Educational Administration: Theory and Practice
2024,30(2), 580-590
ISSN:2148-2403
https://kuey.net/

Research Article

A Review exploration of Load Balancing Techniques in Cloud Computing

Ms. Roopali Gupta[1*], Dr. Om Prakash Sharma[2]

[1*]Research Scholar, Faculty of Engineering and Technology, Jagan Nath University, Jaipur, Rajasthan, India roopaliakshaygupta@gmail.com
[2]Professor, Faculty of Engineering and Technology, Jagan Nath University, Jaipur, Rajasthan, India omprakash1.sharma@jagannathuniversity.com

Citation: Ms. Roopali Gupta, et al. (2024), A Review exploration of Load Balancing Techniques in Cloud Computing, *Educational Administration: Theory And Practice*, *30*(2), 580-590
Doi: 10.53555/kuey.v30i2.1600

ARTICLE INFO

ABSTRACT

Cloud computing revolutionizes IT services delivery, offering consumers access to virtualized resources over the Internet. As it rapidly replaces in-house infrastructure, various categories of cloud services emerge, creating challenges for consumers in selecting suitable providers. Research highlights disparities between provider criteria and consumer needs, intensifying the difficulty of making informed choices. Categorizations based on access (private vs. public) and services (HaaS, PaaS, SaaS, IaaS) further complicate decision-making. Diverse pricing models exacerbate the challenge, hindering quality and cost comparisons. Cloud computing's flexibility, enabled by virtual machine migration and economies of scale, attracts businesses seeking superior IT services at reduced costs. The advent of load balancing in cloud computing becomes paramount for optimizing resource usage, enhancing performance, and managing traffic spikes. This paper examines load balancing techniques and their role in addressing challenges and optimizing cloud computing benefits.

Keyword: Cloud Computing, Load Balancing, Round Robin

I. Introduction

The term "cloud computing" refers to a new model for delivering IT services in which consumers access both hardware and software through the Internet as "services" (in the form of virtualized resources) [1]. Several categories have been established for cloud services based on a variety of technological and economic criteria. Cloud computing is quickly replacing in-house IT infrastructure [2] as a consequence of its many benefits, including scalability, cost savings, and other technical advancements. The quantity and variety of cloud-based offerings have grown rapidly as a result of these considerations. It's vital to have a systematic approach to choosing cloud services that takes into account all of these factors. However, various approaches in research have suggested that the criteria of cloud service providers [3] may not always align with what the end customer really needs. With so many different cloud services to choose from, finding the right one might be difficult for potential consumers. Although there is an ever-increasing number of cloud service providers, it is getting more difficult for customers to choose the appropriate one for their specific needs. Private clouds provide access only to the company that owns them and its affiliates, whereas public clouds are available to anybody. The second categorization is based on the services provided by the cloud and comprises hardware as a service (HaaS) [4], platform as a service (PaaS) [5], software as a service (SaaS) [6], and infrastructure as a service (IaaS). To develop an app, a PaaS user may use the cloud provider's computing platform (like Google's App Engine). Users of infrastructure-as-a-service cloud models, on the other hand, deploy their software to the cloud service providers' virtual computers. This categorization is conceived as a hierarchical structure. Since SaaS operates above PaaS, which is dependent upon IaaS, this idea has practical implications in cloud computing. In most cases, smaller service providers rely on the infrastructure of bigger ones. Moreover, pricing models varied not just across service providers but also between services offered by the same provider on the same infrastructure. Some, like Google's App Engine [7], charge customers based on the number of CPU cycles they consume, while others, like Amazon's Elastic Compute Cloud, base their pricing on the number of virtual machine instances they use. It is becoming more difficult to compare one cloud provider to another in terms of the quality and

cost of service due to the proliferation of cloud providers and the diversity of the services they provide on wildly differing pricing schemes. Another reason cloud computing is so flexible is because virtual machine migration is feasible thanks to virtualization, one of the primary technologies that enable cloud computing. With the help of virtual machine migration, running applications may be moved from one virtual machine to another, even if they are hosted by different IaaS providers [8]. When compared to an in-house IT infrastructure, cloud computing's ability to deliver superior IT services at a reduced cost due to economies of scale and the elasticity of the cloud makes it an appealing alternative for businesses.

II. Load Balancing in Cloud Computing

By spreading the load over several servers and machines, cloud computing eliminates the possibility of any one system being overworked, underutilized, or overwhelmed. For better overall cloud performance, load balancing may be used to fine-tune various limited factors, including processing speed, reaction time, and stability. A load balancer, which is part of the cloud's load balancing architecture [9], is positioned between the servers and the clients. By balancing the demand of various users and tasks across a cloud's available resources, load balancing improves the performance and uptime of cloud-based software. With cloud load balancing, businesses can control how requests from clients are distributed over a collection of servers and networks. Load balancing in the cloud is used to ensure that users of an application get the fastest possible response time while also making the most efficient use of the available resources.

