Smart Internet of Things

www.siot.reapress.com

Smart. Internet. Things. Vol. 1, No. 1 (2024) 93–105.

Paper Type: Original Article

Load Balancing Demystified: A Comprehensive Study of Load Balancing Architectures in Cloud Computing

Malvika Narsipuram[1],*, Ananya Rai[1], Astha Tiwari[1]

[1] Department of Computer Engineering, KIIT University, Bhubaneswar -751024, Odisha, India; n.malvika2003@gmail.com; raiananya08@gmail.com.com; astha.tiwari0910@gmail.com.

Citation:

Received: 16 April 2024 *Revised: 11 July 2024* *Accepted: 28 August 2024*	Narsipuram, M., & Rai, A., Tiwari, A. (2024). Load balancing demystified : a comprehensive study of load balancing architectures in cloud computing. *Smart Internet of Things, 1* (1), 93-105.

Abstract

Load Balancing is the process of effectively dividing incoming network traffic among several backend servers. Essentially, there are three types in load balancing; software, hardware and cloud-native. In order to disperse network and application traffic, a hardware load balancer uses actual hardware that is located on the premises. There are two types of software load balancers: open-source and commercial. Both require installation before use. In contrast to software load balancers, a virtual load balancer (also known as cloud-native load balancer) installs a hardware load balancing device's software on a Virtual Machine (VM). The main challenge is the need to compare their performance across diverse network environments while considering factors such as scalability, cost-effectiveness, and adaptability to dynamic workloads. In our work we have proposed a comparative analysis of cloud load balancing that will help the research community to select appropriate cloud load balancing according to the applications. Software, hardware, and cloud-based each has its own advantages and limitations. Our results demonstrate that software load balancers offer flexibility and ease of deployment, hardware load balancers often provide high performance and dedicated hardware resources, while cloud-based load balancers offer scalability and integration with cloud services. Furthermore the "best" type varies depending on factors such as the scale of the network, budget constraints, performance requirements, and the specific needs of the application or service being load balanced.

Keywords: Load balancing, Cloud computing, Cloud-native solution, Software load balancing, Hardware load balancing, Scalability, IoT.

1 | Introduction

Cloud computing is a technology involving delivery of various computing services, including storage, processing power, databases and networking, over the internet. Instead of relying on a local server or personal computer to handle computing tasks, users utilize cloud services through remote servers hosted on the internet [1]. These servers are like diligent workers responsible for handling tasks like running applications or

✉ Corresponding Author: n.malvika2003@gmail.com

https://doi.org/10.22105/siot.v1i1.38

serving websites. Imagine these servers being left to their own devices, with no guidance on how to manage these tasks. This will lead to a situation depicted in *Fig. 1*, some servers may get bombarded with requests, while others stand by idly [2]. This imbalance will lead to inconsistency, deteriorated performance, and a risk of certain servers crashing under the weight of excessive workloads.

In this example, we want technology that manages the workload so that no server becomes underused, overloaded, or down. Load balancing is a method of effectively distributing network visitors amongst backend servers, ensuring that networking and obligations are well allotted across servers as shown in *Fig. 1*. This virtual visitors controller ensures that no server is overworked beyond its capability [3]. Load balancing increases device performance in a cloud surroundings, allowing customers to get entry to the cloud more quick by way of dispensing and dealing with server loads It returns all hundreds to a fixed of nodes with a configuration a compiled to gain most beneficial sources, improve response time, and save you some nodes from being exceeded [4].

