Analysing Load Balancing Algorithms

Akbar Khan, Niyati Soni

JK Lakshmipat University Jaipur 302026 India

ARTICLE INFO

Keywords:
Static Algorithms
Dynamic Algorithms
Cloud Computing Metrics

ABSTRACT

Cloud computing is the on-demand availability of computer system resources over the internet. Load balancing helps in segregating the traffic among different servers to increase the efficiency of the system. This paper addresses various load-balancing methods categorized into static algorithms and dynamic algorithms. A comparison has been performed between all the algorithms discussing the advantages and disadvantages. Cloud Computing metrics have been addressed which explains the different parameters associated with load balancing.

1. Introduction

Cloud computing is the on-demand delivery of computing resources as services over the internet. Businesses only pay for the resources they use rather than having to purchase, install, or maintain the resources themselves. It helps with the administration of vast amounts of data and enhances resource management. When we create apps, clients send requests for information access or application access. We need load balancers in the modern world to efficiently handle this kind of traffic on resources when several clients are concurrently accessing the resource and sending important requests to the backend. As "traffic controllers," load balancers decide which server is active and able to respond to requests promptly while allocating loads to various servers. In order to reduce latency, the load is transferred to the closest server to the client.

Numerous research have examined task scheduling and load balancing in cloud systems. We start by listing a few current publications that look at load balancing strategies and methods in cloud environments:

- Ghutke et al. (2014) examined numerous strategies that are applied to load balancing and work scheduling issues in cloud computing. To provide readers a thorough understanding of the most current advancements in the industry, they also examined the advantages and disadvantages of the algorithms.

- Mesbahi and Rahmani (2016) examined the most recent load-balancing techniques as well as the requirements and elements that need to be considered while developing and putting into practise appropriate cloud environment load-balancing techniques. In addition to presenting a brand-new taxonomy of load balancing solutions, they also assessed them using pertinent metrics and explored both their benefits and drawbacks. They also found that energy saving takes precedence over other factors in contemporary load balancing techniques. However, because there aren't any simulator tools that can mimic load balancing solutions, their study is limited. Additionally, there isn't any mention of unanswered questions or potential research topics.

- Deepa et al(2017) .'s goal was to offer a thorough analysis of load balancing techniques currently used in cloud computing.

- Shahbaz et al2019.'s presentation included a comprehensive, encyclopaedic overview of load balancing strategies. The merits and disadvantages of existing approaches are highlighted, and significant issues are addressed in order to build effective load balancing algorithms in the future. The study offers fresh perspectives on load balancing in cloud computing.

We conducted a literature survey and an analysis of state-of-the-art methods for future load balancing researchers will benefit from this information as they create new processes and algorithms. This study's goal is to examine alternative methodologies, outline their characteristics, and discuss both their benefits and drawbacks. To offer recommendations to upcoming scholars in the field of load balancing for creating new algorithms and mechanisms, we did a literature review and an analysis of cutting-edge techniques. This study's goal is to examine alternative methodologies, outline their characteristics, and discuss both their benefits and drawbacks. The primary goals of this essay are as follows:

- Analysing the load-balancing algorithms in use.

- Explicitly stating the benefits and drawbacks of the load-balancing strategies in each class.

- Identifying the principal domains in which these static and dynamic algorithms can be used.

Understanding the various load balancer types and approaches that are available to manage the heavy load is made easier by this study. We made a distinction between static and dynamic load balancing systems based on a variety of variables, including reaction time, performance, and many more. We can better understand the circumstances in which a particular load balancing strategy is advantageous after looking at these algorithms.

The remaining sections of the article are arranged as follows. Background information on load balancing and its features are provided in Section 2. In Section 3, various load balancer types are described according to their features and

ORCID(s):

setups. In Section 4, a very extensive comparison of the dynamic and static load balancing techniques currently in use is given. Using the load balancing algorithm taxonomy as a guide, Section 5 offers a comparison of the methods outlined previously. Section 6 discusses a condensed method for applying a specific algorithm to a specific case. Part 7 brings our survey to a close and offers ideas for new subjects.

2. Background

In early days of computing, a mainframe computer was the sole device available and it was used for all computing operations. With the development of computer technology came the idea of distributed computing and the requirement to share the workload among numerous computing resources. In the beginning, this was performed by manually allocating jobs to various resources according to their availability and burden.

With an increase in users and consumers, more people are using the internet daily. The demand for rapid answers, great scalability, and fast processing is increasing as a result of this increase. Users in the millions simultaneously seek data or media. We need a system that can operate as a "traffic controller" and route the requests to the most competent and efficient servers so that we can answer to these requests rapidly and effectively. These servers must be capable of processing the volume of queries without getting overworked and performing poorly.

2.1. Opportunities

The challenge faced by people providing information or developing applications was to efficiently manage the traffic on their servers, so that their clients don't feel any unrequired delay in responses as well as don't get affected by server failure. Due to this circumstance, load balancing was implemented. It divides the client requests among the servers and determines which server can best handle the request, while evaluating the server's ability to handle requests and if necessary, removing unreliable servers until it is fixed. As a result, clients have an overall good user experience.

2.2. The load balancing functioning

After receiving requests from users, the requests are distributed among the Virtual Machines, the load balancer uses load-balancing algorithms (VMs).Which VM should be assigned to the subsequent request is determined by the load balancer. In an effort to correctly provide services to customers, numerous judgements are taken in order to evaluate alternative execution strategies. In this case, load balancers let servers move data effectively while also controlling data flow between servers and endpoints. Load balancers only route traffic to servers that are unable to handle requests by keeping track of the condition of backend resources. A load balancer, which might be hardware or software, uses one or more algorithms to split traffic among numerous web servers. In other words, it evenly distributes loads among a number of servers to prevent any one server from becoming overloaded and hence unreliable.

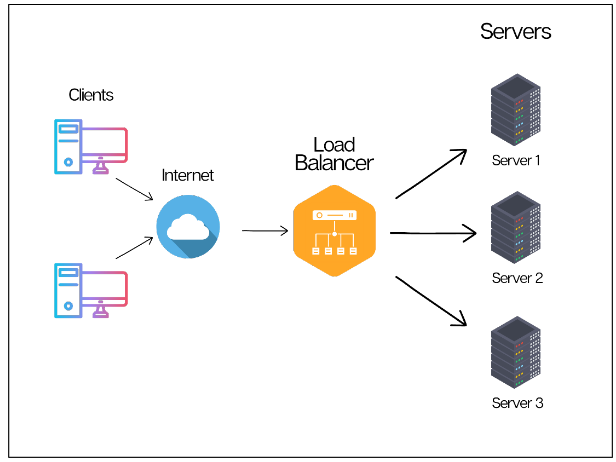

Figure 1: The load balancing functioning

In today's multi-device, multi-app workflows, load balancing is the most scalable method for sustaining the many web-based services. Load balancing collaborates with platforms that make it easier for users to access the numerous desktops and programmes offered in today's digital workplaces in order to provide customers with a more reliable and consistent end-user experience. According to researchers, load balancing intercepts requests sent by users, selects the node that will receive and process them, and then splits the load among numerous nodes. The load balancer then keeps track of the nodes that are available, checks to see if demand is being met, and removes any that are not from rotation if necessary. In the event of a breakdown, load balancers set up more units to provide redundancy. The functions and configurations of load balancers can be categorised in order to manage such sophisticated processes.

