International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 10 August 2023 **Revised:** 28 September 2023 **Accepted:** 20 October 2023*

Cloud Storage Level Service Offering in Virtualized Load Balancer using AWS

[1]Mohanaprakash T A*, [2] S.Preena Jacinth Shalom , [3] Ananthi.S.N, [4] B. Dhanasakkaravarthi, [5] Rajasekaran.A
[*1]Associate Professor , Department of Computer Science and Engineering,
Panimalar Engineering College , Chennai, Tamil Nadu,India
tamohanaprakash@gmail.com
[2] Assistant Professor ,Department of Computer Science and Engineering,
Panimalar Engineering College, Chennai, Tamil Nadu,India
spjshalom@gmail.com
[3]Assistant Professor, Department of Computer Science and Engineering,
S. A. Engineering College, Chennai,
ananthisadhasivam@gmail.com
[4] Assistant Professor, Department of Mechanical Engineering
Agni college of technology .Chennai, Tamil Nadu,India
dhanasakkaravarthi.mec@act.edu.in
[5]Associate Professor, Saveetha School of Engineering,
Department of Artificial Intelligence and Machine Learning ,
Saveetha Institute of Medical and Technical Sciences (SIMATS),Chennai
arajasekaran139@gmail.com

Corresponding Author : tamohanaprakash@gmail.com

Abstract— Distributed computing epitomizes an approach perfectly suited to the realm of IT commitments, leveraging the aggregation of information and resources through electronic cloud service providers utilizing interconnected hardware and software primarily based online, all at a reasonable cost. However, resource sharing can lead to challenges in their accessibility, potentially causing system crashes. To counter this, the technique of distributing network traffic across multiple servers, known as load balancing, plays a pivotal role. This paper ensures that no single server is overwhelmed, thereby preventing overloads and enhancing user responsiveness by equitably distributing tasks. Moreover, it significantly enhances the accessibility of tasks and websites to users. The fundamental objective of this concept is to comprehend load regulation, which operates in tandem with associated frameworks within communication structures like the Web. Load balancing stands as a critical domain within distributed computing, designed to prevent overburdening and to provide equally significant support. Various algorithms are employed to assess the system's complexity. In our proposed strategy, a process is outlined to determine optimal storage space utilization in real-time, utilizing 100 virtual computers, achieving an impressive 92% accuracy rate in its computations. This innovative approach promises efficient resource allocation within the distributed computing framework, thereby optimizing performance and accessibility for end-users.

Keywords- Load Balancing, Storage Level Servcies, Applications, Accuracy, Performacne Indicators

I. Introduction

In cloud computing, load balancing refers to the distribution of client requirements across multiple application servers operating in the cloud background. Cloud load balancing, aided by various types of load balancers, significantly enhances application performance and reliability [1]. The advantages of dispersing resources from typical local sources often involve cost reduction and meeting demand more efficiently [2]. However, traditional load balancing solutions necessitate inputting data into devices and mandate complex IT staff for their setup, modification, and maintenance. Achieving improved outcomes and high reliability typically demands significant IT budgets, especially in larger organizations; cloud-based load balancing is generally unavailable as major cloud service providers typically do not allow external management of their infrastructure [3]. Fortunately, load balancer configuration offers team leaders an opportunity for consistency at a fraction of the cost. Being compatible with fundamental hardware, these solutions are also accessible to smaller businesses. They effectively optimize cloud load and can manage cloud resources like any other application. Distributed computing stands as an endlessly scalable framework within every unit of an organization [4]. All resources within the structure must collaborate to meet user demands. Distributed computing involves disseminating various resources such as storage, servers, connections, software, data, and research over the internet to expedite development, leverage additional resources, and scale website cost savings.

The availability of resources for all users presents the opportunity for web applications to promptly address user demands, a significant advantage for reducing computer usage in various scenarios [5]. Section 2 initiates with an examination of load balancing infrastructure. The proposed load balancing approaches are detailed in Section 3, while Section 4 encompasses the specifics of the outcomes and discussions analyzed using AWS ninja simulation. The article concludes with a concise summary.