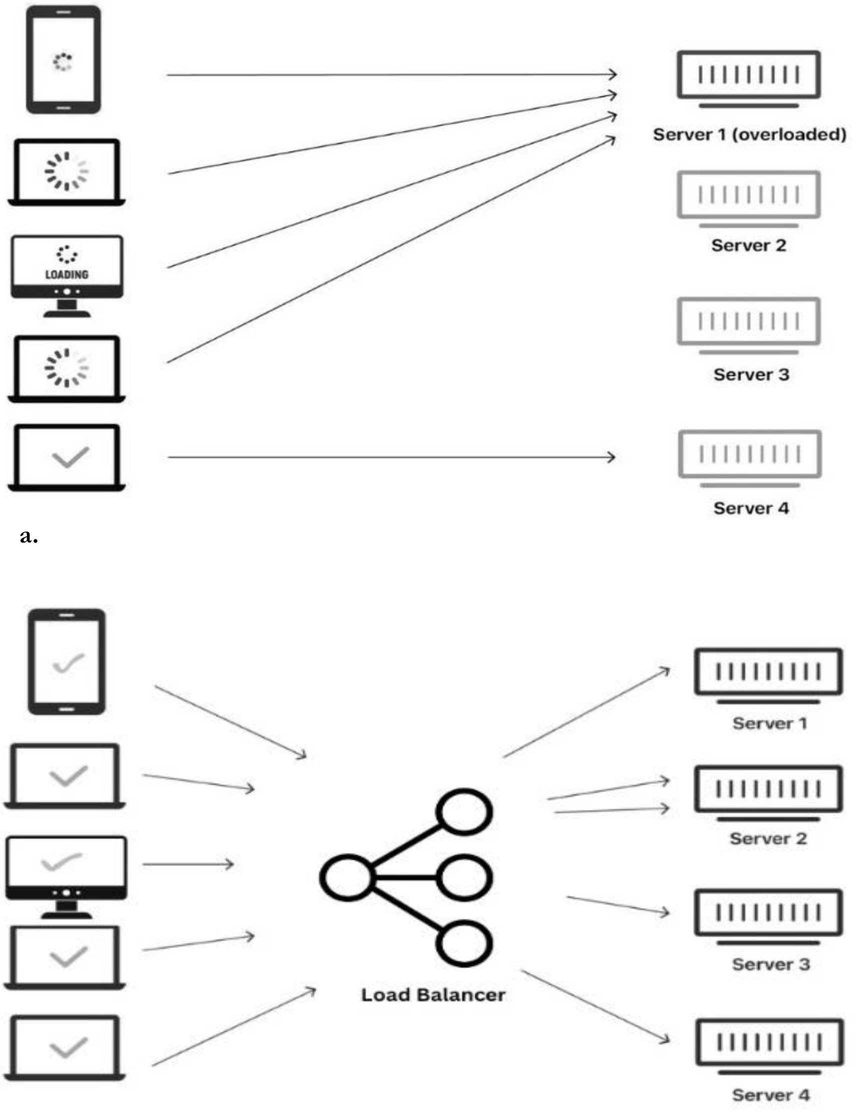

Fig. 1. a. Traffic due to absence of load balancing, b. Load Balancers distributing load evenly.

Many popular websites receive thousands of requests from multiple users in a second. These websites use a device called Load Balancer to distribute these requests smoothly across different servers ensuring that all servers are utilized efficiently, preventing any single server from becoming overloaded [5]. Load Balancers also offer backup capabilities, redirecting requests to another server if one crashes. This ensures high availability and reliability since data is only directed to active servers by the load balancer [6].

Load balancers can be physical devices, software instances, or a combination of both. Depending on the specific network requirements, various types of load balancers can be deployed, each with different storage capacities, functionalities, and complexities. Traditionally, there have been two major types of load balancers: hardware-based and software-based [7]. Hardware-based load balancers are dedicated physical devices with specialized hardware components designed for handling large volumes of traffic. Software load balancers are computer programs or services that run on regular servers. They utilize the computing power and resources of standard hardware setups. Nowadays, combined hardware and software load balancing solutions are being used to benefit from the strengths of each approach [8]. The functioning of a load balancer is depicted in *Fig. 2*.

The current situation shows an increase in the use of a different type of load balancer called a cloud-based load balancer. This is a virtual service provided by cloud service providers that can automatically scale to distribute traffic across virtual instances. Unlike traditional load balancers, these operate on a pay-as-you-go model, focusing on features like global load balancing, enhanced security, and integration with container orchestration platforms [9]. This helps improve performance and availability without needing upfront hardware investments. Companies are taking advantage of these solutions to align with modern cloud architectures, support dynamic workloads, and optimize resource use [10]. Cloud-native load balancing solutions help enterprises achieve increased performance capabilities and possibly reduces costs when compared to traditional load balancing technologies.

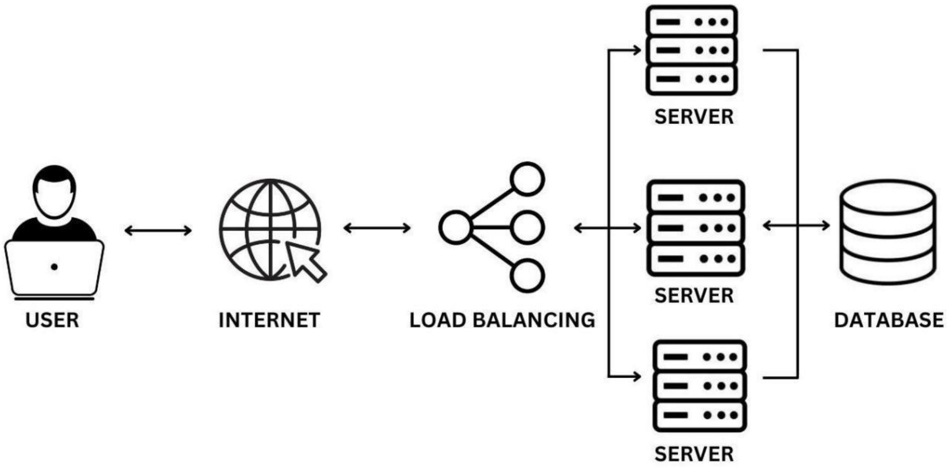

Fig. 2. Working of load balancers.

2 | Literature Review

Cloud computing is a system where multiple resources work together, sharing and combining their capabilities. As more people use cloud services, the related distributed systems become increasingly complex. This involves many different types of systems, so efficient load balancing techniques are crucial. Sometimes, many systems can overwhelm the target web server with numerous requests at once. If the server is overloaded with connections, it won't be able to process new ones [11].