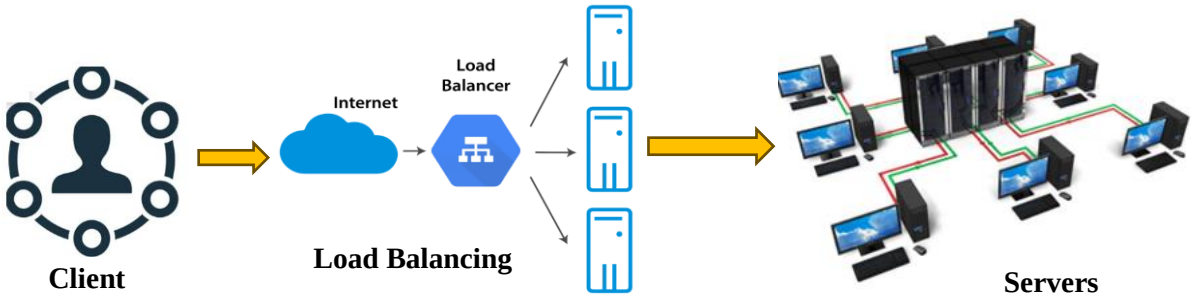

Fig: Load Balancing in Cloud Computing

Load balancing in the cloud is a managed service that is completely software-defined and distributed [10]. Being software-based removes the need to maintain a separate hardware load balancing setup. Exactly like Google's front-end serving architecture, Cloud Load Balancing is based on the latest and greatest open-source technologies. It can handle more than a million requests per second while maintaining great speed and low latency at all times. Cloud Load Balancing optimises the use of Google's high-speed private network by directing traffic to one of eighty or more load balancing nodes situated around the world. Utilizing Cloud Load Balancing, this can ensure that users are receiving content from servers that are geographically near to them. Among the load balancing options provided by Google Cloud are as follows.

- **With just one anycast IP address, communication between devices is simplified.** With Cloud Load Balancing, all of the backend instances in different regions may share a single anycast IP address as the frontend. It offers global load balancing and automated multi-region failover, which redirects traffic to secondary servers in the event that main servers go down. Cloud load balancing is responsive to user, traffic, network, and backend health changes in real time.

- **Using load balancing that is specified in software.** Cloud Load Balancing is a managed service that distributes and balances traffic across several servers using software-defined networking. Since it is neither instance- or device-based, this won't have to worry about the HA, scaling, and administration issues that plague traditional load balancers.

- **The autoscaling feature works without any disruptions.** With Cloud Load Balancing, this can easily handle massive, sudden increases in both user numbers and traffic volumes by rerouting it to more capable parts of the globe. With autoscaling, this can quickly and easily increase capacity from zero to full traffic levels.

- **Load balancing on both Layer 4 and Layer 7 are supported.** Apply load balancing at Layer 4 [11] based on information from higher-level protocols including TCP, UDP, ESP, GRE, ICMP, and ICMPv6. Use

load balancing at Layer 7 to make judgments about how to route requests depending on parameters like the HTTP header and the uniform resource identifier.

- **Balancing of both internal and external loads.** When customers are located outside of Google Cloud, this should use external load balancing, and when they are within, this should employ internal load balancing.

- **Load balancing on a global and regional scale.** This may satisfy high availability goals by spreading the load-balanced resources over one or more regions, allowing the user to terminate connections near to the users.
- **Extraordinary Feature Compatibility:** IPv6 global load balancing [12], source-IP based traffic steering, WebSockets, user-defined request headers, and protocol forwarding for private VIPs are just some of the capabilities that Cloud Load Balancing enables.

2.1 Techniques in Cloud Computing

In order to avoid overloading any one cloud server, load balancing in cloud computing distributes traffic among cloud servers and manages substantial workloads. Consequently, performance is improved while downtime is minimized. By spreading the demand across multiple servers, advanced load balancing in the cloud helps increase server availability and dependability while reducing latency. Implementations of cloud load balancing that are both successful and efficient use various load balancing approaches to guard against server failure and enhance performance. The load balancer may consider factors such as physical distance and server load when deciding where to send traffic in the case of a failover. Network load balancers may take the form of either hardware appliances or purely software-based procedures. Hardware load balancers [13] are less effective at handling cloud traffic and are sometimes disallowed from functioning in vendor-managed cloud systems. Client DNS queries in cloud computing are load balanced across several servers using a software-defined load balancing technique called domain name system (DNS) load balancing [14]. To ensure that DNS queries are fairly spread between servers, the DNS system responds to each client request with a slightly modified version of the list of IP addresses. DNS load balancing eliminates unresponsive servers instantly and enables instantaneous failover and backup. To prevent traffic jams, load balancing in the cloud works similarly to how a traffic cop directs vehicles. The police may use static methods like counting vehicles or seconds to determine how quickly they should go, but they also have access to dynamic methods that allow them to adapt to the constantly changing flow of traffic. Similarly, to avoid revenue loss and a poor user experience due to overloaded apps and servers, load balancing in the cloud operates in a similar way [15]. Load balancing techniques in the cloud come in a wide variety, with some being more common than others. The way they handle network traffic and make decisions about which servers to prioritise client requests varies. Cloud computing's eight most popular load balancing methods are as following.

Round Robin: Round Robin is a basic, recursive method used for load balancing in cloud computing. One of the most popular static load balancing strategies in the cloud is the standard round robin approach. This is one of the simplest methods to put into action, but it may not be the most effective since it presumes that all servers have the same capacity. Two methods, weighted round robin and dynamic round robin, are designed to address this problem.