II. RELATED WORK

The utilization of a stack balancer in a cloud environment facilitates the segregation and distribution of responsibility among at least two servers. This functionality allows for the optimization of infrastructure to enhance scalability, resource allocation updates, and the attainment of minimal response times [6]. Employing a stack balancer is recommended universally, even if you have only a single server, as it provides advantages in all scenarios [7]. Such systems address various needs, such as ensuring network consistency, managing high traffic loads, and addressing sudden application peaks by balancing the workload across multiple servers or cloud instances [8].

Cloud Load Balancing is the methodology of distributing process assets and responsibilities across multiple servers to ensure maximal performance with minimal latency. By segregating at least two servers, hard drives, or other computer resources, it enables improved resource utilization and reduces response times for network operations [9]. This setup involves the stack balancer identifying and directing network requests among multiple servers in the cloud, as depicted in Figure 1. The process of cloud load balancing involves routing tasks and allocating resources in a distributed computing environment. Organizations leverage load balancing to efficiently manage applications or required responsibilities by distributing resources among various computers, networks, or servers [10].

Cloud computing has significantly advanced across various domains, including task scheduling, virtual machine allocation, infrastructure management, energy optimization, and load balancing. The significance of load balancing within the cloud computing landscape, particularly concerning stakeholders such as Cloud Service Providers and Cloud Service Consumers, has garnered substantial scholarly attention. This attention is partly due to the absence of precise classification among numerous methodologies, which is the focus of this section providing a comprehensive overview of existing literature [11-15].

Ghomi et al. [16-17] presented an outline of load balancing algorithms in distributed computing. They categorized a range of load balancing and scheduling algorithms into seven distinct classes, including Hadoop-map-reduce load balancing, agent-based load balancing, commonality-based load balancing, application-oriented load balancing, general load balancing, network-aware load balancing, and workflow-specific load balancing. The algorithms are systematically organized by type, with their respective advantages and disadvantages reviewed. Meanwhile, Milani et al. [18] analyzed the contemporary landscape of load balancing based on their study's findings. They classified various load balancing algorithms into static, dynamic, and hybrid categories, addressing crucial issues related to load balancing's value, necessary metrics, potential, and challenges faced.

To gather the most pertinent information from diverse publishing sources, a comprehensive search using Boolean operations in search strings and the Quality Evaluation Agenda (QAC) was conducted. However, both studies only considered a limited subset of Quality of Service (QoS) metrics such as response time, makespan, scalability, resource utilization, migration time, throughput, and energy savings [19-20]. This limited scope omitted other critical QoS metrics like migration cost, service level violations, equilibrium level, task rejection ratio, and more. To address this gap, this review aims to systematically select metrics for a more thorough analysis [21].

