Two-Tier Load Balancer as a Solution to a Huge Number of Servers

Fawaz Alharbi [1], Mustafa ElGili Mustafa [2]

1.Computer Science Department , Huraymila College of Science and Humanities, Imam Mohammad Ibn Saud Islamic University

2. Computer Science Department, Huraymila College of Science and Humanities , Imam Mohammad Ibn Saud Islamic University

Abstract

High number of users, connected devices and services on the Internet producing high traffic and load on the web servers causes a degradation of the quality of Internet services. A possible solution to this problem is to use a cluster of web servers. The cluster requires a load balancer to provide scalability and high performance of the services offered. The main load balancer is the only entry point to the server cluster in this architecture. In this paper, the researchers propose a two-tier load balancer rather than a single one to achieve more scalability and reduce the load on the main load balancer. The study also compared three features, Response Time Average, the load balancer CPU Utilization, and Servers' CPU Utilization. The comparison uses three algorithms (Round Robin, Number of Connections, and Least Load) through two experiments. The results concluded that the Multi-Tier Load Balancing method offered better CPU utilization than a Single-Tier Load Balancing method for Round Robin and Server Load algorithms. However, the Single-Tier Load Balancing method provided better response time for all three algorithms. Moreover, the Round Robin and Server Load algorithms using a Multi-Tier method balanced the CPU utilization for all servers. This result shows the Multi-Tier method handles huge traffic and large number of servers with better CPU utilization.

Keywords:

Round Robin; Least connection; server load, Load balancer; Multi-tier load balancer; Server Cluster.

1. Introduction

The internet is experiencing rapid growth in its users due to the use of internet services in most aspects of people's lives. According to the United Nations, more than three billion people use the internet every day [1]. Social networking, multimedia streaming, and file exchange cause increased network traffic. Data centers with higher efficiency and many servers are required to provide services to the vast number of internet users. Some organizations try to improve their data center capabilities by utilizing third-party IT resources, such as colocation data centers and public cloud[2]. However, most workloads are still in data centers that are enterprise-owned on-premises facilities [2]. An Uptime Institute survey showed that the majority (58% today and 54% projected in two years) reported that most of their workloads run in organizations' data centers, that is, enterprise-owned and on-premises facilities [2], as shown in Fig 1.

Traditionally, a machine with a single server handles incoming requests but fails if de-

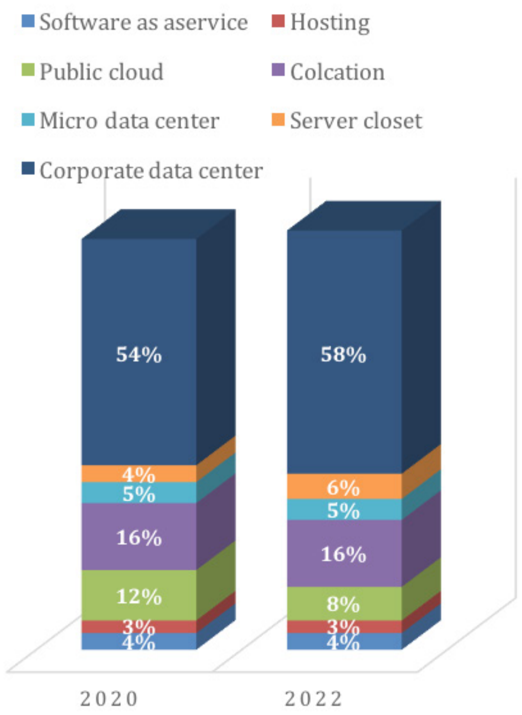
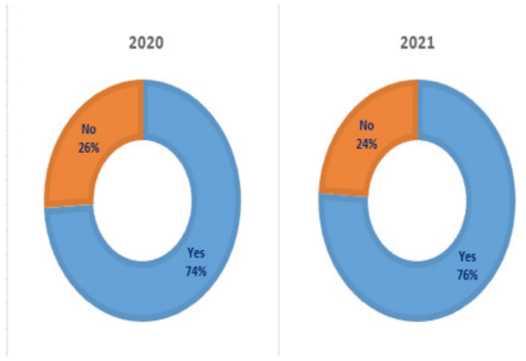


Fig. 1. Comparison of the Distribution of Workload Based on Data Centers type in 2020 and 2022 [2]

mand on several web servers increases [3]. Uptime surveys indicated that about 60% of the respondents noticed some outages in their IT services for several reasons in the past three years, as explained in Fig 2 [4]. Analyzing public outages for 2020 shows that software/IT and Network accounted for almost three out of four outages as shown in Fig. 3 [4].