IP Hash: This easy technique of load balancing divides up requests according to IP address. By generating random hash keys, this load balancing method distributes requests from clients among available servers. Hashes are encrypted versions of the final destination, the source, and the originating IP.

Least Connections: The Least Connections technique is one of the most widely used dynamic load balancing methods in cloud computing, and it shines in situations when traffic spikes. Because it is more efficient to spread traffic over all available servers, Least Connections prioritises that which has the fewest active connections.

Least Response Time: This dynamic method is comparable to least connections in that it prioritises servers according to their ability to respond quickly while also handling a low volume of simultaneous connections.

Least Bandwidth: The least bandwidth approach is another kind of dynamic load balancing used in the cloud, and it works by routing requests from clients to the server that has used the least bandwidth in the most recent time period.

Layer 4 Load Balancers: To balance traffic, L4 load balancers look at the source and destination IP addresses, as well as the protocol (UDP or TCP) and port number used to send and receive data. L4 load balancers [16] perform Network Address Translation (NAT) [17], which just reroutes packets to the correct servers without looking inside them.

Layer 7 Load Balancers: L7 load balancers, which operate on the application layer of the OSI model, look at things like SSL session IDs and HTTP headers to figure out which servers to send requests to. Because they need more information to properly direct requests to servers, L7 load balancers are both more effective and more computationally demanding than L4 load balancers.

Global Server Load Balancing: With Global Server Load Balancing (GSLB), this can spread huge quantities of traffic across data centres without sacrificing speed by leveraging the full potential of L4 and L7 load balancers. The GSLB is especially useful for coordinating requests for services from users located in different physical locations.

2.2 Load Balancer as a Service in Cloud Computing

Load balancing as a service (LBaaS) [18] is a feature provided by many cloud providers, and it is used by clients in lieu of dedicated traffic routing equipment installed on-premises and configured and maintained locally. LBaaS is a common kind of load balancing in the cloud that functions similarly to more conventional forms of load balancing. Instead of distributing traffic over a cluster of computers in a single data centre, LBaaS does so across many cloud environments and operates itself as a subscription or as-needed service. While some LBaaS settings are created and managed by a single cloud service provider, others use traffic distribution techniques that include several cloud service providers, multi-cloud load balancers, and hybrid cloud deployments.

2.3 Benefits of LBaaS include

Rapidly expand capacity to load-balancing services in response to unexpected surges in traffic without requiring manual configuration of supplementary hardware. In the event of a server outage, this may still maintain high availability by connecting to the server that is physically nearest to user. In comparison to hardware-based appliances, the upfront cost of LBaaS is often lower, and ongoing maintenance expenses are also lower requiring less internal resources [19].

III. Research Background

Verma (2022) [1]

Methodology

Author presented the Dual Conditional Moth Flame Algorithm (DC-MFA) in distributed computing for enhanced model efficacy on cloud resources. The study focused on semi-concentrated architecture prevalent in contemporary enterprise environments, emphasizing virtualization's role in constructing digital representations, particularly Virtual Machine (VM) technology for workload partitioning and migration.

Research Gaps

Existing literature lacks comprehensive exploration of optimization algorithms for distributed computing in semi-concentrated architectures. Author’s DC-MFA proposes advancements, yet gaps persist in addressing load balancing challenges in virtual cloud computing, hindering optimal resource minimization and system reliability enhancement.

Nazeri & Khorsand (2022) [2]

Methodology

The study employs a simulation-based assessment using diverse job requests, constrained by various factors. Results showcase the superiority of the proposed fuzzy AHP-TOPSIS hybrid technique over SHARP and BULLET algorithms, excelling in resource usage, user satisfaction, and energy efficiency across scenarios.

Research Gaps

Current research lacks comprehensive solutions for optimizing cloud computing's task scheduling on heterogeneous resources, especially in balancing user satisfaction and resource conservation. Dynamic distribution based on QoS preferences poses a critical challenge for cloud service providers. The study addresses these gaps through a novel fuzzy AHP-TOPSIS approach, integrating FAHP for solution ranking and FTOPSIS for optimal selection, facilitating efficient resource utilization and meeting diverse user needs.

Sefati et al. (2022) [3]

Methodology

Applying Grey Wolf Optimization (GWO) to ensure load balance among resources based on dependability. GWO identifies idle/busy nodes, determines thresholds and fitness functions per node. CloudSim results show cost and response time advantages, proving optimal solutions.

Research Gaps

Despite advancements in cloud computing, load balancing remains a critical challenge. Existing approaches use metaheuristic algorithms, yet maintaining balance in dispersed resource deployments poses difficulties.

Further research is needed to enhance load balancing efficiency and address the evolving complexities of cloud computing systems.

Bharany et al. (2022) [4]
Methodology
The research employs a comprehensive approach, integrating AI, deep learning, IoT, and machine learning. It investigates the link between defects and energy consumption in cloud computing, exploring intelligent fault tolerance methods. The study analyses existing approaches, aiming to enhance understanding and address identified obstacles.

Research Gaps
Current research identifies a connection between defects and energy use in cloud computing. However, gaps exist in understanding the integration of cutting-edge technologies like AI, deep learning, IoT, and machine learning for intelligent fault tolerance. The study seeks to bridge these gaps by delving into existing approaches and addressing obstacles in achieving high fault tolerance and performance in the cloud.