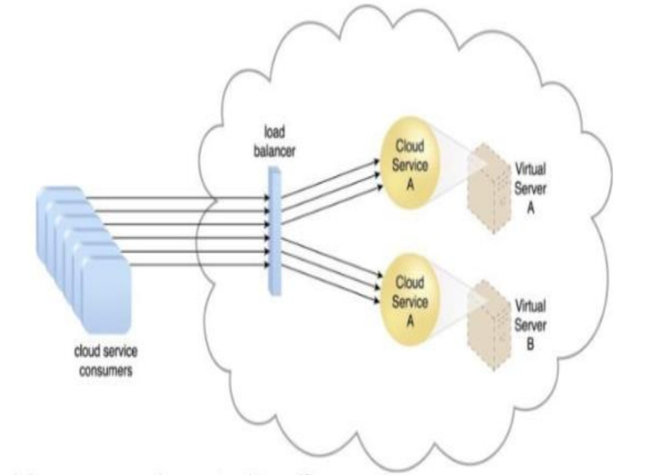

Figure 1: Service offering based on Loads

III. METHODOLOGY

Load change is the progression of the association's traffic through the assortment of servers. The stack balancer is the server that does this. Load troublesome practices because of group execution and programming. More reasonable and simpler to utilize. Associations can convey their client applications quicker and with special execution through the cloud. The enhancement assists keep the site with dealing light and responsive. Can solidify the extended client traffic with useful transfers from balancers and seize them on different servers. or on the other hand arrangement gadgets. This is critical for online areas that oversee huge quantities of Web guests each second. These useful burden balancers are significant for moving work to arrangements or other specific organizations. Flexibility to traffic tops - During any assertion of results, a working college site could bomb totally because of the chance of serious areas of strength for getting. Cloud Burden Balancers. Decrease reaction time to client needs Essentially work on the pace of resource use Further develop execution Keep a strong system Increment the versatility of the structure to answer changes Stack Balancer Whether a

Article Received: *10 August 2023* ***Revised:*** *28 September 2023* ***Accepted:*** *20 October 2023*

center point falls flat, obligation can move to a more unique one The center point can be migrated when it is disseminated to various servers or drives in the association [23].

Load balancing is driven by various fundamental objectives that focus on optimizing system performance and responsiveness. These include managing sudden traffic surges on servers, promptly reducing response times for user requests, improving resource utilization efficiency, and significantly enhancing overall system performance. Ensuring system stability and flexibility to adeptly handle changes are also key goals. Additionally, load balancing aims to minimize waiting times in queues and streamline workflow processes for a more efficient operational structure.

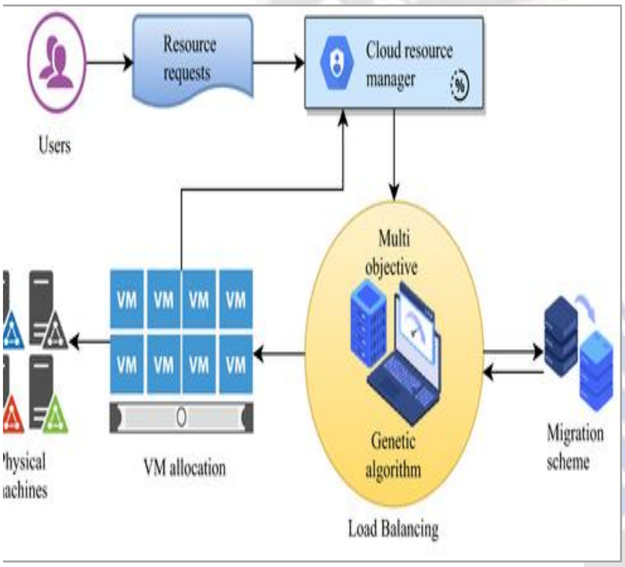

Figure 2: Proposed system - Load Balance Services

A. Load Balancer

Load change is likewise an essential piece of sky climbing. The cloud establishment necessities to extend easily to deal with both upstream and downstream traffic. At the point when a cloud "scales" it normally runs various virtual servers and runs a wide assortment of uses. The part that conveys the traffic between these new appearances is the battery balancer. Without a heap balancer, the virtual servers being checked will most likely be unable to identify approaching traffic in an arranged way etc. expect that they are not envisioning it. It can likewise recognize blocked off servers and traffic in the works for the people who are still out of luck. Contingent upon the stack tuning estimations, load balancers might evaluate whether a given server (or gathering of servers) is probably going to be dominated quicker and course traffic to different centers than these proactive capacities the probability of your cloud organizations. Performing load tuning that can emphatically diminish can likewise be essential to accomplish green circulated figuring. The clarifications behind this are: Restricted power utilization - Burden tuning can diminish power utilization by restricting pointless risk to fundamental centers or virtual machines [23].

