A REVIEW ON EXISTING CLOUD LOAD BALANCING TECHNIQUES

*Nimmy Francis** [1], *N. V. Balaji* [2]

ABSTRACT

Cloud computing is a set of services provided via the internet, or "the cloud." It involves the use of remote servers in order to store and retrieve data rather than using local hard drives and internal data centres.

Pre-cloud computing, organizations needed to buy and maintain their own servers for business requirements. It involved purchasing sufficient server space to reduce downtime and outages, and to support peak traffic volume. Consequently, vast amounts of server space were left idle for most of the time. Cloud service providers today enable enterprises to minimize the need for on-site servers, maintenance staff, and other expensive IT resources.

Load balancing refers to the method of distributing incoming network traffic evenly across multiple servers or computing resources to ensure no single system is overwhelmed. It plays a vital role in optimizing resource utilization, enhancing system responsiveness, and maintaining service reliability. By dynamically reallocating workloads, it enables organizations to meet varying performance demands and application requirements efficiently.

In cloud computing environments, load balancing is especially crucial, as it helps manage traffic across distributed resources in real time. It not only facilitates better handling of peak loads but also supports fault tolerance by redirecting traffic from failed or underperforming servers to healthy ones. This proactive management helps minimize downtime, ensures high availability of services, and enhances the end-user experience.

Furthermore, cloud load balancing contributes to improved scalability by allowing infrastructure to expand or contract based on demand without compromising performance. It also plays a role in securing the system by isolating workloads and preventing malicious traffic from impacting critical services. As a result, load balancing serves as a foundational element in modern cloud infrastructure, supporting both operational efficiency and robust service delivery.

Keywords: Load balancing, Cloud computing

I. INTRODUCTION

Cloud Load balancing is a form of load balancing. In this, workloads and computing characteristics are shared in cloud on various resources. It supports the management of workload requirements or application requirements by dividing the resources between various servers, networks or systems [5]. Cloud load balancing encompasses retaining the processing of workload traffic that occur over the Internet. It saves costs in terms of document management systems, and enhances the sharing of loads between multiple computing resources and overall application performance by mitigating the load on servers for managing and maintaining applications and networks [6]. The internet traffic is increasing very swiftly year after year, which is around 100% per annum of the current traffic. Therefore, the server workload increases at a faster rate and, as such results in overloading servers, primarily for highly demanded web server.

II. LOAD BALANCING ARCHITECTURE

A typical load balancer employed in cloud systems is shown in Figure 1, where the load balancer divides the load according to the standard procedures listed below [1]:

- Gets incoming service requests from a large number of customers.

Department of Engineering[1],
Amaljyothi College of Engineering,Kanjirapally, Kerala[1]
gardensenimmy @gmail.com
Department of Science, Commerce and Management[2],
Karpagam Academy of Higher Education, Coimbatore[2]

* Corresponding Author

The server monitor daemon calculates the load size requested for inbound load requests generated by clients and periodically verifies and checks the status of load in the server pool.

- Uses load-balancing technique, algorithm, or heuristic to choose the best server.

Millions of data packets are routed per second by the advanced network computing system. To ensure that the servers can handle the load without affecting the end user, this massive data traffic needs to be divided among the supplied servers in an effective way. To oversee this server load balancing among the servers, some servers have been designated. One of the main reasons for load balancing is to maintain high availability [1].

The following are some of load balancing's main advantages:

- Aids in traffic monitoring and control; Improves resource availability and utilization; Distributes network load according to node capacity; and Reduces infrastructure over provision.

Offering on-demand services, quick elasticity, or just scaling up or down in response to needs are the objectives of cloud computing architecture [1].