Negi et al. (2021) [5],
Methodology
A clustering-based multiple objective dynamic load balancing approach is proposed, followed by task scheduling for underloaded VMs. Utilizing multi-objective techniques, including order preference by similarity to ideal solution with particle swarm optimization, user tasks are aligned with diverse cloud-based criteria. VM migration decisions, guided by the VM manager, enhance load balance among PMs by mitigating overcrowded and underutilized conditions. The approach harmoniously integrates machine learning, multi-objective, and soft computing techniques for efficient PM and VM balance.

Research Gaps
Existing research lacks a comprehensive approach to dynamic load balancing in cloud environments. Limited focus on multi-objective techniques, such as order preference by similarity to ideal solution with particle swarm optimization, hinders optimal task scheduling. Additionally, there's a gap in addressing both energy efficiency and load balance through VM migration decisions. This study bridges these gaps, offering a unique hybridization of machine learning, multi-objective, and soft computing techniques for improved PM and VM equilibrium.

Moori et al. (2022) [6],
Methodology
The study addresses cloud computing challenges by introducing the LATOC approach. It prioritizes tasks based on key criteria, utilizing optimized particle swarm optimization for efficient distribution across virtual machines. Cloudsim simulations validate LATOC's effectiveness in improving critical statistics compared to other approaches.

Research Gaps
Existing research highlights cloud computing challenges, emphasizing software/hardware complexity and task distribution issues. Recent studies reveal shortcomings in scheduling, leading to load balancing problems. LATOC addresses this gap by intelligently prioritizing and distributing tasks, demonstrating improved cloud computing metrics in diverse use cases.

Shafiq et al. (2021) [7]
Methodology
Utilizing a comprehensive approach, this study evaluates various Load Balancing strategies across static, dynamic, and cloud environments. Inspired by nature, algorithms are rigorously scrutinized. The research employs graphical representations for clarity and proposes a fault-tolerant framework to enhance Data Centre Response Time and overall performance.

Research Gaps
Existing literature on Load Balancing lacks in-depth exploration of fault-tolerant frameworks, creating a research gap. Addressing this void can further optimize cloud-based services, ensuring uninterrupted application performance and improved user satisfaction.

Sharma et al. (2021) [9],
Methodology
The study employs a comprehensive analysis of diverse load-balancing algorithms, assessing their impact on various performance metrics. Assumptions guide practical application, emphasizing a balanced approach to

maximize multiple metrics. The research prioritizes qualities like flexibility, scalability, and on-demand access, delivering them as a utility in cloud environments. Load distribution maintains system characteristics during high-demand scenarios, with a focus on identifying and balancing overloaded and underloaded nodes across CPU, network, and memory loads.

Research Gaps

While various load-balancing algorithms are explored, a unified framework for balancing diverse performance metrics remains a gap. The article highlights the need for further research to optimize the simultaneous maximization of multiple metrics, striking an optimal balance. Additionally, the study suggests investigating novel strategies that enhance load balancing while preserving packaging qualities like flexibility and scalability in cloud environments.

Mubeen et al. (2021) [10],
Methodology

The study employs an adaptive load-balanced task scheduling (ALTS) approach in cloud computing. Incoming tasks are mapped to available VMs to optimize resource usage, minimize make span, and adaptively reduce SLA violations. Performance metrics (ARUR, make span, SLA violation) are compared with existing GA, ACO, and GAACO methods.

Research Gaps

Existing job scheduling methods like GA and ACO address cloud data center performance, but the exponential increase in task scheduling solutions poses an NP-hard challenge. Achieving fully optimum user task scheduling is difficult. The study introduces ALTS, showing significant advantages over current methods in make span, SLA violations, and resource consumption, indicating a promising avenue for further research.

Mapetu et al. (2021) [15],
Methodology

Conduct a comprehensive analysis and simulation trials using genuine PlanetLab and random workloads to evaluate a proposed solution for the NP-hard optimization issue. Assess its performance in reducing data centre energy usage while maintaining SLAs and limiting VM migrations.

Research Gaps

Existing solutions, like dynamic VM consolidation, fall short in swiftly delivering an optimal resolution for the NP-hard problem of balancing data centre energy efficiency with SLA adherence and VM migration constraints.

Khorsand & Ramezanpour (2020) [20],
Methodology

A committee establishes standards for cloud-based scheduling, applying Best-Worst Method (BWM) for criterion weighting. Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) ranks solutions based on criteria importance. CloudSim benchmarks and ANOVA statistically compare proposed and current algorithms using metrics like makespan, energy consumption, and resource usage.

Research Gaps

Existing studies lack a comprehensive approach to prioritize and evaluate cloud-based scheduling factors. This research addresses this gap by introducing a BWM and TOPSIS-based algorithm, providing a systematic methodology to optimize energy efficiency and service quality in cloud data centres.

Chauhan et al. (2020)[21],
Methodology

A multi-criteria decision analysis–based cloud selection approach is developed, comparing Weighted Sum Model, Fuzzy Analytic Hierarchy Process, and Fuzzy Revised Analytic Hierarchy Process using ten criteria. AHP and revised AHP show comparable results, with AHP having superior computational capability, addressing the increasing demand for high-powered computing in mobile cloud systems.