TABLE I: MAJOR CONSIDERATIONS OF LOAD BALANCER

Defined Services	**Load Balancer**	**Virutal Machine**	**Service Networking**
This permits you to sort out various duty adjusts. It likewise assists the organization with working as a virtual machine and a stockpiling variant.	It gives systems administration and content conveyance administrations utilizing a scope of burden adjusting calculations.	Oversee Actual Chargers in the virtual machine	Load Adjusting as a Help (LBaaS) utilizes load adjusting innovation to meet the quick traffic and application needs of associations conveying private cloud foundation.
Performance Indicator	**Internet Traffic**	**Burst Time Measurement**	**Flexibile Usecase**
Load adjusting strategies are more modest and simpler to carry out than their partners.	Associations can deal with their clients' applications quicker and convey better execution for moderately minimal price.	Cloud adjusting gives versatility to deal with traffic. With effective burden adjusting, you can undoubtedly deal with client traffic at an undeniable level with the presence of servers and organization peripherals.	It shows versatility, adaptability and the capacity to deal with traffic.

B. Operational Level Service as Load Balancer

a. Static Burden Adjusting:

Concerning stack changing, a load changing methodology is "static" expecting it ignores the situation of the system. The situation of the system incorporates sports very much like the level of stack of the unmistakable processors (and in a couple of occasions in like manner the floods). All matters considered, structure broad speculations are made in advance of time, for instance, look occurrences and approaching resource essentials. The numbers, energy and correspondence speeds of the computer processors are besides known [18-20]. Static weight changing is consequently intentional to pursue a given relationship of occupations with the close by processors so one the display compositions is restricted. Static weight balancers are typically fixated on a switch, or master, which appropriates and progresses the stores. This lower might recall measurements roughly the tasks to be dispersed and a customary execution time. The upside of static estimations is that in actuality relentless endeavors aren't hard to establishment and amazingly capable [18-21].

b. Dynamic Burden Adjusting:

Dynamic estimations consider the ongoing stack of every IT unit (otherwise called centers) rather than static burden tuning techniques. Associations could then move from an over-burden center point to a stacked center point for better dealing with. are more challenging to construct generally speaking, they can deliver astounding outcomes, particularly when execution times shift broadly from one message to another. Since you needn't bother with an alternate place for task adjusting, the strong burden adjusting design can make it more versatile [22]. It's an interesting position when organizations have a processor be distributed by their particular status. The powerful errand, then again, implies the capacity to revamp tasks on a constant reason as shown by the structure, and clearly includes a load fit estimation requires over the top correspondence to come to its end results, the gamble. to digress from the general point of the issue [23].