To maximize performance, minimize server response time, and distribute traffic evenly across multiple servers, load balancers are essential. Previous research has examined various load balancers available, such as hardware, software, and virtual load balancers. These load balancers significantly improve performance by utilizing server instances optimally, ensuring no server in the cluster is overloaded [12].

A physical device called a hardware load balancer is used to distribute web traffic across multiple network servers. This routing can be based on available server connections, processing power, and resource usage, or it can be randomly determined (like round-robin). The primary objective of using load balancing in between servers and clients is to ensure flexibility and scalability [3].

Proper request distribution also guarantees consistent performance for every user, preventing inaccessibility caused by a single server failure. Different routing approaches and algorithms are used in various load balancing scenarios to ensure optimal performance. To enhance the performance and usability of hardware load balancing at lower costs and effort, Load Balancing as a Service (LBaaS), popularly known as Cloud load balancing was discovered [13]. It offers benefits like worldwide server load balancing and suitability for widely distributed environments [14]. Cloud load balancers are convenient because they don't require physical hardware, so they can easily scale up or down as needed. In contrast, Hardware Load Balancers (HLDs) have fixed limitations. To increase capacity with HLDs, you'd have to order, ship, install, and maintain more appliances, which costs money and reduces productivity. Hardware load balancers can only work with other appliances, not cloud servers. However, cloud load balancers can integrate with both hardware load balancers and cloud servers, providing more flexibility [15].

As observed from various scheduling techniques, a single algorithm cannot solve all challenges. Some algorithms focus on factors like energy, cost, time, and Quality of Service (QoS), whereas some algorithms consider different aspects like resource utilization, response time, scalability and availability. Future studies should take into consideration these limitations as well as discover new possibilities arising from recent developments in this area of study [16].

A major concern for IoT systems is maintaining end-to-end security as multiple technologies like sensor and wireless networks, RFID devices, and public and private clouds with their helpers like data centers and edge nodes are connected to them. Current security methods for networks involve encryption and authentication, which help prevent attacks from outsiders. However, additional techniques are needed to stop malicious attacks from insiders [17]. This requires developing secure protocols that can authenticate events triggering network functions, stopping unauthorized changes. Load balancing, a key part of distributed systems, is crucial here. It divides workload across nodes to boost efficiency and response time, ensuring no node is overloaded while others sit idle. This guarantees each processor or network node performs a similar amount of work at any given time [18].

To better understand how software-based load balancing works, a research paper [19] compares different commercial software-based load balancers. First, the AVI Vantage's network load balancer, a full software load balancer provides an intelligent Web Application Firewall (iWAF) and Elastic Service Mesh [20]. Next is the Barracuda Load Balancer ADC is designed to maximize application performance by offloading compute-intensive SSL transactions, freeing up server resources for the applications. Additionally, the Barracuda load balancer uses optimization technologies like caching, compression, and TCP pooling to ensure scalability and faster application delivery [21].

According to an article [10], F5 offers intelligent and customized load balancing strategies to analyze and route clients to available resources, helping free up busy sites and systems. In today's application-centric environment, A10's server load balancing solutions assist in meeting availability demands, ensuring security, and improving user experience [22]. Another article [12] explains that the Citrix ADC BLX, a 100% software load balancer, does not require any certifications as it runs on Linux Virtual Machines (VMs) as a software program. It is the software version of the Citrix ADC MPX, a physical appliance that provides robust hardware-based application delivery and load balancing with options for high-performance web application security and SSL offload support. A journal [23] provides a comparison of different load balancing techniques currently in use, along with a compilation of experimental findings based on criteria like response time and processing time. *Table 1* shows the hardware and cloud load balancer.

Table 1. Hardware and cloud load balancers.

Hardware Load Balancer	Cloud Load Balancer
High CAPEX	Low CAPEX
The OPEX/Maintenance vary from low to high	The OPEX/Maintenance is low
Distribution algorithms are arbitrary or data driven	Distribution algorithms are arbitrary or data driven
Load distribution is present in network layer and application layer	Load distribution is present in network layer and application layer
TTL reliance is there (in the DNS/GSLB scenarios)	TTL reliance is not there
High scalability	Low scalability
It is compatible with server	It is compatible with server, cloud, and hybrid

3|Basic Concepts

3.1|Hardware Load Balancing

Hardware load balancing is a type of physical equipment specifically designed to manage and distribute user requests among a complex network of computers. They usually have privately owned operating systems running on their devices. In contrast, software load balancing techniques are compatible with any existing architecture [24]. Today, most of the major load balancers utilize x86-based computer networks. They don't have any specific hardware requirements, they only need some pre-configuration for functioning properly.

A major distinction between software and hardware load balancers is the scalability factor. While hardware load balancers are static and service operations require additional hardware from the data center, software options can scale elastically as demand increases and ultimately software solutions are customizable [25]. They can be individually configured and modified, and integrate seamlessly with intricate hybrid and multi-cloud architectures. This advantage is not available with hardware load balancing.