Research Gaps

Despite advancements in cloud selection methods, the study reveals a need for further exploration of computational efficiency and scalability. Additionally, the impact of offloading on IoT device battery life and response time in critical situations requires more investigation. The diversity of cloud service providers and their unique offerings necessitates continued research to optimize cloud solutions and enhance customer satisfaction.

Jyoti et al. (2020)[22],
Methodology
This study employs a systematic review approach, gathering and analyzing data on load balancing algorithms, brokering policies, and scheduling types from diverse load balancers. Comparative assessments are conducted to unveil trends and variations, offering insights into optimizing cloud infrastructure.

Research Gaps
Existing literature lacks a comprehensive overview of current load balancing practices across diverse load balancers in the context of evolving cloud computing. Critical gaps include nuanced analyses of security measures, scalability challenges, and the integration of brokering policies, hindering a holistic understanding for practitioners and researchers.

Devaraj et al. (2020)[23],
Methodology
Implemented a Cloud Computing (CC) framework with the FIMPSO algorithm, leveraging Firefly (FF) for search space narrowing and Improved Multi-Objective Particle Swarm Optimization (IMPSO) for optimal solution determination. Evaluated key indicators, enhancing resource utilization and job reaction time in load balancing.

Research Gaps
Existing Cloud Computing (CC) load balancing methods lack a comprehensive approach. This study addresses gaps by introducing FIMPSO, combining Firefly and IMPSO, demonstrating superior performance in resource utilization, job reaction time, and average load compared to other approaches.

Hosseinzadeh et al. (2020)[24],
Methodology
This research employs a systematic literature review to comprehensively analyze existing Multi-criteria Decision Making (MCDM)-based service selection techniques. It establishes a taxonomy based on MCDM methods, scrutinizes procedural modifications, assesses diverse datasets and QoS criteria, and explores evaluative elements and contexts.

Research Gaps
Existing literature lacks a unified overview of Multi-criteria Decision Making (MCDM) techniques for service selection. This study addresses this gap by presenting a comprehensive analysis of various MCDM-based service selection methods, identifying nuances in their approaches, and suggesting avenues for future research.

3.1 Concise Tabulation, Technology, Hindrance and Findings:

Author	Year	Technology Used	Hindrance	Findings
Verma	2022	Dual Conditional Moth Flame Algorithm in distributed computing on cloud resources.	Lack of optimization algorithms exploration for semi-concentrated architectures. Verma's DC-MFA addresses load balancing challenges in virtual cloud computing.	Introduced DC-MFA for enhanced model efficacy. Focused on semi-concentrated architecture. Emphasized virtualization, VM technology, and workload partitioning on cloud resources.
Nazeri & Khorsand	2022	Simulation-based assessment using fuzzy AHP-TOPSIS hybrid technique for task scheduling.	Limited solutions for optimizing cloud computing's task scheduling on heterogeneous resources.	Proposed fuzzy AHP-TOPSIS for superior resource usage, user satisfaction, and energy efficiency. Outperformed SHARP and BULLET algorithms in diverse scenarios.
Sefati et al.	2022	Grey Wolf Optimization for load balance in cloud computing.	Load balancing challenges persist in dispersed resource deployments.	Applied Grey Wolf Optimization for load balance. Demonstrated cost and response time advantages in CloudSim results.
Bharany et al.	2022	Integration of AI, deep learning, IoT, and machine learning for fault tolerance.	Gap in understanding the integration of cutting-edge technologies for intelligent fault tolerance.	Explored link between defects and energy consumption in cloud computing. Analyzed existing approaches to enhance fault tolerance.
Moori et al.	2022	Introduced LATOC approach for task prioritization and distribution using PSO.	Shortcomings in scheduling and load balancing problems in cloud computing.	Addressed cloud computing challenges with LATOC approach. Utilized PSO for efficient task distribution. Validated effectiveness through CloudSim simulations.
Negi et al.	2021	Clustering-based multiple objective dynamic load balancing with VM migration.	Lack of comprehensive approach to dynamic load balancing in cloud environments.	Proposed clustering-based approach for dynamic load balancing. Integrated multi-objective techniques for efficient PM and VM balance. Utilized machine learning and soft computing for optimal task scheduling.
Shafiq et al.	2021	Comprehensive evaluation of Load Balancing strategies with a fault-tolerant framework.	Lack of in-depth exploration of fault-tolerant frameworks in Load Balancing literature.	Evaluated various Load Balancing strategies. Proposed a fault-tolerant framework for enhanced Data Centre Response Time.
Sharma et al.	2021	Analysis of diverse load-balancing algorithms for balancing multiple metrics.	Gap in a unified framework for balancing diverse performance metrics.	Explored various load-balancing algorithms. Emphasized the need for a unified framework for balancing multiple metrics.