3. Types of Load Balancers

3.1. Based on Functions

Traffic is divided among the OSI layer's many layers by load balancers. Traffic is distributed at Layer 4 of the OSI model by network load balancers. It does not take into account application-level information like the type of content and instead operates based on network variables like IP address and destination ports. Next, Application Load Balancers analyse data from a much larger range of online application protocols, such as HTTP headers, FTP, SMTP, DNS, and SSL sessions, by operating on Layer 7. For network health checking, server monitoring, network traffic optimisation, minification, and caching, it manages the server traffic depending on individual usage and behaviour. A global server load balancer is used to manage several cloud data centres that are spread out worldwide. Due to the ability to access other data centres around the globe for business continuity, it also enables speedy recovery and flawless company operations in the event of server failure or disaster at the data centre.

3.2. Based on Configurations

Load Balancers can be distinguished based on its configuration to identify whether it is in software, hardware or even virtual form. Hardware-based load balancing solutions first appeared in the 1980s. To split the burden among several servers, these solutions made use of specialised hardware. Large enterprises with high-end computing needs were the only ones who could deploy these devices because they were frequently expensive and difficult to manage. The demand for load balancing solutions increased dramatically with the development of the internet. Software-based load balancing solutions started to appear in the middle of the 1990s. To disperse the burden among several servers, these solutions made use of software algorithms. They quickly grew in popularity because they were more affordable and simpler to administer than hardware-based alternatives. Next came the invention of virtual load balancers, which are hardware load balancers operating on a virtual system. It has weaker scalability, no central management, and no automation.

4. Types of load balancing Algorithms

With the aid of various load balancing strategies that choose servers depending on a certain configuration, load balancers can distribute requests to specified servers.

4.1. Static Load Balancing Algorithms

Regardless of the condition of the backend server handling the requests, these algorithms operate in the same way. Although they are easier and more effective to execute, they can result in an unequal allocation of requests. By giving the system previous knowledge, load balancing is accomplished in this method. They are non-preemptive, which means that once the load is assigned to a node, it cannot be moved to another node. This technique takes less time to execute because it requires less communication. The system's current status is not considered when allocating resources, which is a major negative.

4.1.1. Round Robin Algorithm

A common scheduling algorithm in cloud computing is round-robin. It is a preemptive scheduling algorithm, which means that it pauses the execution of one process or task to give resources to another. This algorithm's fundamental principle is to distribute resources equally and effectively among many processes or activities. A set time quantum, or the maximum length of time a process or task can run before being preempted, is given to each process or task in the round-robin algorithm. After that, the processes or tasks are arranged in a circular queue, with the first one receiving CPU time.

When a process or task's time quantum runs out, it is preempted and the subsequent process or task in the queue is given CPU time. As the time quantum is constant for all processes or tasks, the round-robin algorithm makes sure that each one receives an equitable share of CPU time. By preventing any one of them from using up all of the CPU resources, this makes sure that all activities and tasks are

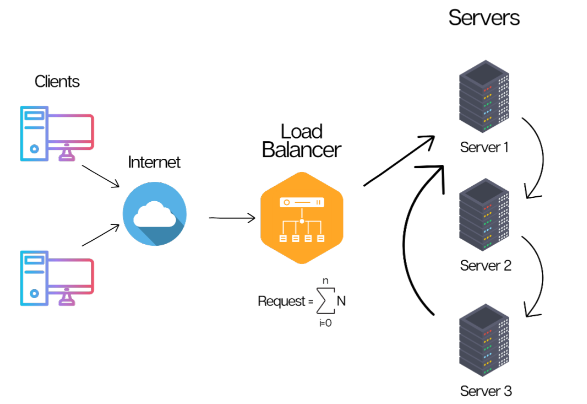

Figure 2: Round Robin Algorithm

completed quickly and successfully. It is also very scalable, which makes it appropriate for cloud computing situations where numerous processes or tasks must be planned and carried out concurrently. The round-robin algorithm's ease of use and lack of need for complicated algorithms or heuristics are two of its main features.

By using this method, a number of servers receive the client requests. Each server receives a client request, which is then passed forward. The load balancer is instructed by the algorithm to go back to the top of the queue and repeat. When the servers' processing power and storage capacity are roughly equal, the round robin server load balancing process produces the best results.

Assume that an organisation has a cluster of three servers consisting of Server A, Server B, and Server C.

Server A receives the initial request.

Server B receives the second request.

Server C receives the third request.

Based on this order, the load balancer keeps sending requests to the servers. In order to handle heavy traffic, this makes sure that the server load is divided equally.

4.1.2. Central Manager Algorithm

In cloud computing systems, the Central Manager Algorithm is a common resource management technique. The fundamental concept behind this algorithm is to have a centralised manager who is in charge of managing resource allocation and de-allocation to various users and cloud apps. In this technique, the central management keeps track of the many resources that are at their disposal, including CPU, memory, storage, and network bandwidth. When a user or application asks resources, the central manager determines whether they are available and assigns them to the user or programme who requested them. The central manager method also permits dynamic resource allocation, wherein resources may be added or subtracted in accordance with the workload and demand at any given time. By doing this, it is ensured that resources are used effectively and are not wasted.

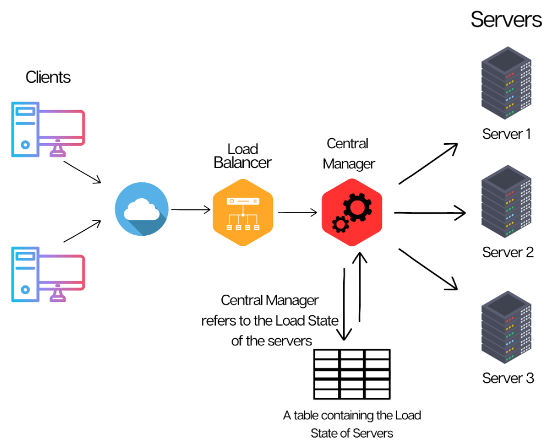

Figure 3: Central Manager Algorithm

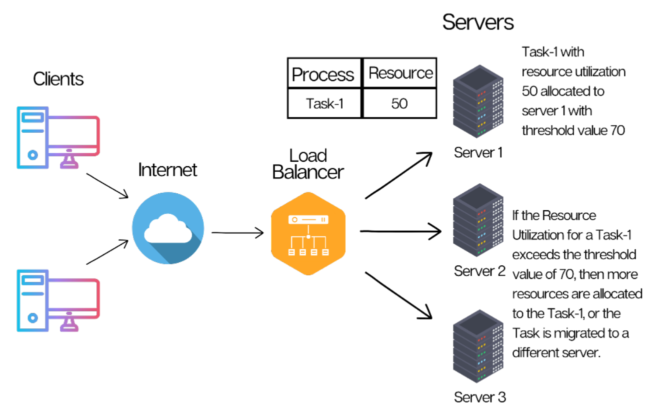

Figure 4: Threshold Algorithm

The fact that the central manager algorithm offers centralised control over the resources, making it simpler to operate and maintain, is one of its fundamental advantages. It also enables efficient resource allocation because resources can be dynamically distributed in accordance with demand. The central manager algorithm, in sum, is a helpful resource management technique in cloud computing systems. It offers centralised control over resources, enables effective resource use, and supports dynamic resource allocation.