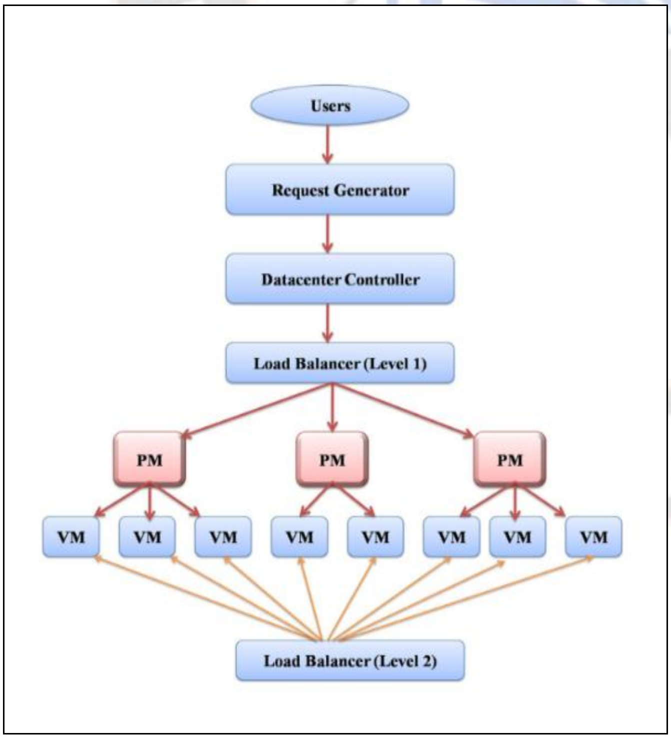

Figure 3: VM Operations in each level

When dealing with a cloud load balancer, the approach to inputting metrics can vary depending on the specific cloud service provider or type of load balancer in use. Typically, metrics in cloud load balancing encompass various performance, usage, and health indicators. These metrics can encompass data on incoming and outgoing network traffic, latency in request processing, error rates, throughput, and the status of backend servers or instances, which might include metrics like CPU utilization, memory usage, and disk I/O. The syntax for collecting and inputting these metrics can differ depending on the particular cloud provider's API or monitoring tools. For example, most cloud service providers offer APIs for retrieving metrics. Amazon Web Services (AWS) uses the CloudWatch API, where the syntax involves constructing API calls with specific parameters such as metric names, time ranges, and dimensions. Additionally, some cloud services provide monitoring dashboards or tools that allow users to configure and visualize metrics by setting up configurations specifying the metrics, time intervals, and thresholds. AWS CloudWatch syntax for the Command Line Interface (CLI), which would be used to retrieve specific metrics in AWS environments.

IV. SIMULATION AND RESULT

The proposed structure for impromptu burden adjusting in the Distributed computing Climate is laid forward here. The proposed structure's essential goal is to establish a high accessibility cloud climate that forestalls framework disappointments and works with the recuperation of client exercises, thus helping the wellbeing of Distributed computing applications. In this part, we will examine the results of our exploration on the general benefits of a few cloud-based load-adjusting techniques. Proactively based strategies obviously give more prominent idea to task planning and asset booking with 2, 5, 10, 20 burden balancers, yet give less thought to VM planning, which represents 92%. Table 3 shows that Pseudo-code for recreation setting load balancer boundaries.

A. AWS Simulations

The proposed structure for impromptu burden adjusting in the Distributed computing Climate is laid forward here. The proposed structure's essential goal is to establish a high accessibility cloud climate that forestalls framework disappointments and works with the recuperation of client exercises, thus helping the wellbeing of Distributed computing applications. In this part, we will examine the results of our exploration on the general benefits of a few cloud-based load-adjusting techniques. Proactively based strategies obviously give more prominent idea to task planning and asset booking

***Article Received:** 10 August 2023 **Revised:** 28 September 2023 **Accepted:** 20 October 2023*

with 2, 5, 10, 20 burden balancers, yet give less thought to VM planning, which represents 92%. Table 3 shows that Pseudo-code for recreation setting load balancer boundaries.

B. Pseudo code for Reenactment setting load balancer boundaries

This is the default load adjusting technique in AWS Code Ninja and has no mandate values like weight, max_fails, fail_timeout, down, reinforcement, slow_start, max_conns

```
upstream application {
   server s1.application.com slow_start=30s;
   server s2.application.com weight=3;
   server s3.application.com max_fails=5;
   server s4.application.com fail_timeout=10s;
   server s5.application.com reinforcement;
}
upstream application {
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream application {
   least_conn;
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream application {
   ip_hash;
   server s1.application.com;
   server s2.application.com weight=3;
}
upstream application {
   hash request_uri;
   server s1.application.com;
   server s2.application.com weight=3;
}
```

Weight=3 implies s2 will be chosen 3 fold the amount of as different servers. The default weight is 1 slow_start=30s indicates the time (30seconds) a server which was down is given to recuperate its tasks prior to being overburdened with associations, which might make it go down once more. max_fails=5 implies there ought to be 5 break associations inside the period determined by fail_timeout mandate, before the heap balancer denotes a server as broken. Development of virtual machines: The thought is to make a machine as a report or record. How should the central piece of the data be used be used by staying aware of its delay. A strong weight balancer ought to consider rapidly changing necessities for figure, memory, contraption plan, to say the least. The AWS Test system is a completely overseen administration that permits us to do blame infusion probes AWS, which thus works on the speed, discernibleness, and strength of our applications. The most common way of placing an application under coercion in testing and creation settings by setting off startling occasions like a spike in computer chip or memory use, seeing how the framework answers, and making changes in view of what is realized. Figure 5 shows that reproduction consequence of aws ninja portrayal