				Suggested investigating novel strategies for efficient load balancing in cloud environments.
Mubeen et al.	2021	Adaptive load-balanced task scheduling (ALTS) approach in cloud computing.	Existing job scheduling methods pose NP-hard challenges for optimum user task scheduling.	Introduced ALTS for adaptive load-balanced task scheduling. Showed advantages over current methods in make span, SLA violations, and resource consumption.
Mapetu et al.	2021	Evaluation of an NP-hard optimization solution for data centre energy efficiency.	Shortcomings in swiftly delivering optimal resolution for balancing energy efficiency, SLA adherence, and VM migrations.	Conducted comprehensive analysis and simulation trials for an NP-hard optimization solution. Assessed performance in reducing data centre energy usage while maintaining SLAs and limiting VM migrations.
Khorsand & Ramezanpour	2020	Best-Worst Method and TOPSIS-based algorithm for cloud-based scheduling.	Lack of a comprehensive approach to prioritize and evaluate cloud-based scheduling factors.	Established standards for cloud-based scheduling using BWM. Applied TOPSIS for ranking solutions based on criteria importance. Statistically compared proposed and current algorithms using CloudSim benchmarks and ANOVA.
Chauhan et al.	2020	Multi-criteria decision analysis–based cloud selection approach.	Need for further exploration of computational efficiency and scalability in cloud selection methods.	Developed a cloud selection approach based on multi-criteria decision analysis. Compared Weighted Sum Model, Fuzzy AHP, and Fuzzy Revised AHP. Emphasized computational efficiency and scalability.
Jyoti et al.	2020	Systematic review of load balancing algorithms, brokering policies, and scheduling types.	Lack of a comprehensive overview of current load balancing practices across diverse load balancers.	Employed a systematic review approach to analyse load balancing practices. Conducted comparative assessments to unveil trends and variations. Identified gaps in security measures, scalability challenges, and brokering policy integration in existing literature.
Devaraj et al.	2020	FIMPSO algorithm for Cloud Computing (CC) load balancing.	Existing CC load balancing methods lack a comprehensive approach.	Implemented FIMPSO algorithm for CC load balancing. Leveraged Firefly and Improved Multi-Objective Particle Swarm Optimization for optimal solution determination. Demonstrated superior performance in resource utilization, job reaction time, and average load compared to other approaches.
Hosseinzadeh et al.	2020	Systematic literature review of Multi-criteria Decision Making (MCDM) for service selection.	Lack of a unified overview of MCDM techniques for service selection in existing literature.	Conducted a systematic literature review of MCDM-based service selection techniques. Established a taxonomy based on MCDM methods. Scrutinized procedural modifications, assessed diverse datasets and QoS criteria, and explored evaluative elements and contexts. Identified nuances in approaches and suggested avenues for future research.

IV. Load Balancing Algorithms in the Cloud Computing

4.1 Round-Robin Algorithm

The round-and-robin algorithm stands out for its simplicity and effectiveness in load balancing, particularly in time-triggered scenarios. In a cloud computing context, tasks are randomly distributed among machines, emphasizing workload balance facilitated through data centres. The algorithm operates on a time-sharing principle, allocating processors to tasks based on assigned time slots within a circular queue. New processes are added to the end, and the algorithm randomly selects and shifts processes, accommodating completion variations. However, to address uneven loading and optimize allocation, a Weight round-robin load balancing mechanism is introduced. This enhancement ensures proportional distribution of weights, assigning greater significance to more powerful CPUs. The algorithm directs a continuous flow of work to servers with higher weight values, leading to eventual convergence. This optimized approach significantly improves load balancing, rectifying disparities in processing power and enhancing overall system performance. In a nutshell, the round-and-robin algorithm, while effective, benefits from the Weight round-robin mechanism to enhance load balancing, particularly in cloud computing scenarios, by optimizing resource allocation and promoting efficient processing. [25].

4.2 Opportunistic Algorithm

This technique for distributing work among systems is static in nature and does not take into account the actual load being placed on any one machine at any given time. As a result, it ensures that all nodes are actively engaged by allocating all outstanding jobs at random among the available nodes. In turn, this causes the algorithm to have subpar load-balancing performance [26]. Since it doesn't take into account the time required to implement the node, the processing operation is slowed down. There will be bottlenecks in the cloud infrastructure if there are nodes in the idle state. Algorithm 3.3 Min-Min. The system prioritises activities with the lowest time requirements. It's quick and easy, and it boosts efficiency. To begin, the quickest possible time to finish all loads is determined. This minimal value is then used to schedule the job in the machine. The job is deleted from the accessible task list when the machine's current execution time has been updated. All the jobs

in the set are assigned to the comparable machine one by one, and the procedure repeats itself until that is complete [27].

4.3 Max-Min Algorithm

By first identifying the quickest way to do each work, the max-min algorithm determines the highest possible value that may be achieved. The algorithm then chooses a time-intensive job and gives it to the corresponding machine. After then, the algorithm revises the estimated time remaining for each task's completion and eventually removes the completed tasks from the list. This method is distinct from the min-min algorithm in that it includes just one lengthy job among a series of activities that are executed simultaneously [28].

4.4 Active Monitoring Algorithm

This is a dynamic load balancing method that gives work to the virtual machine that is now underutilised or underutilised overall. For load balancing, controllers keep track of all servers and requests in an index table. Therefore, the data centre anticipates that the index table will identify the servers that are least loaded or idle when a new request is received. When distributing work across the servers, the algorithm follows the principle of "first come, first served." The server-id is used to determine which server is responsible for a given job, and the index table's state is updated accordingly whenever work is distributed across servers. And similarly, when a job is done, the data centre and the controllers get the news, and the index table's server state is reduced. When a user makes a request over the internet, the load balancer will re-examine the index table and divide up the processes appropriately [29].