Has your organization had any IT service outages over the past three years?

Fig. 2. Types of Outages in Data Centers [4]

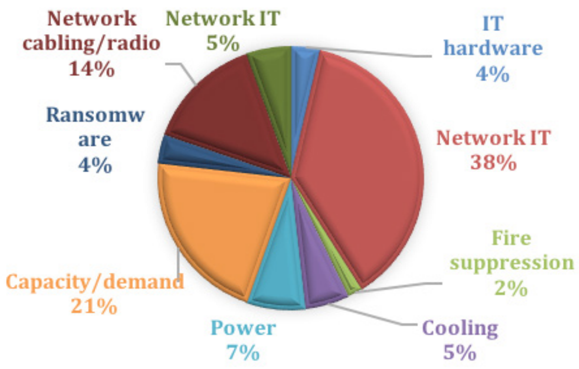


Fig. 3.
Causes of Publicly Reported Outages in 2020 [4]

One way to deal with this issue is to replace the server with a faster one or increase hardware capacities. However, this solution could lead to a scaling problem[5]. Distributing the requests among a cluster of web servers could be a solution to deal with such a situation. Many web servers require to be load balanced so that no server is overloaded or underloaded. The requests then will have a better response time and less processing time. A load-balancing architecture allows requests to be distributed among clustered web servers[5]. In this paper, the researchers proposed a two-tier load One way to deal with this issue is to replace the server with a faster one or increase hardware capacities. However, this solution could lead to a scaling problem[5]. Distributing the requests among a cluster of web servers could be a solution to deal with such a situation. Many web servers require to be load balanced so that no server is overloaded or underloaded. The requests then will have a better response time and less processing time. A load-balancing architecture allows requests to be distributed among clustered web servers[5]. In this paper, the researchers proposed a two-tier load balancer, which can be used when dealing with a vast amount of traffic. The main contribution of this research is the critical analysis for implementing a two-tier load balancing architecture through comparing three features, Response Time Average, the load balancer CPU Utilization, and Servers' CPU Utilization. The comparison uses three algorithms (Round Robin, Number of Connections, and Least Load) through two experiments to evaluate the performance using various algorithms. The rest of this paper is organized as follows. Section 2 offers background information about the Architecture of the Web Server Cluster. Section 3 describes Load Balancing architectures and algorithms. Section 4 reviews current works related to the implementation of tow-tier load balancing architecture. Section 5 explains the research problem followed by Section 6 and 7 which describes the proposed solution and the simulation methodology. In Section 8, the results are discussed. Then, section 9 provides the conclusion of our study.

2. The Architecture of the Web Server Cluster

A single web server is not sufficient to support a high-traffic web application. Thus, using clustered web servers increases the reliability and availability of the web server [6]. A cluster-based web system is composed of N server machines, as shown in Fig. 4, which are connected through a high-speed network to carry out user requests.

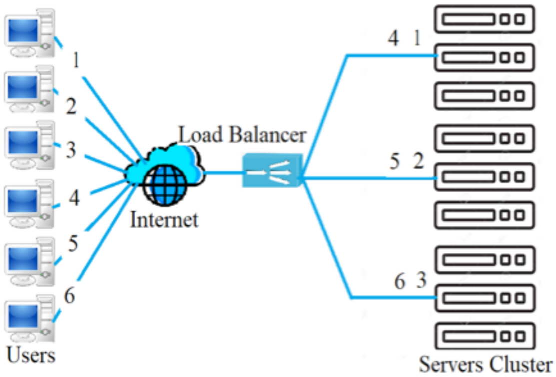


Fig. 4. Architecture of a Web Server Cluster

Each server machine acts as a node with its own computing resources [7]. Four basic features need to be addressed: [8]

- Load balancing: Since a server cluster contains several nodes, its computing task depends on the composition of each node. The system performance optimization can be achieved only when every node sustains balancing.
- Single System Image: The server cluster must make all nodes transparent to users by providing an abstract user interface as a single system.
- Scalability: Due to the high usage of multimedia applications on the server and heavy expenditures, users might divide the investment into several parts. The server also should always provide high scalability, e.g., to add new resources or to satisfy the possible increased demands.
- Availability: Clusters are susceptible to partial failures, and the probability of partial failure increases with the size of system resources. Partial failures must be addressed in any implementation

of clusters to provide a higher level of availability and reliability.