With the goal of maximising resource usage and reducing response time, it divides incoming traffic among several servers. The algorithm is composed of a central management (CM) node that receives incoming requests and distributes them to accessible worker nodes according on their load state. This makes sure that all available worker nodes are utilised to their full potential and that incoming traffic is split equally among them. Although the CMA is a straightforward and efficient load-balancing algorithm for cloud computing environments, it does have certain drawbacks, such as relying on a single CM node that might fail.

4.1.3. Threshold Algorithm

In cloud computing, the threshold algorithm is a method for distributing workloads among several servers or network nodes. The algorithm's goal is to establish a threshold value that indicates when extra resources should be dedicated to a particular job or process.

Each node in the network checks its own workload as part of the threshold algorithm and compares it to the threshold value. The node asks the cloud service provider for more resources if the workload exceeds the threshold. Next, in order to aid the node in finishing its mission, the provider allots extra resources to it, such as processing power or memory. The threshold method has the benefit of automatically scaling resources in response to demand. It is cheaper and more efficient to consume resources as the workload falls since nodes can return unused resources to the provider. The threshold algorithm also guarantees that tasks are completed successfully and effectively by distributing resources as needed to handle increasing workloads. This might result in enhanced performance and faster processing times. The threshold method is a crucial tool for workload management in cloud computing environments since it helps to cut expenses, increase performance, and optimise resource utilisation.

4.1.4. Randomized Algorithm

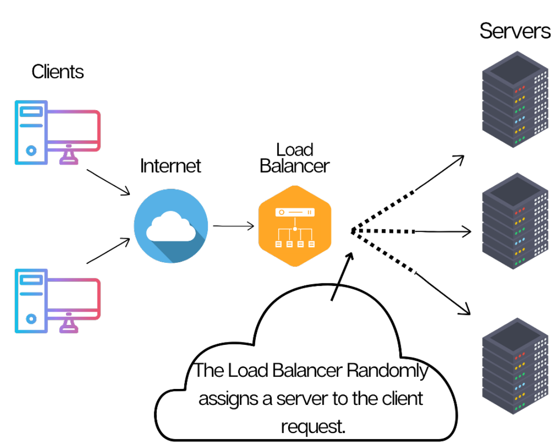

Figure 5: Randomized Algorithm

In cloud computing, randomization is a type of technique that is used to address issues or optimise resource allocation. These algorithms employ probabilistic techniques, which entail producing random numbers or making random decisions in order to address a specific issue. Load balancing is one application of a randomised algorithm in cloud computing. The purpose of load balancing is to divide workloads among several servers so that no one server is overworked. An algorithm that uses random selection or a probability distribution can be used to choose which server to send a particular work to.

Resource allocation is another application of a randomised algorithm in cloud computing. In this situation, resource allocation to various tasks or processes might be based on probabilistic calculations or random selection using randomization. A randomised algorithm, for instance, can be used to distribute memory or processing capacity among several nodes according to their workload or other considerations.

4.2. Dynamic Load Balancing Algorithms

While distributing requests, these algorithms take into account the backend server's condition and server load. They are a little more difficult yet effectively distribute requests since they need communication between the load balancers and servers. It relies on three strategies: the transfer strategy, which aids in determining which tasks should be transferred to other nodes for processing and are eligible for transfer; the location strategy, which chooses a node to carry out the transferred task; and the information strategy, which provides each node with all pertinent information, including the location and transfer strategies.

4.2.1. Central Queue Algorithm

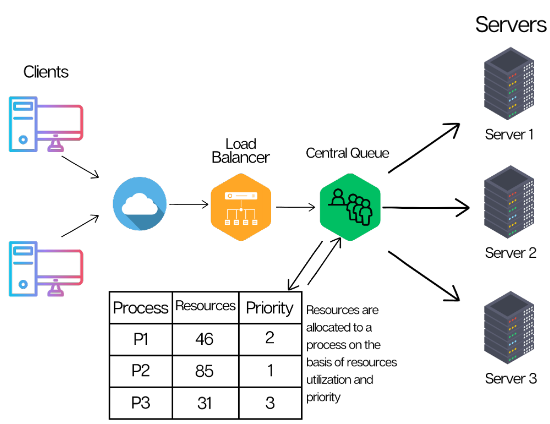

Figure 6: Central Queue Algorithm

In order to maximise resource allocation and usage, the Central Queue Algorithm is a resource management technique used in cloud computing settings. The fundamental concept behind this method is to have a central queue that controls how resources are distributed among many tasks or processes. In the Central Queue Algorithm, each job or process is added to a centralised queue, and resources are given to each task or process based on the priority and demand. A scheduler controls the queue and decides the distribution of resources and the sequence in which activities or processes are carried out. The Central Queue Algorithm's ability to efficiently utilise resources by allocating resources in accordance with actual demand and priority is one of its main features. This helps avoid over-provisioning of resources, which can result in resource waste and higher expenses. The Central Queue Algorithm additionally enables effective resource allocation and usage for batch activities or lengthy processes that need high resource use over a prolonged period of time. The queue makes ensuring that activities or processes are carried out in the order they were received, resulting in a more equitable workload distribution and avoiding resource depletion for low-priority operations.

In conclusion, the Central Queue Algorithm is a valuable resource management method in cloud computing environments, particularly for batch operations or long-running processes that need significant resource use over an extended period of time. To minimise the overhead and latency connected with the centralised queue, however, and to verify that the algorithm is acceptable for the specific requirements of the application, careful design and monitoring are required.

4.2.2. Local Queue Algorithm

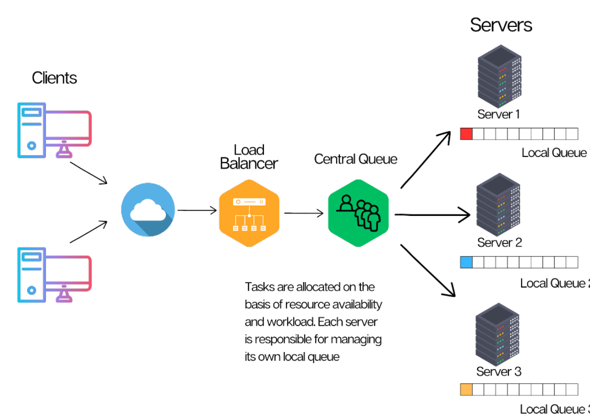

Figure 7: Local Queue Algorithm

In order to maximise resource allocation and usage, the Local Queue Algorithm is a resource management technique used in cloud computing settings. The fundamental concept behind this technique is the use of a local queue to control how resources are distributed among various tasks or processes on each server or virtual machine. In the Local Queue Algorithm, each server or virtual machine has its own local queue, and resources are assigned to each task or process based on the priority and demand within the local queue. Each server or virtual machine controls its own queue, decides how tasks or processes are carried out, and decides how resources are distributed. The Local Queue Algorithm's ability to efficiently utilise resources by allocating resources in accordance with actual demand and priority within each server or virtual machine is one of its main advantages. This helps prevent resource overprovisioning, which can lead to resource waste and greater costs. Moreover, the Local Queue Algorithm offers efficient resource allocation and utilisation for applications that demand low latency or real-time processing. By guaranteeing that actions or processes are completed quickly and with little overhead thanks to the local queue, resources are used more efficiently. In conclusion, the Local Queue Algorithm is a useful resource management technique in cloud computing environments, especially for applications that demand low latency or real-time processing. To ensure that the algorithm is appropriate for the particular requirements of the application, however, and to avoid resource starvation on servers or virtual machines that are under a lot of load, careful planning and monitoring are required.