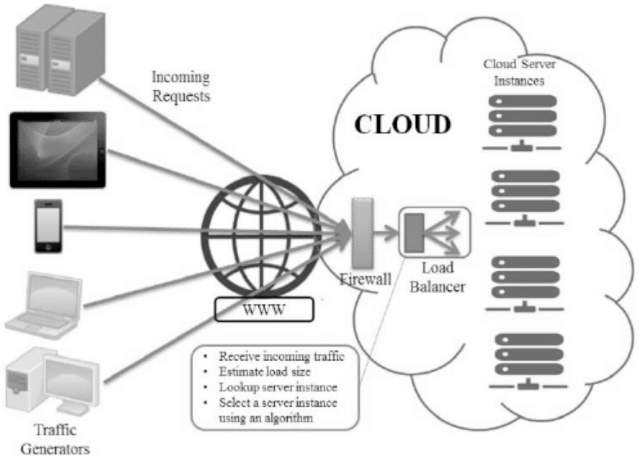

Figure 1: Cloud Load Balancing

III. CLASSIFICATION AND SOLUTIONS FOR LOAD BALANCING

As depicted in Figure 2, there are six categories into which the current load balancing solutions can be divided. The parts that follow go into further information about each of these.

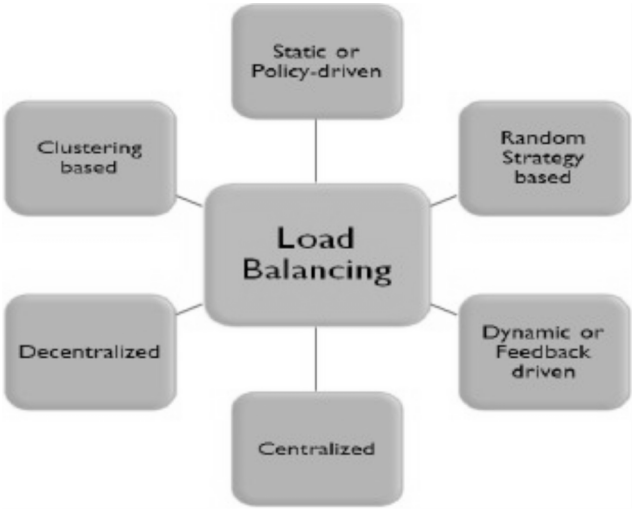

Figure 2: Load Balancing Solutions Classification

A. Static Load-Balancing

Static load balancing, also referred to as policy-driven load balancing, operates based on a predefined set of rules and configurations. These policies are typically defined by evaluating several parameters, such as server performance capabilities, availability status, response time, current resource usage, and fault tolerance. The main objective of static load balancing is to distribute incoming workloads in a consistent manner, without real-time adaptation to fluctuating conditions.

In cloud computing environments, effective load balancing is critical to maximizing system throughput. To achieve this, three fundamental requirements must be satisfied: the ability to efficiently restructure shared information, compatibility with heterogeneous environments, and the maintenance of a stable average system response time [8]. Yao [6] proposed an approach designed to meet these conditions, which involves estimating the resource requirements of incoming requests, recording profitability metrics in a database, and matching those requests to the most suitable server. However, this approach is not without limitations—it struggles with maintaining server connectivity and is only effective in scenarios involving lightly loaded nodes.

To overcome these shortcomings, a refined process is suggested for evaluating cloud resources. This includes the following steps:

1. When the request queue on a server surpasses a predefined threshold, the current request (R1) initiates communication with a randomly selected nearby request (R2) located on a neighboring server.
2. The necessary database updates are transferred from the initial server's queue, and R2 is then assigned to a new server for processing.

Decisions regarding load distribution are generally made using fixed thresholds, upper and lower resource limits, and various predefined evaluation criteria. These mechanisms aim to balance efficiency with reliability while adapting to the diverse and dynamic nature of cloud- based systems.

B. Dynamic Load-Balancing

Unlike static load balancing, dynamic load balancing—also known as feedback-driven load balancing—has the capability to adapt in real-time by scaling the number of active servers and redistributing workloads based on current traffic conditions [5]. This approach enhances flexibility and responsiveness, making it better suited for dynamic cloud environments where workloads can fluctuate unpredictably.

Within the framework of the Open Cloud Computing Federation (OCCF), Zhang [4] introduced a load balancing model that leverages principles from ant colony optimization and complex network theory [7]. This model uses distributed control centers, known as Regional Load Balancer Nodes (RLBNs), to manage and direct traffic intelligently across the network.

The core functions of the proposed system are as follows:

1. When the load on an RLBN exceeds or falls below a specified threshold, the system redistributes the workload between two selected nodes to maintain balance.
2. After redistribution, the routing table—containing information about neighboring nodes and network topology—is updated to reflect the changes.
3. If all neighboring connections (edges) have been evaluated and adjusted accordingly, the system then progresses by visiting another connected RLBN, continuing the load- balancing process.