TABLE II:TEST BED RESULT AND ACCURACY VALUES

Number of VMs	Load Balancer	Weight	Hop Count	Accuracy (%)
10	10,20,50	100,500,1000	100,500,1000	98,96,95
20	10,20,50	100,500,1000	100,500,1000	96,95,95
50	10,20,50	100,500,1000	100,500,1000	95,96,96
100	10,20,50	100,500,1000	100,500,1000	95,96,95
500	10,20,50	100,500,1000	100,500,1000	95,96,95

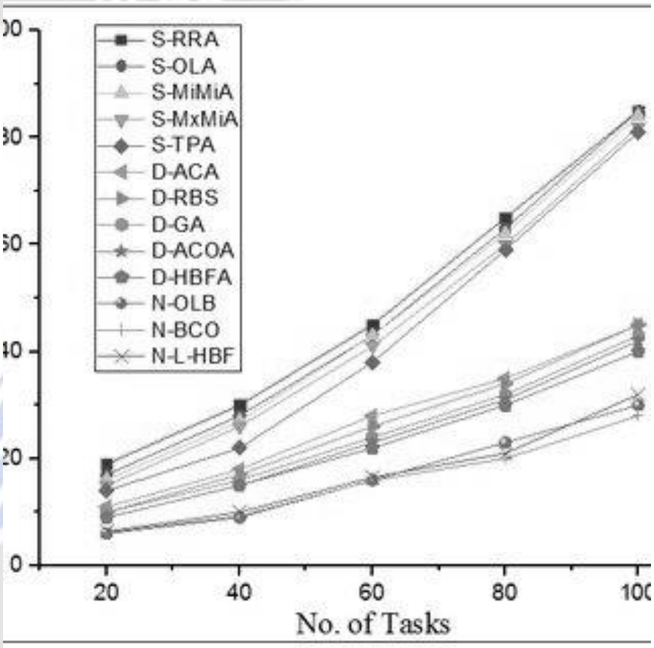

Figure 5(a)

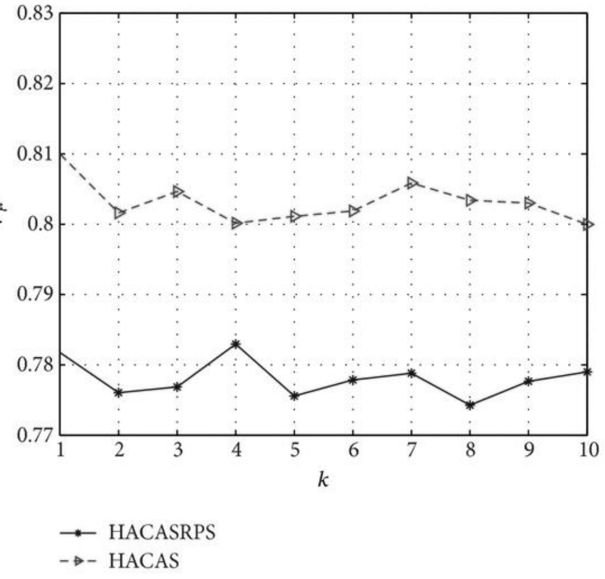

Figure 5(b)

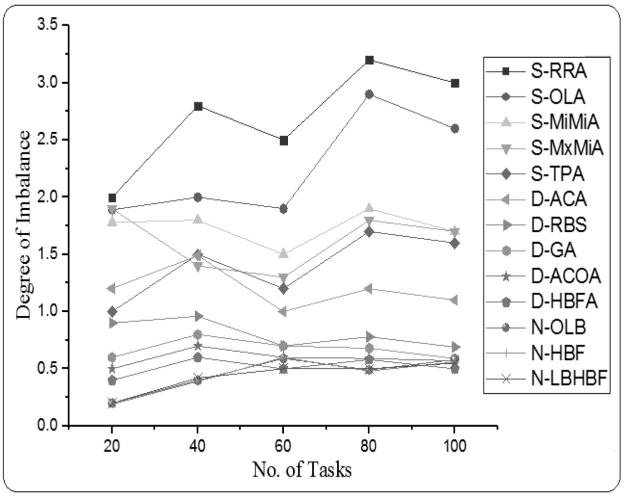

Figure 5(c)