The local queue algorithm provides several advantages over the central queue algorithm. Firstly, it reduces the communication overhead between nodes in the cloud system since each node operates independently. Secondly, it improves the fault-tolerance of the system since a failure in one node only affects the tasks in its local queue. Finally, it allows for better resource allocation since each node can optimize its resource usage based on its workload and available resources.

4.2.3. Least Connection Algorithm

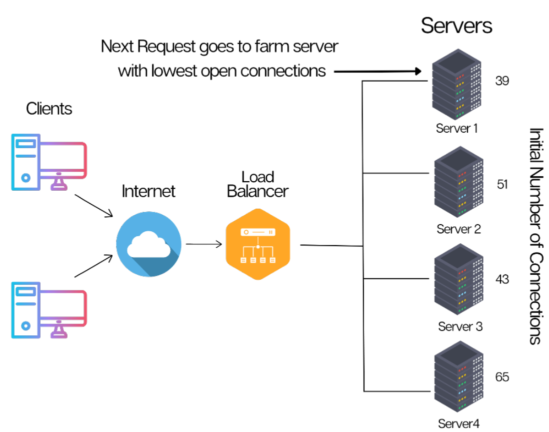

Figure 8: Least Connection Algorithm

In cloud computing environments, the Least Connection Algorithm is a load balancing technique used to split up incoming network traffic among several servers or virtual machines. The fundamental idea behind this method is to route new traffic to the server that currently has the fewest active connections. The load balancer continuously tracks each server's number of active connections while using the Least Connection Algorithm, and it routes new incoming connections to the server that has the fewest connections at any given time. This ensures that the server with the lowest load receives new incoming connections, leading to a more balanced distribution of network traffic among all servers. While servers are distributed based on actual demand, one of the Least Connection Algorithm's main benefits is that it enables effective resource usage. New connections are routed to less busy servers when a server is overloaded, resulting in a more equitable allocation of burden across all servers. By doing this, server overload is avoided, which can result in higher latency and worse performance.

In cloud computing contexts, the Least Connection Algorithm is a helpful load balancing tool. It enables effective resource management and avoids server overload. To minimise the overhead and latency associated with continuous monitoring, however, and to make sure the algorithm is appropriate for the application's unique requirements, careful design and monitoring are required.

A dynamic load balancing mechanism called least connection load balancing distributes client requests to the application server that is currently hosting the fewest active connections. This method considers the load of active connections. When application servers have the same specifications, longer-lasting connections may cause one server to become overloaded. This method works well with incoming requests that have a range of connection times and a group of servers that have resources and processing speeds that are similar.

4.3. Comparison of different load balancing algorithms

The static and dynamic load balancing algorithms have been carefully examined and analysed. The table below summarises our findings in terms of the benefits and drawbacks of different algorithms. The names of the algorithms that scholars have developed, together with benefits and drawbacks, are listed in the analysis table.

S.No.	Environment	Algorithms	Advantages	Disadvantages
1	Decentralized	Round Robin Algorithm	It is straightforward and simple to apply. The servers that are accessible split up all requests evenly	When the jobs have different execution times, it cannot produce the desired results
2	Centralized	Central Manager Algorithm	It functions best when each host generates its own dynamic activities	A bottleneck situation may result from the high level of interprocess communication required by this technique
3	Decentralized	Threshold Algorithm	There is minimal remote memory access	It is difficult to compare each node to the stated threshold because of its complexity
4	Decentralized	Randomized Algorithm	It is easy to put into practise	One server may get overloaded while others go underused as a result
5	Centralized	Central Queue Algorithm	Advantageous for heterogeneous nodes	Less resources are being used
6	Decentralized	Local Queue Algorithm	This method can be modified to meet the unique requirements of the system. It is more scalable and better at utilising resources	It distributes the weight unevenly and is highly dependent on the load's properties. Implementing it is difficult
7	Decentralized	Least Connection Algorithm	Advantageous for homogenous nodes.	Not helpful for projects of various duration

Table 1
Comparison of Load Balancing Algorithms

5. Comparison on the basis of parameters

We go over the cloud computing load balancing metrics in this section. Many load-balancing methods have been proposed by researchers, as was previously indicated. The following is a summary of the metrics that have been proposed in the literature on load balancing for use with load-balancing algorithms:

- Nature: This component has to do with understanding the nature or behaviour of load balancing algorithms, specifically whether the algorithm is static or dynamic, pre-planned or not.
- Overhead: It calculates the amount of overhead involved, such as that brought on by workload relocation, interprocessor communication, and inter-process communication. If it is high, the algorithm has a higher overhead cost and is therefore less efficient than one with a lower overhead.
- Throughput: It is used to determine how many tasks have been accomplished in terms of execution. For optimal performance, it ought to be high.

- ProcessMigration: This parameter shows whether or not the load-balancing system's nodes transfer work load among themselves. It gives details about the moment a system decides to transfer a workload. It determines whether a process may be handled locally or if a remote system node must handle it. If it is present, the algorithm performs better.
- ResponseTime: This parameter shows how long the load-balancing approach takes to adjust to how much work is distributed among the system's nodes. The reaction time of static load balancing algorithms is lower. Algorithms for dynamic load balancing could have a longer reaction time. lower is preferable.
- ResourceUtilization: This parameter shows how much of the server node's resources are being used. Resource usage can be used to accomplish automatic load balancing. There may be more processes in a dispersed system, which would necessitate greater processing power. Resources can be sent to underloaded processors more effectively if the load-balancing algorithm is able to make use of the available resources. Static load balancing methods only assign tasks to processors in an effort to minimise response time, ignoring the possibility that using this task assignment may result in a situation in which some processors finish their work early and sit idle due to a lack of work. As a result, static load balancing algorithms use fewer resources. Dynamic load balancing techniques make better use of resources since they ensure that loads are distributed evenly across all processors and that none of them are left idle.
- FaultTolerant: This value serves as a measure for the load balancing technique's fault-tolerance capacity. With the help of this feature, an algorithm can keep functioning normally even in the case of a failure.
- WaitingTime: This parameter specifies the amount of time in the ready queue. Less waiting time means greater performance.
- Scalability: This characteristic is described as the capacity of an algorithm to produce optimum results with any limited number of nodes. The higher the better.
- Performance: This parameter is used to assess the system's effectiveness.