This decentralized and adaptive approach enables more efficient resource utilization and improves overall system performance in cloud computing environments. By mimicking the collective behavior of ant colonies, the model promotes self-organization and robustness in handling varying workloads.

C. Random Strategy based Load-Balancing

In complex computing environments, simply over provisioning servers to handle all potential workloads is an inefficient approach to load balancing. Instead, effective allocation algorithms must account for inherent uncertainties within the system. To address this, Randles [2] proposes a biased random sampling method, wherein each physical server is modeled as a set of virtual nodes, each maintaining information about its current resource availability.

In this approach, it is assumed that the rate of incoming requests is balanced by the rate of outgoing responses. Based on this assumption, each virtual node is assigned a specific number of incoming edges that represent potential connections from other nodes. These edges help form a directed graph, linking the node responsible for assigning requests (the allocating node) to the one handling execution (the executing node).The final step involves using a random sampling mechanism to select the most suitable server, guided by the availability data of the virtual nodes. This probabilistic method enhances the system's ability to make efficient load distribution decisions under uncertain or dynamic conditions.

D. Centralized Load-Balancing

Platforms for cloud computing that can run several instances of various operating systems are growing more necessary. However, when there are several running processes and a limited number of servers, Central Load Balancing for Virtual Machines (CLBVM) becomes relevant. CLVBM distributes the load among servers in a distributed system in an even manner. Bhandani and Sanjay [3] have suggested the following tactic: Different virtual machines are assigned unique identifiers while the CPU load

is being collected. Heavy (H), moderate (M), and light (L) loads are the categories into which the collected data is separated. The data is then used by a master server to balance the H and L loads initially.

Table 1: Evaluation of different load balancing techniques in comparison

Type	Technique	Strength	Limitations
Static Load Balancing	Artificial Bee Colony Search	Up to a specific number of resources, it improves utilization while maintaining reaction time, and it is simple and adaptable with fewer control parameters. • Boosts the system's overall throughput and effectiveness	• Different algorithms must be employed depending on the volume of requests; efficiency declines as system resources are increased; scalability is required for marketing levels.
	Two Phase Scheduling	• Increase's overall performance and efficiency • Reduce's completion times • Enhance's reso urce utilization	There is no improvement in overall response time or throughput
	Artificial Ant Colony Search	Reduces overhead significantly; achieves heavy performance, resource utilization, and the scalability; is fault tolerant; adapts to heterogeneous situations; and has excellent scalability	There is uncertainty over the maximum execution time and the idle time for each iteration. The response time is also very slow.
Dynamic	Event Driven	Excellent resource consumption due to component analysis; Excellent scalability for commercial use Scaling resources up or down in real-time while maintaining QoS at 99.34%	Has little to no-fault tolerance; doesn't enhance throughput, the performance, or reaction time.

Random	Biased Random Sampling	• Enhanced performance with a large and comparable resources population • Possesses strong marketing resource scalability. • Adaptable and simple, with no trouble being modified to match specific requirements	When resource populations differ from one another, performance ability is lost. When conditions are biassed based on specified conditions, effectiveness is increased. Response time, resource utilisation, or fault tolerance are unaffected.
Centralized	Central LB	It has good response times and resource utilisation, distributes the load equally, and boosts overall performance.	• Has little to no effect on fault tolerance; • Is unsuitable for high - demand applications due to lack of scalability
	Lock Free	It is easier to mana ge many load balancing operations with just one load balancer, which increases overall performance.	• The algorithm's capacity for load balancing is significantly reduced if one of the tiers fails to distribute the requests fairly.
Decentralized	Decentralized Content Aware LB	Reduction of idle time at each nodes, increasing performance by reducing searching time, and has good scalability.	There has been no improvement in fault tolerance or throughput.
Clustering	Active clustering	Has excellent scalability and performs well with a lot of system resources, which are used to boost throughput.	High system resource requirements Efficiency sharply declines as population diversity rises

IV. FUTURE NEEDS

Need 1: Utilize platform heterogeneity as a first need: Knowing complex networks is necessary in order for us to be able to take advantage of the system by understanding the interactions and trade-off's is crucial.