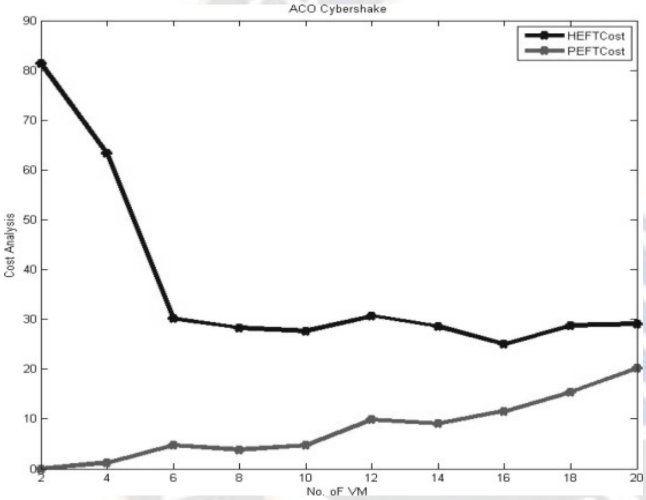

Figure 5(d)

Figure 5(a-d):AWS Simulation results

V. Conclusion

This paper examines various load balancing strategies presented in the literature. The investigation focuses on cloud-related load imbalances and their contributing factors. The activities linked with load balancing and an offline model for load balancing were briefly explored. Cloud computing allows multiple users to access diverse resources online based on their needs. However, significant barriers exist in distributed computing. Load regulation emerges as a critical challenge in this field. This dossier delves into several static and dynamic measurements. The heterogeneous nature of the cloud is well-known. Static measurements simplify the presentation and monitoring of the environment but fail to replicate the diverse nature of cloud environments. Dynamic load adjustment measurements are complex to illustrate but better suit the varied nature of cloudy situations. Cloud computing brings about cost reductions, enhancing system performance, infrastructure expansion, service time, and management. In proposing more effective future load balancing methods, this paper scrutinizes the challenges of current load balancing algorithms. Most analyzed publications minimally address or neglect the most crucial Quality of Service (QoS) indicators such as migration time, traffic cost, power usage, service level breaches, task rejection rates, and level of equilibrium. It's observed that the complexity of the method is not heavily considered while evaluating load balancing effectiveness as the research progresses.

AUTHORS' CONTRIBUTION

Author 1 and 2 implemented the concept and drafted the article with assistance of authors 3,and 4, respectively. Author 5 reviewed the article.

CONFLICT OF INTEREST

The authors declare that have no competing interest.References

References

[1] Khan Z, Singh R, Alam J, Saxena S (2020) Classification of Load Balancing Conditions for parallel and distributed systems. International Journal of Computer Science Issue 8: 411- 419.

[2] Angurala, M., Bala, M., & Bamber, S. S. (2021). A novel technique for energy replenishment and load balancing in wireless sensor networks. Optik, 248, 168136.

[3] Jarraya, M., & Elloumi, S. (2022). Load balancing scheduling algorithms for virtual computing laboratories in a Desktop-As-A-Service Cloud Computing Services. Computer Communications, 192, 343-354.

[4] 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

[5] Gulati A, Chopra RK (2019) Dynamic Round Robin for Load Balancing in a Cloud Computing, International Journal of Computer Science and Mobile Computing: 274-278.