3.2|Software Load Balancing

Administrators can distribute network traffic to other servers using software load balancing. To check client requests, load balancers check application-level information such as IP address, HTTP header, and the content of the request. After examining the servers, the load balancer selects the most appropriate one and forwards the request to that server [26]. Usually, software-based solutions are provided by the Application Delivery Controller (ADC), which runs on a VMs or on a standard server. Unlike HLD, software load balancer doesn't require any dedicated load-balancing device even though it provides the same functionality. The load balancing software is able to operate on any regular or virtual server [27].

Scalability is the primary benefit of software-based load balancing over hardware-based load balancing. A software-based load balancer can dynamically and instantaneously adapt to changes in network traffic by adding or removing virtual servers based on demand [28]. Because they work in a wide variety of environments, they are even more scalable than hardware load balancers. Standard desktop OSs, remote servers, cloud services, bare metal, and containers can all be configured to work with them. Because hardware load balancers cannot be programmed, they are less flexible [29].

Software load balancers can help businesses save money, particularly when the companies employ load balancing as a service or lbaas. despite the fact that if an IT company buys its own software load balancers, it will count as OPEX rather than CAPEX. Software load balancers save time and money since they are simple to deploy and on demand [30]. Last but not least, load-balancing software provides an additional degree of protection by having the capacity to discard erroneous data packets even before they get to the server by residing between the client and the server.

3.3 | Cloud Load Balancing

More and more companies are resorting to cloud-native solutions to guarantee higher availability and better performance, since the amount of internet traffic and workloads continues to rise. Cloud computing can be defined as the process of dividing up workloads, traffic, and user requests across multiple servers running on a cloud environment [31]. This technique makes sure that every cloud resource has a load that it can reasonably handle, preventing computers or servers in a cloud environment from being either overloaded or underutilized. As a result, cloud optimization is enhanced [32].

With the aid of efficient load balancing, businesses can meet the demands of cloud-based applications meanwhile reducing latency, boosting reliability, optimizing performance, and preventing downtime. As-needed cloud load balancing is provided via Cloud Load Balancing as a Service, or LBaaS, which takes the place of dedicated, on-premises hardware [33]. LBaaS provides increased scalability since load balancing in the cloud can handle traffic spikes without requiring the reconfiguration of physical infrastructure. Connecting to the nearest servers guarantees higher availability, and LBaaS can save maintenance and investment costs in comparison to hardware-based solutions [34]. *Table 2* provides a comparative analysis of different types of load balancing in a concise manner.

So, how does cloud load balancing work? Load balancers are used in the cloud between backend servers and client machines. They make use of algorithms that consider different factors like server load and geographic distance, to divide requests efficiently. These algorithms typically fall into two categories, namely static and dynamic [35]. Static algorithms include algorithms such as the Round Robin algorithm, which distributes requests among cloud servers in a basic, recurring sequence.

Weighted Round Robin provides a higher weight to servers that have more capacity, allowing them to handle a large volume of incoming application traffic. IP Hash is a service that maps a client's IP address to specific servers by performing a hash, or mathematical computation, on the address [36]. Dynamic algorithms include algorithms such as the least connections algorithm which distributes workload between the servers having the lowest number of active connections. least response time distributes traffic among the servers having lowest average response time and fewest connections. Lastly, least bandwidth sends requests to the servers consuming the lowest amount of bandwidth during a recent period, etc [14], [15].

Table 2. A comparison between various kinds of load balancers.

	Hardware Load Balancer	Software Load Balancer	Cloud-based Load Balancer
Cost	It can be expensive, especially for high-performance models.	It is generally more affordable than hardware load balancers.	It can be cost-effective since users only have to pay for the resources they use.
Performance	It has high performance as they are optimized for load balancing tasks.	It may have lower performance as compared to hardware load balancer, especially under heavy loads.	Usually depends on the cloud provider but it is designed to provide high performance to efficiently distribute incoming traffic.
Scalability	Scalability is limited since enhancing capability requires buying more hardware.	It can be easily scaled by adding more resources or upgrading the underlying hardware.	It is highly scalable as they can easily accommodate changes in traffic and resource demands.

Table 2. Continue.

	Hardware Load Balancer	Software Load Balancer	Cloud-based Load Balancer
Maintenance	It needs ongoing configuration and maintenance but may require specialized knowledge.	It may require ongoing software updates and maintenance which requires technical knowledge and automation tools.	Simplified management as the cloud provider takes care of maintenance and updates but may have less control over configuration and customization.
Flexibility	It is generally less flexible as compared with other types of load balancing since it often comes with limited configurations.	It provides moderate flexibility as it can be deployed on a variety of platforms and environments including cloud-based infrastructure.	It offers unparalleled flexibility due to its virtualized and dynamic scalable nature and various configurable options.
Example	A hardware load balancer can be used by a major online retailer to split up incoming web traffic among several web servers, guaranteeing fast response time and a seamless online shopping experience for its users.	A startup with a growing user base can deploy a software load balancer on a virtual machine hosted in the cloud, distributing incoming requests among multiple application servers to handle increased traffic.	To ensure smooth app performance and fast responses, a mobile app developer can employ a cloud-based load balancer offered by any cloud provider to distribute incoming API calls among several backend servers.

