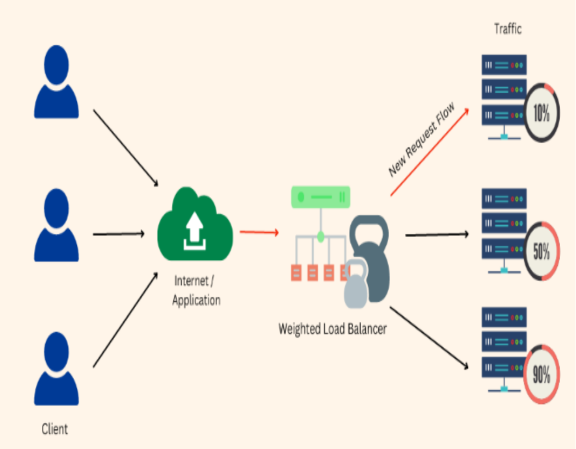

Fig 2 Weighted Round Robin Algorithm Model

Proposed Work Flow Model

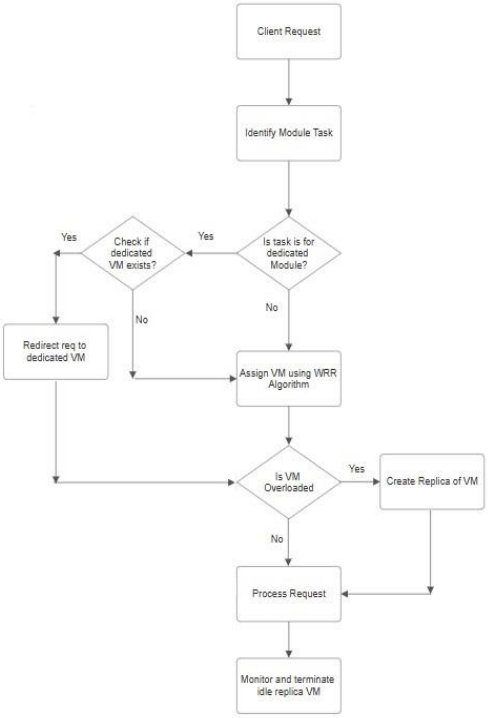
Client Request
Identify Module Task
Is task is for dedicated Module?
Yes
Check if dedicated VM exists?
Yes
No
No
Redirect req to dedicated VM
Assign VM using WRR Algorithm
Is VM Overloaded
Yes
Create Replica of VM
No
Process Request
Monitor and terminate idle replica VM

Fig 3 Work FLow Diagram

The Model flowchart represents a systematic and adaptive framework for the management of client requests within a cloud computing environment, leveraging dynamic virtual machine (VM) orchestration to optimize performance and resource utilization. Upon receipt of a client request, the system conducts an initial analysis to ascertain the corresponding module task and its specific processing requirements. Thereafter, an evaluation is performed to determine whether the task necessitates a dedicated module. In instances where a dedicated module is required, the system verifies the existence of a pre-established dedicated VM; if available, the request is promptly redirected to the appropriate VM, thereby ensuring resource exclusivity and enhanced service performance. In the absence of an existing dedicated VM, or when the task does not assigned dedicated resources, the system employs the Weighted Round Robin (WRR) scheduling algorithm to allocate the request to an appropriate VM, thereby facilitating equitable load distribution in accordance with the computational capacities of the available resources.

Following VM assignment, the system monitors the load status of the selected VM; if an overload condition be detected, a replica VM is instantiated to distribute the computational load and maintain service continuity. Conversely, if no overload is identified, the request proceeds to execution on the initially assigned VM. Subsequent to request processing, the system engages in continuous monitoring of replica VMs, terminating idle instances in a proactive manner to conserve computational resources and minimize operational expenditure.

By combining the WRR and concept of virtualiztion this methodology ensures a scalablity of available resources for optimized performance by creating replica of VMs for migrating task load to other servers due intitial overloaded VM. Its priority is to create VM prior for partcular mode as a dedicated server or a VM so it can directly handle request for respective module to reduce throughput time. It is also space saving solutin because it create VM or replicated VM for particular module rather than duplicating whole server replicas. This is also a cost-effective operational model capable of dynamically adapting to various client demands by using possibleby minimal resources and scaling it for optimized approach and effectivity within cloud-based infrastructures .

Simulation

CloudSim is a powerful and extensible simulation framework developed by the CLOUDS Laboratory at the University of Melbourne for modeling and simulating cloud computing environments. It enables researchers and developers to evaluate the performance of cloud infrastructures, services, and resource management policies without deploying real-world cloud platforms. CloudSim supports the simulation of large-scale data centers, VM provisioning, and customizable resource allocation strategies. Its flexibility and modular design have made it a vital tool for studying load balancing, scheduling, and energy-efficient cloud computing solutions. CloudSim is like building blocks by the use of which you can make your simulation environment. Thus, Cloudsim is not a ready made solution, which you can set the parameters and then collect the results to use in your project[1].