Need 2: Removal of the risk of vendor lock-in through standardisation. There is a need for the load balancing products available today to integrate with each other. This issue can be addressed by LBaaS providers that are compliant with the norms and regulations of OpenStack technology [1].

Need 3: Load Balancing with Energy Efficiency: It is important to come up with load balancing techniques that are energy-efficient so that the performance of cloud computing can be enhanced.

Need 4: Fault tolerance capability: Cloud failover management is important because failure can be caused by many sources because of the richness of cloud networks. Therefore, a load balancer has to be extremely fault tolerant.

Need 5: For a flexible communications API, since modern software services allow integration of numerous enterprises across an extensive variety of back-end systems. Due to this, most communication interfaces, such as HTTP, HTTPS, TCP, UDP, SSL, REST API, and Web Services, should be accessible for LBaaS to be successful.

V. CONCLUSIONS

In conclusion, this article has provided an in-depth examination of load balancing within the context of cloud computing, focusing on its role as a service, its integration into modern business models, and its inherent flexibility in dynamic computing environments. Through the evaluation of various algorithms, methodologies, and strategic approaches, we have highlighted how load balancing contributes to improved performance, scalability, and resource optimization in distributed systems. The study also offered insights into how load balancing is evolving as a service, aligning with the broader shift toward cloud-native and service-oriented architectures.

While significant progress has been made, the technologies that support load balancing in the cloud are still maturing. Issues related to security, reliability, cost-efficiency, and interoperability remain areas of concern that require continued research and innovation. For enterprises to fully embrace cloud computing with confidence, these foundational technologies must become more robust, transparent, and secure. As the field advances, future developments are expected to enhance not only the technical capabilities of load balancing systems but also their strategic value in ensuring seamless, resilient, and efficient cloud services.

REFERENCES

[1] Neutron-LBaaS. Available [Internet]: https:/wiki.openstack.org/wiki/Neutron/LBaaS

[2] M. Randles, D. Lamb, and A. Taleb-Bendiab, “A comparative study into distributed load balancing algorithms for cloud computing,” in Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, Apr. 2010, pp. [pages if available].

[3] A. Bhadani and S. Chaudhary, “Performance evaluation of web servers using central load balancing policy for virtual machines on cloud,” in Proceedings of the Third Annual ACM Bangalore Conference, Article No. 16, New York, NY, USA, 2010.

[4] Z. Zhang and X. Zhang, “A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation,” in Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 2010, pp. 240–243.

[5] A. Khiyaita, M. Zbakh, H. El Bakkali, and D. El Kettani, “Load balancing cloud computing: State of

art," in Proceedings of the 2012 National Days of Network Security and Systems (JNS2), 2012, pp. 106–109.

[6] J. Yao and J. He, "Load balancing strategy of cloud computing based on artificial bee algorithm," in Proceedings of the 2012 8th International Conference on Computing Technology and Information Management (ICCM), Seoul, Korea, Apr. 24–26, 2012, pp. 185–189.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599–616, 2009.

[8] M. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Transactions on Parallel and Distributed Systems, vol. 12, no. 10, pp. 1094–1104, 2001.

[9] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, N. Nitin, and R. Rastogi, "Load balancing of nodes in cloud using ant colony optimization," in Proceedings of the 14th International Conference on Computer Modelling and Simulation (UKSim), 2012, pp. 3–8.

[10] G. Jung, M. A. Hiltunen, K. R. Joshi, R. Schlichting, and C. Pu, "Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS), 2010, pp. 62–73.

[11] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience, vol. 24, no. 13, pp. 1397–1420, 2012.

[12] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Workload analysis and demand prediction of enterprise data center applications," in Proceedings of the 2007 IEEE 10th International Symposium on Workload Characterization, 2007, pp. 171–180.

[13] H. Mehta, M. A. Dave, and A. K. Sharma, "Policy based load balancing in cloud computing," Procedia Computer Science, vol. 45, pp. 778–785, 2015.

[14] M. Rouse, "Round Robin Load Balancing," [Online]. Available:https://www.techtarget.com/searchnetworking/definition/round-robin-DNS

[15] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "CloudAnalyst: A cloudsim-based tool for modeling and analysis of large scale cloud computing environments," in Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, pp. 446–452.