There are many types of cloud load balancing, a few of them have been covered below. First we have DNS or Domain Name System load balancing that relies on the DNS infrastructure to distribute incoming traffic among multiple servers or resources. It works by providing a domain name to multiple IP addresses, effectively directing clients to different servers based on various policies. It is relatively simple to implement, as it doesn't require any specialized hardware or software. It provides basic load balancing and failover capabilities. It can also distribute traffic across geographically distributed servers, improving performance for users in different regions [37].

There are some disadvantages associated with DNS load balancing. It is restricted to DNS resolution time, which updates more slowly than other load-balancing methods. Response time, resource usage, and server health are not taken into account. Applications needing fine-grained load distribution or session persistence can find DNS load balancing unsuitable [17]. DNS load balancing can be used by a Content Delivery Network (CDN) to direct users to the closest edge server based on their geographical location, ensuring faster content delivery and reduced latency [38]. Next we have Global Server Load Balancing (GSLB) that is a technique used to distribute traffic across geographically dispersed data centers.Integrating DNS load balancing with other cutting-edge features like health checks, offers a more clever and effective way to distribute traffic. Unlike DNS load balancing, it provides load balancing and failover capabilities across multiple data centers or geographic locations, can also improve performance and reduce latency for users by directing them to the closest or best-performing data center and supports advanced features, such as server health checks, session persistence, and custom routing policies. But it comes with some disadvantages.

It can be more complex to set up and manage than other load balancing techniques [39]. It may also require specialized hardware or software, increasing costs. It can be subject to the limitations of DNS, such as slow updates and caching issues. A global company may utilize GSLB to split up incoming requests for its web applications among multiple data centers globally, guaranteeing optimal performance and high availability for customers in various regions. To obtain the highest potential speed, scalability, and reliability, multiple load balancing approaches are combined in a different sort of cloud load balancing called hybrid load balancing. To provide the most efficient and adaptable load-balancing technique for a given circumstance, it usually combines hardware, software, and cloud-based solutions. Main advantage of hybrid load balancing is that it can provide the best combination of performance, scalability, and reliability by leveraging the strengths of different load balancing techniques. Additionally, it provides a great deal of flexibility because it can be customized to fit particular demands and infrastructure, enabling businesses to modify and enhance their load-balancing plan as circumstances change.

It can be more difficult to set up, administer, and manage than single-technique solutions since it combines a variety of load-balancing strategies, and it might call for a greater degree of skill and knowledge of these techniques. Possibly more expensive because it requires a mix of cloud-based services, software, and hardware. A hybrid load-balancing approach can be used by any major online streaming platform. It combines DNS load balancing for global traffic management, cloud-based load balancers for scalable content delivery, and hardware load balancers in data centers for high-performance traffic distribution. Millions of users worldwide are guaranteed excellent speed, scalability, and dependability using this strategy.

There are two more types of cloud load balancing that are not widely known. The first one is Transport Layer load balancing, popularly known as Layer 4 load balancing. It functions at the fourth tier of the OSI model, namely the transport layer. Based on data from the TCP or UDP header, including source and destination IP addresses and port numbers, it divides incoming traffic. Because it makes choices based only on limited information from the transport layer, it operates extremely quickly and effectively. It can also manage a large range of traffic kinds and protocols. It's comparatively easy to implement and administer. However, there are some drawbacks. It is less effective in some situations due to unawareness of application-level information.It might not be appropriate for applications that need fine-grained load distribution or session persistence since it doesn't account for resource usage, response time, or server health.

To ensure that players are evenly divided among the available game servers for seamless gameplay, online gaming platforms primarily use it to distribute game server traffic based on IP addresses and port numbers. Next we have application layer load balancing, commonly known as Layer 7 load balancing. As evident from its name, this type of load balancing works on the application layer of the OSI model, i.e., the seventh layer. To distribute incoming traffic more effectively, it considers application-specific data, including HTTP headers, cookies, and URL routes. It offers load balancing that is more intelligent and precise since it takes application-level data into account.

It also supports advanced features, such as session persistence, content-based routing, and SSL offloading, and can be tailored to specific application requirements and protocols However, because it needs a more thorough examination of incoming data than Layer 4 load balancing, it is slower and takes more resources. Specialized hardware or software may also be needed to manage application-level traffic processing and inspection. When compared to other load-balancing strategies, it is more difficult to configure and maintain. Layer 7 load balancing can be used by a web application with several microservices to route incoming API calls according to the URL path, guaranteeing that each microservice receives only the requests it is in charge of processing.

4 | Proposed Study

4.1 | Research Gap

Cloud computing uses technology to support people and systems. The primary goal of cloud computing is to share resources, software, and information online in order to reduce capital and operational costs, improve response and data processing times, improve system stability, and allow for future system modifications. As a result, there are many technical issues that need to be addressed, such as server consolidation, fault tolerance, high availability, and scalability. Researchers are trying to find ways to maintain load during VM migration and avoid overloading systems [5]. Some other future challenges include the possibility of VM-based malware, targeted attacks (shared-tenancy environment), service provider trust, vendor lockouts, data security if there are many layers, problems with botnet hosting,etc. [14], [19].