4.5 Equally Spread Current Execution Algorithm

Distributing the workload evenly among data centre servers, this is a dynamic load balancing mechanism. The algorithm will choose all the processes from the list, rank them, and then figure out how big they are and how much capacity they have. The programme then determines which server can manage the workload with the least amount of time spent doing so. Taking into account the virtual machine's resources and a rough estimate of the expected workload helps determine which server is the best option. The programme then distributes the work to the most suitable virtual machine in terms of size and processing power [30].

V. Advantages and Disadvantages of Load Balancing

5.1 Advantages of Load Balancing

With cloud computing, it's important to employ a load balancer. When it comes to reliable cloud computing, load balancers are important. In this section, we explored the advantages of load balancing in cloud computing and some of its advantages as in following,

Simpler Automation: Through load balancing in the cloud, businesses may anticipate traffic bottlenecks using predictive analytics and receive near-real-time insights into applications, which can then be used to inform strategic choices. All of these factors are essential to the process of automation.

Seamless Management of Traffic Spikes: In cloud computing, load balancing coordinates the use of many servers to provide seamless, high-performance capacity during peak periods of use, without requiring the intervention of IT personnel. This uniform dispersal allows for the fastest possible responses and the best possible outcomes, regardless of how rapidly requirements may change. Therefore, load balancing aids businesses in taking advantage of surges in demand without being overwhelmed by the resulting increase in network traffic. Retaining consumers and reducing churn may be aided by load balancing and scalable server capacity. [31]

Emergency and Disaster Recovery: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform [32] are just a few examples of cloud service providers that have the technology and expertise to detect downed servers and reroute traffic between regions in the event of a crisis or natural catastrophe. Furthermore, depending on the load balancing techniques supported by cloud load balancers in a network, administrators may frequently predict in advance which servers will be overworked. Some cloud load balancers are "planned" to quickly reroute server traffic to nodes that are in better shape to handle requests, hence lowering the probability of data loss and service interruption.

High Performing Applications: When traffic increases, efficiency and performance must grow as well, and cloud-based load balancing makes both possible.

Flexibility: The option to temporarily redirect traffic to other servers gives programmers more freedom when it comes to applying patches, fixing bugs, and testing in a live environment.

Cost-Effective: Load balancing in the cloud reduces the total cost of ownership while improving cloud service performance and dependability. Since they are hosted in the cloud or offered as a service, cloud load balancers are affordable for start-ups, SMEs, and even large corporations.

DDoS Attack Mitigation: Distributing traffic over numerous servers, rerouting traffic away from an overloaded server during a DDoS assault, and lowering the attack surface are all ways in which load balancers protect against DDoS attacks. By removing vulnerable nodes from the network, load balancing in cloud computing increases the network's resistance to these kinds of assaults. [33]

5.2 Disadvantages of Load Balancing

Although load balancing is essentially essential in the cloud setting, it does provide some unique difficulties when applied to cloud computing. Both cloud computing and load balancing have a number of attractive features, but one of load balancing's most glaring shortcomings is its inability to scale. In most cases, the load balancer's scalability is limited by the number of nodes it uses to distribute processes. In addition, there are a number of obstacles unique to the cloud, such as power use, performance tracking, quality of service management, resource scheduling, and service availability [34].

VI. Conclusion

The cloud computing's evolution and the proliferation of service categories demand a strategic approach to provider selection. The complexities of categorizations, pricing models, and provider misalignment with consumer needs necessitate a systematic decision-making process. Load balancing emerges as a crucial component, ensuring optimal resource usage, seamless management of traffic variations, and high-performing applications. Various load balancing techniques, from traditional methods like Round Robin to more advanced algorithms, provide flexibility and scalability. Despite the advantages, challenges such as scalability limitations, power consumption, and service availability persist. This review article acknowledged both the benefits and drawbacks, businesses must navigate the dynamic landscape of cloud computing and load balancing to harness their full potential in delivering efficient, reliable, and cost-effective IT services.

References

1. Verma, G. (2022). Secure VM migration in cloud: multi-criteria perspective with improved optimization model. Wireless Personal Communications, 124(1), 75-102.
2. Nazeri, M., & Khorsand, R. (2022). Energy Aware Resource Provisioning for Multi-Criteria Scheduling in Cloud Computing. Cybernetics and Systems, 1-30.
3. 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.
4. Bharany, S., Badotra, S., Sharma, S., Rani, S., Alazab, M., Jhaveri, R. H., & Gadekallu, T. R. (2022). Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy. Sustainable Energy Technologies and Assessments, 53, 102613.
5. Negi, S., Rauthan, M. M. S., Vaisla, K. S., & Panwar, N. (2021). CMODLB: an efficient load balancing approach in cloud computing environment. The Journal of Supercomputing, 77(8), 8787-8839.
6. Moori, A., Barekatain, B., & Akbari, M. (2022). LATOC: an enhanced load balancing algorithm based on hybrid AHP-TOPSIS and OPSO algorithms in cloud computing. The Journal of Supercomputing, 78(4), 4882-4910.
7. Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2021). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences.
8. Gandhi, R., Liu, H. H., Hu, Y. C., Lu, G., Padhye, J., Yuan, L., & Zhang, M. (2014). Duet: Cloud scale load balancing with hardware and software. *ACM SIGCOMM Computer Communication Review*, *44*(4), 27-38.
9. Sharma, M., Kumar, R., & Jain, A. (2021). Load balancing in cloud computing environment: A broad perspective. In Intelligent Data Communication Technologies and Internet of Things (pp. 535-551). Springer, Singapore.
10. Mubeen, A., Ibrahim, M., Bibi, N., Baz, M., Hamam, H., & Cheikhrouhou, O. (2021). Alts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing. Processes, 9(9), 1514.
11. Tsai, W., Bai, X., & Huang, Y. (2014). Software-as-a-service (SaaS): perspectives and challenges. Science China Information Sciences, 57(5), 1-15.
12. Sidhu, A. K., & Kinger, S. (2013). Analysis of load balancing techniques in cloud computing. International Journal of computers & technology, 4(2), 737-741.
13. Faizan, J., EL-Rewini, H., & Khalil, M. (2008). Introducing reliability and load balancing in mobile IPv6-based networks. *Wireless Communications and Mobile Computing*, *8*(4), 483-500.
14. Naik, V. S., Desai, P., Preksha, J., & Nethravathi, B. (2021). IAAS: THE FUTURE OF IT INFRASTRCUTURE.