V. RESULT OF SIMULATION

In previous section, we conducted a simulation and documented the resulting scores in a text file (log file) for later analysis. This generated data enabled us to recognize trends and patterns more efficiently. By utilizing the gathered data, we created graphs to represent the results visually, offering clearer insights into the outcomes of the simulation. The images below showcase the graphs generated from the simulation data.

Pie Chart

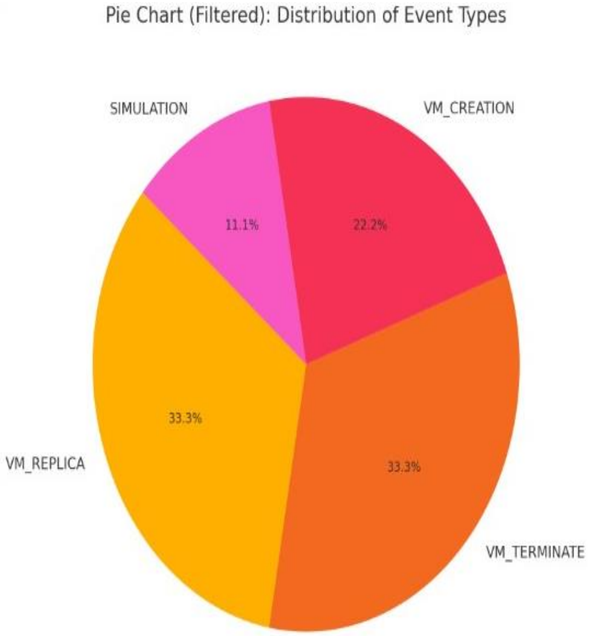

Fig 4 Pie chart

The pie chart represents the distribution of major event types recorded during a CloudSim dynamic simulation process. Each section of the chart shows how frequently different event types occurred during the simulation relative to the others.

The most significant factor VM_CREATION of 22.2% represents creation of in initial VM at beginning of simulation. VM_REPLICA of 33.3% represents creation of dynamic VM replicas due to overloading to maintain load balancing. VM_TERMINATION of 33.3% refers to termination of idle dynamic replica VMs to reduce cost and optimize resource utilization. SIMULATION part having 11.1% includes initializing and wrapping up of simulation environment.

Bar Graph

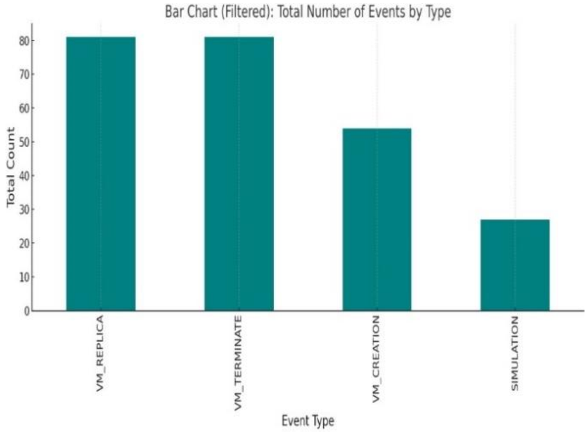

Fig 5 Bar Graph

The bar chart represents the total count of different event types after filtering.VM_REPLICA and VM_TERMINATE events are the most frequent, both having the highest total counts (around 80+ events each).VM_CREATION follows with a moderate number of occurrences (approximately 55 events).SIMULATION events have the lowest count (around 27 events), indicating fewer simulation lifecycle operations compared to VM management activities.The chart highlights that virtual machine operations (replication and termination) dominate the system's activities, suggesting a dynamic environment with frequent VM scaling and shutdowns.

Overall, VM resource management (replica and terminate) is prioritized over the creation of new VMs and simulation-level activities.

Heat Map

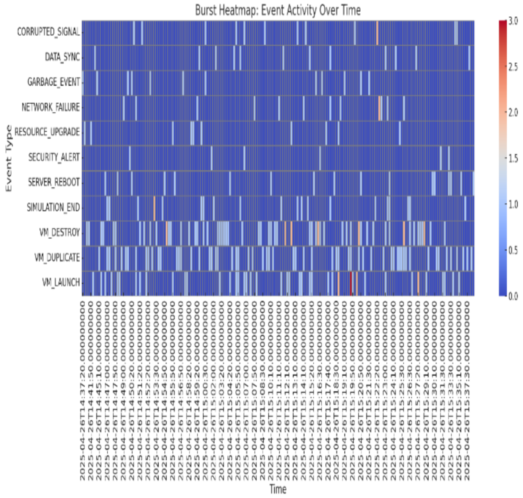

Fig 6 Heat Map

The heatmap gives visualization of intensity of activity of different types of event over time during simulation process. Here X-axis represents Timeline(Date and Time stamps) and Y-axis represents different types of events which takes place in entire simulation process. Color scale denotes the intensity of activity as dark blue(0.0) shows null activity, light blue (upto 3.0) represents increasing intensity of activity and red is for peak bursts.