Eventually, the Internet of Things (IoT) will shape the future of cloud computing, with the obvious challenge of preventing it from becoming the Internet of Over Things (IoOT) and above [18]. Research needs to be done to exhaust challenges posed by the Internet underneath and serious new cyberattack threats will go after new connected devices that are more susceptible. Another key issue in load balancing in cloud computing is how to manage complexity as networks grow exponentially every day. Managing cryptocurrencies is another future problem associated with the cloud environment, because future currency markets are slowly moving towards digital currencies [20].

SI methods by processing requests, their exchange and availability of many VMs can contribute to resources the use of which has increased and reduced the response time so the integration of remote data centers SI techniques with clouds in the common terrain will be the focus of future studies. Therefore, it is important to provide SI techniques such as autonomous multi-objective scheduling to optimize the competing objectives in the least possible time [13].

4.2 | Problem Formulation

The increasing number of clients and networks in this era is making it challenging to access records. Cloud distribution systems allow the customer to apply all of their services and execute their applications for less expenditure than they would need to pay for the underlying structure [21]. The computational component is probably overloaded with multiple users requesting to run their programs. There are several techniques in present structures that use stability masses to obtain either the initial node frame or dynamic node load [22].

The dynamic state of the nodes and system rearrangement so as of the burden percent are nevertheless taken into account within the gift machine algorithms. The existing technique's dynamic algorithms consequently do not provide better output, response times, lower prices, or less delays. Performance optimization, reliability, security and scalability are some of the key challenges associated with load balancing [23].

This research has a goal of answering a set of questions that were identified from literature surveys, these questions need to be answered before going into the load balancing process. Some of these questions have been answered in literature while others still remain unanswered. The questions are given as follows:

I. What causes the load unbalancing problem? This question addresses the causes behind load unbalancing problems. It requires a thorough study of each of the factors leading to load unbalancing.

II. Why is load balancing so important in cloud computing? This question explores the problems and challenges faced by cloud service providers.

III. How do different types of load balancers (software-based, hardware-based, cloud-based) compare in terms of performance and scalability?

IV. What are the main load balancing algorithms used in modern load balancer systems?

After these questions, it is important to dig deeper into the realm of load balancing as these will help us to understand things clearly. By carefully examining and answering each question, we aim to gain a better understanding of our topic [8]. Through this process, we aim to gain a comprehensive understanding that comes only from beyond surface-level observations. This section serves as a roadmap for this study, as it gives us a glimpse of where we are headed and what we might find along the way.

4.3 | Potential Solutions

After comparing and studying different load balancers, we have prepared *Table 3* comparing different load balancers available in the market.

Table 3. Comparison of different load balancers.

Feature	AVI Vintage	Barracuda Load Balancer	A10 Load Balancer	F5 Load Balancer	Citrix BLX	Cloud Load Balancer
Type of design(hardware/software)	100% Software	Available in both types	Hardware and software	Both	Software	Cloud based software
Central management	Yes	Yes	Yes	Yes	Yes	Yes
Autoscaling	Yes	Can be scaled on demand	Via virtual ADC	On demand	Yes	Yes
HA	Yes	Active-passive pair	Active-active Active-standby	Yes	Yes	Yes
Compatibility	Yes, it is multi infrastructure	Yes, it is multi infrastructure	Yes, it is multi infrastructure	Yes, it is multi infrastructure	Yes, it is multi infrastructure	Yes, it is multi-cloud infrastructure
Automation (available VIA)	Ansible	Puppet/Chef	Ansible	F5 automation	Ansible	Ansible

5 | Conclusion and Future Scope

In conclusion, Cloud computing technology is made up of a large number of resources. It allows the users to access the correct and essential resources in a network as required. Applications with various resource demands require high performance computing capabilities [28]. This helps to maximize resource utilization, time consumption, and avoid processor overhead [29]. Through our research, we investigated different types of load balancers, including software-based, hardware-based and cloud-based solutions. Each approach offers unique advantages and designs that address different application scenarios and needs. This in-depth analysis reveals their strengths, weaknesses, and suitability for different strategies.

In addition, this paper discusses the challenges and considerations in load balancing design and implementation, including factors such as scalability, latency, fault tolerance, and security [31]. It also focuses on the role of emerging technologies such as machine learning and artificial intelligence to improve load balancing mechanisms and dynamically optimize changing workloads Load balancing plays an important role in optimizing distributed system performance and reliability and once developed to some extent, some areas for further development are included matter in the paragraph.

Integration with Edge Computing: With the rise of edge computing and the growth of Internet of Things (IoT) devices, there is a growing need for load balancers optimized for edge environments. Auto-scaling and Self-Healing Mechanisms: Future research can explore techniques for building intelligent load balancers that dynamically scale resources up or down based on demand and automatically detect and recover from failures to ensure uninterrupted service availability. Multi-cloud Load Balancing: With the increasing adoption of multi-cloud and hybrid cloud architectures, there is a need for load balancers capable of distributing traffic across multiple cloud providers and on-premises data centers [31]. Security and Privacy Enhancements: Load balancers are critical components in ensuring the security and privacy of distributed systems. Future research can explore novel techniques for load balancer security, including protection against attacks, intrusion detection, and encryption [32].