5.1. Differentiating algorithms on the basis of Load Balancing metrics

Above-described parameters help in realizing which algorithm can perform better in which area and how an algorithm performs in different scenarios. Here we thoroughly investigated and analysed the static and dynamic load balancing algorithms on the basis of metrics. Our observations in the form of low, high or medium and yes or no for presence of a metric in various algorithms. The analysis table contains the names of the algorithms proposed by researchers and their performance metrics on different parameters if they posses a metric and if yes then in which intensity.

h

Techniques	Round Robin	Central Manager	Threshold Algorithm	Randomized Algorithm	Central Queue	Local Queue	Least Connection
Nature	Static	Static	Static	Static	Dynamic	Dynamic	Dynamic
Overhead	Low	High	Low	Low	High	High	High
Throughput	Low	Low	Low	High	High	High	High
Process Migration	No	No	No	No	No	Yes	No
Response Time	Less	Less	Less	Less	More	More	Less
Resource Utilization	Less	Less	Less	Less	Less	More	More
Fault Tolerence	No	Yes	No	No	Yes	Yes	No
Process Migration	More	More	More	More	Less	Less	Less
Scalability	High	Low	High	Less	Low	High	High
Performance	Low	Low	Low	Less	High	High	High

Table 2
Metric Based Comparison of Load Balancing Algorithms

6. Circumstances to use a particular algorithm

Till now we saw, there are static and dynamic load balancing algorithms. Static algorithms are designed for system with low fluctuation in incoming load, where traffic is divided equally among the servers and requires deeper information about available system resources. It does not require real-time communication with the servers. The allocated load cannot be retransferred to other servers during runtime. Dynamic algorithms are designed for the system with high fluctuation in incoming load, where traffic is dynamically divided among the servers. It does not necessarily need deeper information about system resources beforehand. It requires real-time communication actively with the servers. The allocated load can be retransferred among servers to reduce the underutilization of resources.

After differentiating all algorithms on the basis of above-mentioned parameters, we concluded the scenarios in which these algorithms can be used which is explained in below table.

Algorithms	Circumstances where algorithm fit in.
Round Robin	Can be when there are multiple servers available having similar configurations having low latency requirement.
Central Manager	Can be used when there are large number of requests from client side, dynamic and unpredictable workloads and require high availability.
Threshold Algorithm	Can be used when the system require scalability and it can have high traffic and has dynamic workloads and require resource utilization.
Randomized Algorithm	Can be used when we have limited information about incoming requests. Can be used in Testing and debugging environments. Can be used in distributed systems, where multiple servers may be responsible for handling different parts of a larger workload
Central Queue	Can be used when there are huge number of servers available and unpredictable workload. Can be used in environments where there are multiple types of requests and have to handle according to priority. Can be used in system having limited resources, so proper resource utilisation is required and fault-tolerance is required.
Local Queue	When there are many nodes in the distributed system and each node has roughly equal resources, this algorithm can be used. When the workload is dynamic and unpredictable, with bursts of activity that require quick resource allocation and large number of tasks can be categorized into smaller ones and scalability and high availability is required, we can use this algorithm.
Least Connection	It can be used when there are large number of incoming connections having resource intensive tasks. Can be used when there are many asynchronous requests that require different levels of processing time and scalability is required.

Table 3
Best Suited Circumstances for the Algorithms

7. Conclusion and Future Work

This paper includes a research of load balancing algorithms along with a comparative analysis of the aforementioned load balancing algorithms in cloud computing with respect to stability, resource utilisation, static or dynamic, scalability, and process migration.

After thorough inspection, it has been shown that the load balancing method assumes that workload is dispersed according to whether it is being compiled or run. Depending on the compile or run time, it could be static or dynamic. Static algorithms are more stable than dynamic algorithms and their behaviour can also be predicted with ease. Dynamic algorithms perform better in dispersed environments. Future plans call for the implementation of a revolutionary dynamic load balancing algorithm. It is also crucial to give a simulation model to assess the parameters or components in order to address the random selection-based load distribution problem.

8. Open Research Questions

The use of load balancing in microservices architectures is still a topic of active research despite its rising popularity. There are some future research questions:

Microservices architecture, or simply "microservices," is an architectural design strategy for developing applications. With the use of microservices, a complicated application can be broken down into smaller, independent parts, each with its own set of responsibilities. A microservices-based application may make use of several internal microservices to construct its response to one user request. How can load balancing be made more efficient in microservices architectures? Next aspect of research can be in edge computing. In order to increase speed and decrease latency, edge computing involves distributing processing and storage resources closer to the end user. Load balancing in edge computing is difficult, though, because the resources are scattered. How can load balancing be made more effective in edge computing settings? Hybrid cloud setups combine resources from the private and public clouds, which can make load balancing more difficult. How can load balancing be improved in these settings to guarantee the best performance and resource use?

References

[1] S. Aslam, M. A. Shah, Load balancing algorithms in cloud computing: A survey of modern techniques, in: 2015 National software engineering conference (NSEC), IEEE, 2015, pp. 30–35.

[2] S. K. Mishra, B. Sahoo, P. P. Parida, Load balancing in cloud computing: a big picture, Journal of King Saud University-Computer and Information Sciences 32 (2020) 149–158.

[3] P. Kumar, R. Kumar, Issues and challenges of load balancing techniques in cloud computing: A survey, ACM Computing Surveys (CSUR) 51 (2019) 1–35.

[4] M. Mesbahi, A. M. Rahmani, Load balancing in cloud computing: a state of the art survey, Int. J. Mod. Educ. Comput. Sci 8 (2016) 64.

[5] S. Afzal, G. Kavitha, Load balancing in cloud computing–a hierarchical taxonomical classification, Journal of Cloud Computing 8 (2019) 22.

[6] J. M. Galloway, K. L. Smith, S. S. Vrbsky, Power aware load balancing for cloud computing, in: proceedings of the world congress on engineering and computer science, volume 1, 2011, pp. 19–21.

[7] N. J. Kansal, I. Chana, Cloud load balancing techniques: A step towards green computing, IJCSI International Journal of Computer Science Issues 9 (2012) 238–246.

[8] B. Ghutke, U. Shrawankar, Pros and cons of load balancing algorithms for cloud computing, in: 2014 International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2014, pp. 123–127.

[9] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, M. A. Alzain, A load balancing algorithm for the data centres to optimize cloud computing applications, IEEE Access 9 (2021) 41731–41744.

[10] A. Dave, B. Patel, G. Bhatt, Y. Vora, Load balancing in cloud computing using particle swarm optimization on xen server, in: 2017 Nirma University International Conference on Engineering (NUiCONE), IEEE, 2017, pp. 1–6.

[11] S. S. Moharana, R. D. Ramesh, D. Powar, Analysis of load balancers in cloud computing, International Journal of Computer Science and Engineering 2 (2013) 101–108.

[12] T. Deepa, D. Cheelu, A comparative study of static and dynamic load balancing algorithms in cloud computing, in: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), IEEE, 2017, pp. 3375–3378.

[13] S. Mohapatra, S. Mohanty, K. S. Rekha, Analysis of different variants in round robin algorithms for load balancing in cloud computing, International Journal of Computer Applications 69 (2013) 17–21.

[14] R. Panwar, B. Mallick, Load balancing in cloud computing using dynamic load management algorithm, in: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, pp. 773–778.