According to heatmap we can see that activity like VM_LAUNCH, VM_DUPLICATE, and VM_DESTROY show high activity compared to other events by frequenty changing color from blue to light blue shades.Sharp red spots are observed around VM_LAUNCH and VM_DUPLICATE, meaning that massive bursts of VM launching and duplication happened during those periods.A few spikes also occur for VM_DESTROY, suggesting rapid de-provisioning.SIMULATION_END, SERVER_REBOOT, SECURITY_ALERT, RESOURCE_UPGRADE, NETWORK_FAILURE, DATA_SYNC, and CORRUPTED_SIGNAL show sparse activity.These events appear mostly dark blue, indicating low frequency or sporadic occurrence.

The simulation is dynamic and reactive, adapting to load changes by scaling VMs up or down.Workload fluctuations are mainly handled through VM operations rather than server-wide resets or upgrades. Bursts occur randomly but are more common during mid and later stages of the simulation timeline.

VII. CONCLUSION

In this research, an advance load balancing model using virtualization and dynamic task migration was presented to solve the ongoing issues of resource optimization and service continuity in cloud computing systems. Through the combination of virtual machine replication and the Weighted Round Robin (WRR) algorithm, the model provides fair task allocation, reduces system response time, and improves the effective utilization of computational resources. Simulation findings gleaned through CloudSim support the efficacy of the solution, reflecting significant improvements in system scalability, fault tolerance, and cost of operation reduction. Additionally, dynamic instantiation and shutdown of module-specific VM replicas help towards a space-saving and cost-saving mode of operation. This study proves that virtualization, in conjunction with strategic use of smart load balancing algorithms, has the capability to greatly improve the performance of cloud infrastructure to the increasing needs of high availability and quality service provision. The model can be fine-tuned with predictive analytics and AI-driven load forecasting in future studies to add more strength to adaptive resource management in dynamic clouds.

REFERENCES

[1] Sefati, S., Mousavinasab, M., & Zareh Farkhady, R. (2022). Load balancing in cloud computing environment using the grey wolf optimization algorithm based on the reliability: performance evaluation. The Journal of Supercomputing, 78(1), 18-42.

[2] Shahid, M. A., Alam, M. M., & Su’ud, M. M. (2023). Performance evaluation of load-balancing algorithms with different service broker policies for cloud computing. Applied Sciences, 13(3), 1586.

[3] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences, 34(7), 3910-3933.

[4] Ghomi, E. J., Rahmani, A. M., & Qader, N. N. (2017). Load-balancing algorithms in cloud computing: A survey. Journal of Network and Computer Applications, 88, 50-71.

[5] Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing–A hierarchical taxonomical classification. Journal of Cloud Computing, 8(1), 1-24.

[6] Laha, J., Pattnaik, S., & Chaudhury, K. S. (2024). Dynamic Load Balancing in Cloud Computing: A Review and a Novel Approach. EAI Endorsed Transactions on Internet of Things, 10.

[7] Lohumi, Y., Gangodkar, D., Srivastava, P., Khan, M. Z., Alahmadi, A., & Alahmadi, A. H. (2023). Load Balancing in Cloud Environment: A State-of-the-Art Review. IEEE Access, 11, 134517-134530.

[8] Kulkarni, M., Deshpande, P., Nalbalwar, S., & Nandgaonkar, A. (2022). Taxonomy of Load Balancing Practices in the Cloud Computing

Paradigm. International Journal of Information Retrieval Research (IJIRR), 12(3),300292

[9] Ijeoma, C. C., Samuel, A., Okechukwu, O. M., & Chinedu, A. D. (2022). Review of hybrid load balancing algorithms in cloud computing environment. arXiv preprint arXiv:2202.13181.

[10] Sasidhar, T., Havisha, V., Koushik, S., Deep, M., & Reddy, V. (2016). Load Balancing Techniques for Efficient Traffic Management in Cloud Environment. International Journal of Electrical and Computer Engineering (IJECE), 6(3), 963-973.

[11] Selvan, M. A. (2024). Multipath Routing Optimization for Enhanced Load Balancing in Data-Heavy Networks.

[12] Almhanna, M. S., Murshedi, T. A., Al-Salih, A. M., & Almuttairi, R. M. (2024). Dynamic Allocation of Weights Using the Minimally Connected Method to Augment Load Equilibrium in Distributed Systems. International Journal of Intelligent Engineering & Systems, 17(1).

[13] Elrotub, M., & Gherbi, A. (2018). Virtual machine classification-based approach to enhanced workload balancing for cloud computing applications. Procedia computer science, 130, 683-688.