In conclusion, our research underscores the importance of load balancers in modern IT infrastructures and provides valuable insights for organizations seeking to implement effective load balancing solutions [33]. By choosing the right load balancer technology and best practices, businesses can achieve enhanced performance, reliability, and scalability for their applications, thereby meeting the evolving demands of today's digital landscape.

Author Contributions

Malvika Narsipuram: Conceptualization, methodology, writing – original draft, supervision.

Ananya Rai: Data collection, formal analysis, and writing – review & editing.

Astha Tiwari: Visualization, writing – review & editing, and project administration.

Funding

No specific funding was received for this research.

Data Availability

The data supporting the findings of this study are available upon reasonable request from the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest in relation to this work.

References

[1] Rani, S., Kumar, D., & Dhingra, S. (2022). A review on dynamic load balancing algorithms. *2022 international conference on computing, communication, and intelligent systems (ICCCIS)* (pp. 515–520). IEEE.

[2] Rai, H., Ojha, S. K., & Nazarov, A. (2020). Comparison study of load balancing algorithm. *2020 2nd international conference on advances in computing, communication control and networking (ICACCCN)* (pp. 852–856). IEEE.

[3] Rahman, M., Iqbal, S., & Gao, J. (2014). Load balancer as a service in cloud computing. *2014 IEEE 8th international symposium on service oriented system engineering* (pp. 204–211). IEEE. https://ieeexplore.ieee.org/abstract/document/6830907/

[4] Mohapatra, H., & Rath, A. K. (2020). Fault-tolerant mechanism for wireless sensor network. *IET wireless sensor systems*, *10*(1), 23–30. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2019.0106

[5] Moharir, M., Shobha, G., Oppiliappan, A., GVL, R. M. K., Pandit, S. N., Akash, R., & Saxena, M. (2020). A study and comparison of various types of load balancers. *2020 5th IEEE international conference on recent advances and innovations in engineering (ICRAIE)* (pp. 1–7). IEEE. https://ieeexplore.ieee.org/abstract/document/9358333/

[6] Imperva. (2019). *Load balancer hardware*. https://www.imperva.com/learn/availability/hardware-load-balancer-hld/

[7] Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE-LEACH protocol. *IET wireless sensor systems*, *9*(6), 358–365. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2018.5229

[8] LoadBalancerADC. (2024). *Barracuda networks*. https://www.barracuda.com/products/application-protection/load-balancer

[9] Load Balancing Your Applications. (2024). *Matters, Why Intelligent Load Balancing*. https://www.f5.com/solutions/use-cases/load-balancing-your-applications

[10] Balancing, L. (2018). *Networks, A10*. https://www.a10networks.com/solutions/multi-cloud/load-balancing/

[11] Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. *IET wireless sensor systems*, *9*(6), 447–457. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-wss.2019.0081

[12] Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. *Journal of king saud university - computer and information sciences*, *34*(7), 3910–3933. https://www.sciencedirect.com/science/article/pii/S131915782100046X

[13] Shona, M., & Sharma, R. (2023). Implementation and comparative analysis of static and dynamic load balancing algorithms in sdn. *2023 international conference for advancement in technology (ICONAT)* (pp. 1–7). IEEE. https://ieeexplore.ieee.org/abstract/document/10080430/

[14] Deepa, T., & Cheelu, D. (2017). A comparative study of static and dynamic load balancing algorithms in cloud computing. *2017 international conference on energy, communication, data analytics and soft computing (ICECDS)* (pp. 3375–3378). IEEE. https://ieeexplore.ieee.org/abstract/document/8390086/

[15] Mohapatra, H., & Rath, A. K. (2020). Survey on fault tolerance-based clustering evolution in WSN. *IET networks*, *9*(4), 145–155. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-net.2019.0155

[16] Contributor, S. (2020). *solarwinds serv-U FTP server overview, pricing and download 2024*. https://www.dnsstuff.com/what-is-server-load-balancing

[17] Afzal, S., & Kavitha, G. (2019). Load balancing in cloud computing–A hierarchical taxonomical classification. *Journal of cloud computing*, *8*(1), 1–24. https://doi.org/10.1186/s13677-019-0146-7

[18] Joshi, S., & Kumari, U. (2016). Load balancing in cloud computing: challenges & issues. *2016 2nd international conference on contemporary computing and informatics (IC3I)* (pp. 120–125). IEEE. https://ieeexplore.ieee.org/document/7917945

[19] Ebadifard, F., & Babamir, S. M. (2021). Autonomic task scheduling algorithm for dynamic workloads through a load balancing technique for the cloud-computing environment. *Cluster computing*, *24*(2), 1075–1101. DOI:10.1007/s10586-020-03177-0

[20] Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: issues and challenges. *Journal of grid computing*, *14*(2), 217–264. DOI:10.1007/s10723-015-9359-2

[21] Houssein, E. H., Gad, A. G., Wazery, Y. M., & Suganthan, P. N. (2021). Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. *Swarm and evolutionary computation*, *62*, 100841. https://www.sciencedirect.com/science/article/pii/S221065022100002X