[15] C.-L. Hung, H.-h. Wang, Y.-C. Hu, Efficient load balancing algorithm for cloud computing network, in: International Conference on Information Science and Technology (IST 2012), April, 2012, pp. 28–30.

[16] R. Rathore, B. Gupta, V. Sharma, K. K. Gola, A new approach for load balancing in cloud computing, International Journal Of Computers & Technology 13 (2014) 5193–5198.

[17] S. B. Shaw, A. Singh, A survey on scheduling and load balancing techniques in cloud computing environment, in: 2014 international conference on computer and communication technology (ICCCT), IEEE, 2014, pp. 87–95.

[18] V. R. Kanakala, V. K. Reddy, K. Karthik, Performance analysis of load balancing techniques in cloud computing environment, in: 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), IEEE, 2015, pp. 1–6.

[19] H. Shoja, H. Nahid, R. Azizi, A comparative survey on load balancing algorithms in cloud computing, in: Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE, 2014, pp. 1–5.

[20] J. M. Shah, K. Kotecha, S. Pandya, D. Choksi, N. Joshi, Load balancing in cloud computing: Methodological survey on different types of algorithm, in: 2017 International Conference on Trends in Electronics and Informatics (ICEI), IEEE, 2017, pp. 100–107.

[21] N. K. Chien, N. H. Son, H. D. Loc, Load balancing algorithm based on estimating finish time of services in cloud computing, in: 2016 18th International conference on advanced communication technology (ICACT), IEEE, 2016, pp. 228–233.

[22] N. Shah, M. Farik, Static load balancing algorithms in cloud computing: Challenges & solutions, International Journal of Scientific & Technology Research 4 (2015) 365–367.

[23] C. Kumar, S. Marston, R. Sen, A. Narisetty, Greening the cloud: a load balancing mechanism to optimize cloud computing networks, Journal of Management Information Systems 39 (2022) 513–541.

[24] M. Kumar, S. C. Sharma, Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing, Procedia computer science 115 (2017) 322–329.

[25] S. Kumar, D. S. Rana, Various dynamic load balancing algorithms in cloud environment: a survey, Int. J. Comput. Appl 129 (2015) 0975–8887.

[26] A. Daryapurkar, M. V. Deshmukh, Efficient load balancing algorithm in cloud environment, International Journal Of Computer Science And Applications 6 (2013) 308–312.

[27] S. Parida, B. Panchal, An efficient dynamic load balancing algorithm using machine learning technique in cloud environment, International journal of scientific research in science, engineering and technology 4 (2018) 1184–1186.

[28] A. Pradhan, S. K. Bisoy, A novel load balancing technique for cloud computing platform based on pso, Journal of King Saud University-Computer and Information Sciences 34 (2022) 3988–3995.

[29] P. G. Gopinath, S. K. Vasudevan, An in-depth analysis and study of load balancing techniques in the cloud computing environment, Procedia Computer Science 50 (2015) 427–432.

[30] N. NRaghava, D. Singh, Comparative study on load balancing techniques in cloud computing, Open journal of mobile computing and cloud computing 1 (2014) 31–42.

[31] J. Kaur, S. Kinger, A survey on load balancing techniques in cloud computing, International Journal of Science and Research (IJSR) 3 (2014) 2662–2665.

[32] S. Sethi, A. Sahu, S. K. Jena, Efficient load balancing in cloud computing using fuzzy logic, IOSR Journal of Engineering 2 (2012) 65–71.

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

[34] M. Elmagzoub, D. Syed, A. Shaikh, N. Islam, A. Alghamdi, S. Rizwan, A survey of swarm intelligence based load balancing techniques in cloud computing environment, Electronics 10 (2021) 2718.

[35] R. G. Rajan, V. Jeyakrishnan, A survey on load balancing in cloud computing environments, International Journal of Advanced Research in Computer and Communication Engineering 2 (2013) 4726–4728.

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

[37] S. Aslam, M. A. Shah, Load balancing algorithms in cloud computing: A survey of modern techniques, in: 2015 National software engineering conference (NSEC), IEEE, 2015, pp. 30–35.

[38] X. Ren, R. Lin, H. Zou, A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast, in: 2011 IEEE international conference on cloud computing and intelligence systems, IEEE, 2011, pp. 220–224.

[39] T. Deepa, D. Cheelu, A comparative study of static and dynamic load balancing algorithms in cloud computing, in: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), IEEE, 2017, pp. 3375–3378.

[40] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, M. A. Alzain, A load balancing algorithm for the data centres to optimize cloud computing applications, IEEE Access 9 (2021) 41731–41744.

[41] R. Panwar, B. Mallick, Load balancing in cloud computing using dynamic load management algorithm, in: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE, 2015, pp. 773–778.

[42] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, J. Al-Jaroodi, A survey of load balancing in cloud computing: Challenges and algorithms, in: 2012 second symposium on network cloud computing and applications, IEEE, 2012, pp. 137–142.

[43] A. Khiyaita, H. El Bakkali, M. Zbakh, D. El Kettani, Load balancing cloud computing: state of art, 2012 National Days of Network Security and Systems (2012) 106–109.

[44] P. Beniwal, A. Garg, A comparative study of static and dynamic load balancing algorithms, International journal of advance research in computer science and management studies 2 (2014) 1–7.

[45] N. NRaghava, D. Singh, Comparative study on load balancing techniques in cloud computing, Open journal of mobile computing and cloud computing 1 (2014) 31–42.

[46] S. Bhattacherjee, R. Das, S. Khatua, S. Roy, Energy-efficient migration techniques for cloud environment: a step toward green computing, The Journal of Supercomputing 76 (2020) 5192–5220.

[47] S. Patel, H. Patel, N. Patel, Dynamic load balancing techniques for improving performance in cloud computing, International Journal of Computer Applications 138 (2016) 1–5.

[48] D. Kashyap, J. Viradiya, A survey of various load balancing algorithms in cloud computing, Int. J. Sci. Technol. Res 3 (2014) 115–119.

[49] A. Tripathi, S. Singh, A literature review on algorithms for the load balancing in cloud computing environments and their future trends, Computer Modelling & New Technologies 21 (2017) 64–73.

[50] A. Kumar, S. Pandey, V. Prakash, A survey: Load balancing algorithm in cloud computing, in: Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE), 2019.

[51] S. Acharya, D. A. D'Mello, Energy and cost efficient dynamic load balancing mechanism for resource provisioning in cloud computing, International Journal of Applied Engineering Research 12 (2017) 15782–15790.

[52] N. Thapliyal, P. Dimri, Green cloud computing strategies for load balancing, Journal of Computational and Theoretical Nanoscience 17 (2020) 2695–2698.

[53] S. Ray, A. De Sarkar, Execution analysis of load balancing algorithms in cloud computing environment, International Journal on Cloud Computing: Services and Architecture (IJCCSA) 2 (2012) 1–13.

[54] A. Singh, D. Juneja, M. Malhotra, Autonomous agent based load balancing algorithm in cloud computing, Procedia Computer Science 45 (2015) 832–841.

[55] M. Katyal, A. Mishra, A comparative study of load balancing algorithms in cloud computing environment, arXiv preprint arXiv:1403.6918 (2014).