15. Mapetu, J. P. B., Kong, L., & Chen, Z. (2021). A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing. The Journal of Supercomputing, 77(6), 5840-5881.
16. Stanik, A., Hovestadt, M., & Kao, O. (2012). Hardware as a Service (HaaS): Physical and virtual hardware on demand. In 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings (pp. 149-154). IEEE.
17. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., ... & Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing (Vol. 17). *Technical Report UCB/EECS-2009-28*, EECS Department, University of California, Berkeley.
18. Ferretti, S., Ghini, V., Panzieri, F., Pellegrini, M., & Turrini, E. (2010). Qos–aware clouds. In 2010 IEEE 3rd International Conference on Cloud Computing (pp. 321-328). IEEE.
19. Belgaum, M. R., Musa, S., Alam, M. M., & Su'ud, M. M. (2020). A systematic review of load balancing techniques in software-defined networking. IEEE Access, 8, 98612-98636.
20. Khorsand, R., & Ramezanpour, M. (2020). An energy-efficient task-scheduling algorithm based on a multi-criteria decision-making method in cloud computing. International Journal of Communication Systems, 33(9), e4379.
21. Chauhan, N., Agarwal, R., Garg, K., & Choudhury, T. (2020). Redundant IAAS cloud selection with consideration of multi criteria decision analysis. Procedia Computer Science, 167, 1325-1333.
22. Jyoti, A., Shrimali, M., Tiwari, S., & Singh, H. P. (2020). Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey. Journal of Ambient Intelligence and Humanized Computing, 11(11), 4785-4814.
23. Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. Journal of Parallel and Distributed Computing, 142, 36-45.
24. Hosseinzadeh, M., Hama, H. K., Ghafour, M. Y., Masdari, M., Ahmed, O. H., & Khezri, H. (2020). Service selection using multi-criteria decision making: a comprehensive overview. Journal of Network and Systems Management, 28(4), 1639-1693.
25. Garg, S., Gupta, D. V., & Dwivedi, R. K. (2016). Enhanced active monitoring load balancing algorithm for virtual machines in cloud computing. In 2016 International Conference System Modeling & Advancement in Research Trends (SMART) (pp. 339-344). IEEE.
26. Gandhi, R., Hu, Y. C., & Zhang, M. (2016). Yoda: A highly available layer-7 load balancer. In *Proceedings of the Eleventh European Conference on Computer Systems* (pp. 1-16).
27. Esfandiari, H., Korula, N., & Mirrokni, V. (2015). Online allocation with traffic spikes: Mixing adversarial and stochastic models. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation* (pp. 169-186).
28. Olteanu, V. A., Huici, F., & Raiciu, C. (2015). Lost in network address translation: Lessons from scaling the world's simplest middlebox. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization* (pp. 19-24).
29. Lacity, M. C., & Reynolds, P. (2014). Cloud Services Practices for Small and Medium-Sized Enterprises. *MIS Quarterly Executive*, 13(1).
30. Rahman, M., Iqbal, S., & Gao, J. (2014). Load balancer as a service in cloud computing. In *2014 IEEE 8th international symposium on service-oriented system engineering* (pp. 204-211). IEEE.
31. Hong, Y. S., No, J. H., & Kim, S. Y. (2006). DNS-based load balancing in distributed Web-server systems. In The Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and the Second International Workshop on Collaborative Computing, Integration, and Assurance (SEUS-WCCIA'06) (pp. 4-pp). IEEE.
32. Ghutke, B., & Shrawankar, U. (2014). Pros and cons of load balancing algorithms for cloud computing. In 2014 International Conference on Information Systems and Computer Networks (ISCON) (pp. 123-127). IEEE.
33. Truong-Huu, T., Tham, C. K., & Niyato, D. (2014). To offload or to wait: An opportunistic offloading algorithm for parallel tasks in a mobile cloud. In 2014 IEEE 6th international conference on cloud computing technology and science (pp. 182-189). IEEE.
34. Garg, S., Gupta, D. V., & Dwivedi, R. K. (2016). Enhanced active monitoring load balancing algorithm for virtual machines in cloud computing. In 2016 International Conference System Modeling &Advancement in Research Trends (SMART) (pp. 339-344). IEEE.