[6] Shahbaz Afzal and G. Kavitha, "Load balancing in cloud computing – A hierarchical taxonomical classification", Journal of Cloud Computing volume 8, Article number: 22 (2019). DOI: https://doi.org/10.1186/s13677-019-0146-7

[7] Bhawesh kumawat , Rekha kumawat,"A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment using Cloud Analyst", IJESC Volume 7 Issue No.3

[8] Ms. Shalini Joshi , Dr. Uma Kumari "Load Balancing in Cloud Computing:Challenges & Issues" , Conference: 2016 2nd International Conference on Contemporary Computing and Informatics, DOI:10.1109/IC3I.2016.7917945.

[9] N.Kato et al., "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," IEEEWireless Commun., vol. 24, no. 3, pp. 146–153, Jun. 2017

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

[11] Javadpour, A., Sangaiah, A. K., Pinto, P., Ja'fari, F., Zhang, W., Abadi, A. M. H., & Ahmadi, H. (2023). An energy-optimized embedded load balancing using DVFS computing in cloud data centers. Computer Communications, 197, 255-266.

[12] Abdi, M., Ginzburg, S., Lin, X. C., Faleiro, J., Chaudhry, G. I., Goiri, I., ... & Fonseca, R. (2023, May). Palette load balancing: Locality hints for serverless functions. In Proceedings of the Eighteenth European Conference on Computer Systems (pp. 365-380).

[13] Shahid, M. A., Alam, M. M., & Su'ud, M. M. (2023). Performance Evaluation of Load-Balancing Algorithms with

Different Service Broker Policies for Cloud Computing. Applied Sciences, 13(3), 1586.

[14] Latchoumi, T. P., & Parthiban, L. (2022). Quasi oppositional dragonfly algorithm for load balancing in cloud computing environment. Wireless Personal Communications, 122(3), 2639-2656.

[15] Jena, U. K., Das, P. K., & Kabat, M. R. (2022). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences, 34(6), 2332-2342.

[16] Yu, D., Ma, Z., & Wang, R. (2022). Efficient smart grid load balancing via fog and cloud computing. Mathematical Problems in Engineering, 2022, 1-11.

[17] Shakambhari, Raj, J. S., & Anantha Babu, S. (2022). Smart Cyberbullying detection with Machine Learning. In Disruptive Technologies for Big Data and Cloud Applications: Proceedings of ICBDCC 2021 (pp. 237-248). Singapore: Springer Nature Singapore.

[18] Babu, S. A., Basha, M. J., Arvind, K. S., & Sivakumar, N. (2023, June). Analysis of Hate Tweets Using CBOW-based Optimization Word Embedding Methods Using Deep Neural Networks. In Proceedings of International Conference on Computational Intelligence and Data Engineering: ICCIDE 2022 (Vol. 163, p. 373). Springer Nature.

[19] Babu, S. A., James, J. W., & Vedaiyan, R. (2021, October). ARIMA based Time Series Analysis: Forecast COVID-19 Most Vaccinated Process and Active Cases classify using Probability Distribution Curve Rates (ARIMAPDC). In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 546-551). IEEE. S.Manikandan,

[20] M.Chinnadurai, D.Maria Manuel Vianny and D.Sivabalaselvamani, "Real Time Traffic Flow Prediction and Intelligent Traffic Control from Remote Location for Large-Scale Heterogeneous Networking using TensorFlow", International Journal of Future Generation Communication and Networking, ISSN: 2233-7857, Vol.13, No.1, (2020), pp.1006-1012.

[21] Manikandan, S, Chinnadurai, M, "Effective Energy Adaptive and Consumption in Wireless Sensor Network Using Distributed Source Coding and Sampling Techniques",. Wireless Personal Communication (2021), 118, 1393–1404 (2021).

[22] Hongsuk Yi, HeeJin Jung and Sanghoon Bae, “Deep Neural Networks for Traffic Flow Prediciton”, IEEE Conference on BigComp 2017, 971-5090-3015-6/17, pp.328-331

[23] Martın Abadi, et al., TensorFlow:Large-Scale Machine Learning on Heterogeneous Distributed Systems, arXiv:1603.04467v2, 2015.