[56] H. Durani, N. Bhatt, Evaluation and comparison of various static and dynamic load balancing strategies used in cloud computing, in: Soft Computing and Signal Processing: Proceedings of 4th ICSCSP 2021, Springer, 2022, pp. 429–440.

[57] D. Girsa, D. Chauhan, D. Sharma, Optimised load balancing for green cloud computing, in: 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2023, pp. 171–175.

[58] S. Roy, M. A. Hossain, S. K. Sen, N. Hossain, M. R. Al Asif, Measuring the performance on load balancing algorithms, Global Journal of Computer Science and Technology (2019).

[59] M. Katyal, A. Mishra, A comparative study of load balancing algorithms in cloud computing environment, arXiv preprint arXiv:1403.6918 (2014).

[60] A. Dave, B. Patel, G. Bhatt, Y. Vora, Load balancing in cloud computing using particle swarm optimization on xen server, in: 2017 Nirma University International Conference on Engineering (NUiCONE), IEEE, 2017, pp. 1–6.

[61] A. Kazeem Moses, A. Joseph Bamidele, O. Roseline Oluwaseun, S. Misra, A. Abidemi Emmanuel, Applicability of mmrr load balancing algorithm in cloud computing, International Journal of Computer Mathematics: Computer Systems Theory 6 (2021) 7–20.

[62] C.-L. Hung, H.-h. Wang, Y.-C. Hu, Efficient load balancing algorithm for cloud computing network, in: International Conference on Information Science and Technology (IST 2012), April, 2012, pp. 28–30.

[63] H. Shoja, H. Nahid, R. Azizi, A comparative survey on load balancing algorithms in cloud computing, in: Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), IEEE, 2014, pp. 1–5.

[64] M. Xu, W. Tian, R. Buyya, A survey on load balancing algorithms for virtual machines placement in cloud computing, Concurrency and Computation: Practice and Experience 29 (2017) e4123.

[65] J. James, B. Verma, Efficient vm load balancing algorithm for a cloud computing environment, International Journal on Computer Science and Engineering 4 (2012) 1658.

[66] N. K. Chien, N. H. Son, H. D. Loc, Load balancing algorithm based on estimating finish time of services in cloud computing, in: 2016 18th International conference on advanced communication technology (ICACT), IEEE, 2016, pp. 228–233.

[67] I. Naz, S. Naaz, R. Biswas, A parametric study of load balancing techniques in cloud environment, in: Inventive Communication and Computational Technologies: Proceedings of ICICCT 2019, Springer, 2020, pp. 881–890.

[68] B. Ahmad, S. I. McClean, D. Charles, G. Parr, Energy saving techniques comparison for green computing in cloud server, International Journal On Advances in Intelligent Systems (2018) 192.

[69] M. K. Abiodun, Algorithm simulation of an efficient strategy for green cloud computing (2022).

[70] E. Hridya, 'analyzing the performance of load balancing algorithms in cloud computing, Int. J. Inf. Comput. Sci 6 (2019).

[71] C. Vaidya, K. S. Bhure, Survey on cloud computing load balancing, i-manager's Journal on Cloud Computing 7 (2020) 1.

[72] H. M. PATEL, R. R. CHAUDHARI, Load balancing techniques: Essentials, issues and major challenges in cloud environment-a meticulous review (????).

[73] X. Wang, B. Veeravalli, A genetic algorithm based efficient static load distribution strategy for handling large-scale workloads on sustainable computing systems, Intelligent Decision Support Systems for Sustainable Computing: Paradigms and Applications (2017) 7–31.

[74] A. K. Sidhu, S. Kinger, Analysis of load balancing techniques in cloud computing, International Journal of computers & technology 4 (2013) 737–741.

[75] E. R. Aggarwal, Managing load balancing in cloud computing, International Journal of Innovative Research and Advanced Studies (IJIRAS) 5 (????).

[76] X. Wang, N. Li, G. Gong, X. Song, Y. Guo, Load-balancing scheduling of simulation tasks based on a static-dynamic hybrid algorithm, Journal of Simulation 16 (2022) 182–193.

[77] M. Gamal, R. Rizk, H. Mahdi, B. E. Elnaghi, Osmotic bio-inspired load balancing algorithm in cloud computing, IEEE Access 7 (2019) 42735–42744.

[78] N. Shah, M. Farik, Static load balancing algorithms in cloud computing: Challenges & solutions, International Journal of Scientific & Technology Research 4 (2015) 365–367.

[79] S.-L. Chen, Y.-Y. Chen, S.-H. Kuo, Clb: A novel load balancing architecture and algorithm for cloud services, Computers & Electrical Engineering 58 (2017) 154–160.

[80] R. Panwar, B. Mallick, A comparative study of load balancing algorithms in cloud computing, International Journal of Computer Applications 117 (2015).

[81] S. K. Mishra, B. Sahoo, P. P. Parida, Load balancing in cloud computing: a big picture, Journal of King Saud University-Computer and Information Sciences 32 (2020) 149–158.

[82] M. R. Mesbahi, M. Hashemi, A. M. Rahmani, Performance evaluation and analysis of load balancing algorithms in cloud computing environments, in: 2016 Second International Conference on Web Research (ICWR), IEEE, 2016, pp. 145–151.

[83] S. Mousavi, A. Mosavi, A. R. Varkonyi-Koczy, A load balancing algorithm for resource allocation in cloud computing, in: Recent Advances in Technology Research and Education: Proceedings of the 16th International Conference on Global Research and Education Inter-Academia 2017 16, Springer, 2018, pp. 289–296.

[84] D. A. Shafiq, N. Jhanjhi, A. Abdullah, Proposing a load balancing algorithm for the optimization of cloud computing applications, in: 2019 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS), IEEE, 2019, pp. 1–6.

[85] P. Nandal, D. Bura, M. Singh, S. Kumar, Analysis of different load balancing algorithms in cloud computing, International Journal of Cloud Applications and Computing (IJCAC) 11 (2021) 100–112.

[86] M. Alam, Z. A. Khan, Issues and challenges of load balancing algorithm in cloud computing environment, Indian journal of science and Technology 10 (2017) 1–12.

[87] J. Prassanna, P. A. Jadhav, V. Neelanarayanan, Towards an analysis of load balancing algorithms to enhance efficient management of cloud data centres, in: Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16'), Springer, 2016, pp. 143–155.

[88] R. Z. Khan, M. O. Ahmad, A survey on load balancing algorithms in cloud computing, International Journal of Autonomic Computing 2 (2017) 366–383.

[89] K. Sharma, D. Thakur, A. Sharma, R. Khalkar, An efficient load balancing technique in cloud computing (????).

[90] S. K. Mohapatra, P. Nayak, S. Mishra, S. K. Bisoy, Green computing: a step towards eco-friendly computing, in: Emerging trends and applications in cognitive computing, IGI Global, 2019, pp. 124–149.

[91] R. Ben Hamouda, S. Boussema, I. Ben Hafaiedh, R. Robbana, Performance evaluation of dynamic load balancing protocols based on formal models in cloud environments, in: Verification and Evaluation of Computer and Communication Systems: 12th International Conference, VECoS 2018, Grenoble, France, September 26–28, 2018, Proceedings 12, Springer, 2018, pp. 64–79.