[22] Mohapatra, H., & Rath, A. (2020). Fault Tolerance in WSN Through Uniform Load Distribution Function. *International journal of sensors wireless communications and control*, *10*. DOI:10.2174/2210327910999200525164954

[23] Varghese, B., & Buyya, R. (2018). Next generation cloud computing: new trends and research directions. *Future generation computer systems*, *79*, 849–861. https://www.sciencedirect.com/science/article/pii/S0167739X17302224

[24] Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). Future trends in cloud computing. In *Cloud computing with security: concepts and practices* (pp. 235–259). Cham: Springer International Publishing. DOI: 10.1007/978-3-030-24612-9_13

[25] Buyya, R., Srirama, S. N., Casale, G., Calheiros, R., Simmhan, Y., Varghese, B., & Shen, H. (2018). A manifesto for future generation cloud computing: research directions for the next decade. *ACM comput. surv.*, *51*(5). DOI:10.1145/3241737

[26] Bhari, S., & Quraishi, S. J. (2022). Blockchain and cloud computing-a review. *2022 international conference on machine learning, big data, cloud and parallel computing (Com-It-Con)* (Vol. 1, pp. 766–770). IEEE. https://ieeexplore.ieee.org/abstract/document/9850499/

[27] Abdalla, P. A., & Varol, A. (2019). Advantages to disadvantages of cloud computing for small-sized business. *2019 7th international symposium on digital forensics and security (ISDFS)* (pp. 1–6). IEEE. https://ieeexplore.ieee.org/abstract/document/8757549/

[28] Agarwal, M., & Srivastava, G. M. S. (2017). Cloud computing: a paradigm shift in the way of computing. *International journal of modern education and computer science*, *9*(12), 38. https://www.mecs-press.net/ijmecs/ijmecs-v9-n12/IJMECS-V9-N12-5.pdf

[29] Domanal, S. G., & Ram Mohana Reddy, G. (2014). Optimal load balancing in cloud computing by efficient utilization of virtual machines. *2014 sixth international conference on communication systems and networks (Comsnets)*. IEEE. DOI:10.1109/COMSNETS.2014.6734930

[30] Metawei, M. A., Ghoneim, S. A., Haggag, S. M., & Nassar, S. M. (2012). Load balancing in distributed multi-agent computing systems. *Ain shams engineering journal*, *3*(3), 237–249. https://doi.org/10.1016/j.asej.2012.03.001

[31] Liu, W., Wu, M., Ou, X., Zheng, W., & Shen, M. (2000). Design of an i/o balancing file system on web server clusters. *Parallel processing, 2000. proceedings. 2000 international workshops on* (pp. 119–125). IEEE Xplore. DOI: 10.1109/ICPPW.2000.869095

[32] Waghmode, S. T., & Patil, B. M. (2021). Load balancıng technique in distributed systems: a review. *2021 2nd global conference for advancement in technology (Gcat)* (pp. 1–7). IEEE. https://ieeexplore.ieee.org/abstract/document/9587476/

[33] Nguyen, V. H., Khaddaj, S., Hoppe, A., & Oppong, E. (2011). *A qos based load balancing framework for large scale elastic distributed systems* [presentation]. 2011 10th international symposium on distributed computing and applications to business, engineering and science (pp. 146–150). DOI: 10.1109/DCABES.2011.12

[34] Galante, G., & de Bona, L. C. E. (2012). A survey on cloud computing elasticity. *2012 ieee fifth international conference on utility and cloud computing* (pp. 263–270). IEEE. https://ieeexplore.ieee.org/abstract/document/6424959/

[35] Ray, S., & De Sarkar, A. (2012). Execution analysis of load balancing algorithms in cloud computing environment. *International journal on cloud computing: services and architecture (IJCCSA)*, *2*(5), 1–13. https://www.academia.edu/download/38386371/33.pdf

[36] Chandrasekaran, K., & Divakarla, U. (2013). *Load balancing of virtual machine resources in cloud using genetic algorithm* [presentation]. ICCN conference at national institute of technology karnataka, surathkal (pp. 156–168). https://www.researchgate.net/profile/Usha-Divakarla/publication/f

[37] Semchedine, F., Bouallouche-Medjkoune, L., Sayeh, O., Ayoub, S., & Aïssani, D. (2014). DNS-based load balancing with cache for geographically distributed web server systems. *2014 global summit on computer & information technology (GSCIT)* (pp. 1–6). IEEE. https://ieeexplore.ieee.org/abstract/document/6970100/

[38] Jung, J., Kiertscher, S., Menski, S., & Schnor, B. (2014). Self-adapting load balancing for dns. *International symposium on performance evaluation of computer and telecommunication systems (Spects 2014)* (pp. 564–571). IEEE. https://ieeexplore.ieee.org/abstract/document/6879994/

[39] Yokota, H., Kimura, S., & Ebihara, Y. (2004). A proposal of dns-based adaptive load balancing method for mirror server systems and its implementation. *18th international conference on advanced information networking and applications, Aina 2004.* (Vol. 2, pp. 208–213). IEEE. https://ieeexplore.ieee.org/abstract/document/1283788/