[92] S. Shukla, A. K. Singh, V. K. Sharma, Survey on importance of load balancing for cloud computing, in: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), IEEE, 2021, pp. 1479–1484.

[93] A. Saxena, N. Sharma, J. Goyal, S. Saxena, A study on benefits and classification of load balancing in cloud computing environment, in: Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT), 2018, pp. 26–27.

[94] X. Li, M. Zheng, An energy-saving load balancing method in cloud data centers, in: Frontier and Future Development of Information Technology in Medicine and Education: ITME 2013, Springer, 2014, pp. 365–373.

[95] A. Jain, R. Kumar, Critical analysis of load balancing strategies for cloud environment, International Journal of Communication Networks and Distributed Systems 18 (2017) 213–234.

[96] S. Rani, D. Kumar, S. Dhingra, A review on dynamic load balancing algorithms, in: 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, 2022, pp. 515–520.

[97] J. Zhang, Q. Liu, J. Chen, An advanced load balancing strategy for cloud environment, in: 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), IEEE, 2016, pp. 240–243.

[98] Z. Liu, M. Lin, A. Wierman, S. H. Low, L. L. Andrew, Greening geographical load balancing, ACM SIGMETRICS Performance Evaluation Review 39 (2011) 193–204.

[99] W. Dargie, A. Schill, Analysis of the power and hardware resource consumption of servers under different load balancing policies, in: 2012 IEEE Fifth International Conference on Cloud Computing, IEEE, 2012, pp. 772–778.

[100] D. Divya, B. M. Goel, Issues and challenges of load-balancing algorithms in cloud-computing, in: 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), IEEE, 2022, pp. 500–504.

[101] P. Kaur, M. Sachdeva, Optimized load balancing strategy in cloud computing: A, INTERNATIONAL JOURNAL 15 (????).

[102] H. Mahdizadeh, Designing a smart method for load balancing in cloud computing, International Journal of Mechatronics, Electrical and Computer Technology (IJMEC) 7 (2017) 3314–3324.

[103] F. Busato, N. Bombieri, A performance, power, and energy efficiency analysis of load balancing techniques for gpus, in: 2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES), IEEE, 2017, pp. 1–8.

[104] G. Sridevi, A systematic survey on load balancing in the cloud, in: Intelligent Computing and Innovation on Data Science: Proceedings of ICTIDS 2019, Springer, 2020, pp. 761–771.

[105] T. Rajagopal, M. Venkatesan, Energy efficient server with dynamic load balancing mechanism for cloud computing environment, Wireless Personal Communications 122 (2022) 3127–3136.

[106] N. X. Phi, C. T. Tin, L. N. K. Thu, T. C. Hung, Proposed load balancing algorithm to reduce response time and processing time on cloud computing, Int. J. Comput. Netw. Commun 10 (2018) 87–98.

[107] A. P. Sheetal, R. T. Bhima, R. Karampudi, S. D. V. Prasad, Load balancing and parallel computation model for performance and accuracy over the cluster of nodes., Ingénierie des Systèmes d'Information 27 (2022).

[108] S. S. Moharana, R. D. Ramesh, D. Powar, Analysis of load balancers in cloud computing, International Journal of Computer Science and Engineering 2 (2013) 101–108.

[109] Y. Xu, L. Wu, L. Guo, Z. Chen, L. Yang, Z. Shi, An intelligent load balancing algorithm towards efficient cloud computing, in: Proceedings of the 8th AAAI Conference on AI for Data Center Management and Cloud Computing, 2011, pp. 27–32.

[110] R. Sajjan, B. R. Yashwantrao, Load balancing and its algorithms in cloud computing: a survey, International Journal of Computer

Sciences and Engineering 5 (2017) 95–100.
[111] V. R. Kanakala, V. K. Reddy, K. Karthik, Performance analysis of load balancing techniques in cloud computing environment, in: 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), IEEE, 2015, pp. 1–6.
[112] S. Afzal, G. Kavitha, Load balancing in cloud computing–a hierarchical taxonomical classification, Journal of Cloud Computing 8 (2019) 22.
[113] V. Behal, A. Kumar, Cloud computing: Performance analysis of load balancing algorithms in cloud heterogeneous environment, in: 2014 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence), IEEE, 2014, pp. 200–205.
[114] Y. Sahu, R. Pateriya, R. K. Gupta, Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm, in: 2013 5th International conference and computational intelligence and communication Networks, IEEE, 2013, pp. 527–531.
[115] M. Gamal, R. Rizk, H. Mahdi, B. Elhady, Bio-inspired load balancing algorithm in cloud computing, in: Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017, Springer, 2018, pp. 579–589.
[116] M. Gamal, R. Rizk, H. Mahdi, B. E. Elnaghi, Osmotic bio-inspired load balancing algorithm in cloud computing, IEEE Access 7 (2019) 42735–42744.
[117] A. Hota, S. Mohapatra, S. Mohanty, Survey of different load balancing approach-based algorithms in cloud computing: a comprehensive review, in: Computational Intelligence in Data Mining: Proceedings of the International Conference on CIDM 2017, Springer, 2019, pp. 99–110.
[118] N. Mazher, F. Riaz, M. Tariq, F. Ishaque, H. S. Shakir, et al., Performance comparison of load balancing algorithms using cloud analyst in cloud computing (2018).
[119] S. Joshi, U. Kumari, Load balancing in cloud computing: Challenges & issues, in: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), IEEE, 2016, pp. 120–125.
[120] J. M. Shah, K. Kotecha, S. Pandya, D. Choksi, N. Joshi, Load balancing in cloud computing: Methodological survey on different types of algorithm, in: 2017 International Conference on Trends in Electronics and Informatics (ICEI), IEEE, 2017, pp. 100–107.
[121] B. Ghutke, U. Shrawankar, Pros and cons of load balancing algorithms for cloud computing, in: 2014 International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2014, pp. 123–127.
[122] R. M. Abdul-Hussein, A. H. Mohammed, Optimized load balance scheduling algorithm, TELKOMNIKA (Telecommunication Computing Electronics and Control) 20 (2022) 81–88.

A. IPR Certificate

We, Akbar Khan and Niyati Soni, with this certify that the project work submitted by us entitled Analysing Load Balancing ALgorithms to our supervisors, Dr. Alok Kumar and Dr. Utsav Upadhyay, in partial fulfillment of the requirements for the course is a bonafide work carried out by us and has not been previously submitted to any other course. We further certify that no part of this work shall be published, reproduced, or distributed in any form without the prior permission of our supervisors. We understand that any such unauthorized use of the project work may be considered a violation of academic ethics and result in severe penalties. We also affirm that the project work has been carried out under the ethical standards and guidelines set forth by our supervisors. We acknowledge that our supervisor has the right to make any modifications or revisions to the project work that may be deemed necessary. We also agree to abide by any additional terms and conditions as stipulated by our supervisor.

Date: 11-05-2023

Signature of Student:
Akbar Khan
2021BCA002
JK Lakshmipat University, Jaipur

Signature of Student:
Niyati Soni
2021BCA017
JK Lakshmipat University, Jaipur