3. Overview of Load Balancing

Load balancing is a technique of parallel loading between two or more computers, network links, and a Central Processing Unit (CPU) that maximizes throughput, minimizes response time, and avoids overload [9] as shown in Fig. 4. Load balancing divides the load among several nodes to enhance the utilization of the computation power of every node and decrease the average task response time. This process maximizes the system throughput [8]. Load balancing techniques are crucial to support cloud computing implementation since they improve the efficiency of cloud computing services' performance [10].

3.1. Load Balancing Measurement Parameters

Measurement parameters are used to assess the load balancing methods that check whether a given method is good enough to effectively balance the load [16]. These parameters are as described in [16]:

- Throughput: It is the rate of work to be completed in a given amount of time.
- Response time: It is the amount of time between the start of a user request and the completion of that request.
- Fault tolerance: It refers to the feature of the algorithm that lets a system work even during a failure condition of the system.
- Scalability: It is the ability of the algorithm to scale itself based on specific conditions.
- Performance: It is the overall quality of the algorithm regarding the accuracy, cost, and speed.
- Resource utilization: It is used to check the performance of various resources.

3.2. Load-Balancing Algorithms

Load balancing has been a crucial issue for server clusters. Researchers continually proposed better strategies and solutions to improve the processing power of servers [19]. Depending on the implementation method, there are two types of load balancing algorithms [5]:

- Static Load Balancing Algorithms: The current state of a system does not play any role in the load balancer's decision in this group of algorithms. The load balancer decides the allocation and execution of specific services on a group of virtual machines.
- Dynamic Load Balancing Algorithms: The algorithm relies on the current state of the server. The load balancer evaluates the current load of each available virtual machine then assigns service to a suitable and proper virtual machine. The load balancing algorithm must fulfill properties like the maximum number of context switches, throughput, CPU utilization, minimum turnaround time, response time, and waiting time[10].
- The benefits of having efficient load balancing include: [11]
- Uniform distribution of load on nodes.
- Improved overall performance of the system
- Higher user satisfaction
- Faster response

- System stability
- Reduced carbon emission

The following subsections discuss some popular load balancing algorithms:

3.2.1. Round-Robin (RR) Algorithm

The round-robin scheduling algorithm shown in Fig. 5 sends each incoming request to the next node in its list. So, for a three-server cluster (servers A, B, and C), request 1 would be assigned to server A, request 2 would be sent to server B, request 3 would go to server C, and request 4 would be assigned to server A, until completing the cycling or "round-robin" of servers [17]. It treats all the real servers equally without considering the number of incoming connections or the response time of each server [17].

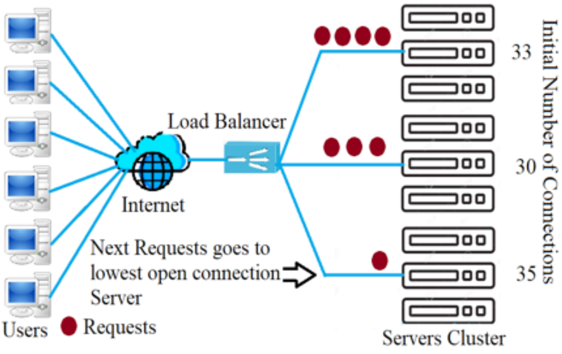

Load Balancer
Internet
Next Requests goes to lowest open connection Server
Users
● Requests
Servers Cluster
33
30
35
Initial Number of Connections

Fig. 5. Round Robin Algorithm

3.2.2. Least-Connection Algorithm

The idea of the least-connection algorithm applies the dynamic scheduling concept by counting live connections for each server dynamically. Least-connection scheduling is an appropriate solution for the virtual server that manages a collection of servers with similar performance since it provides smooth distribution when a load of requests is high [17].

The least-connection scheduling algorithm manages network connections to the server based on the minimum number of established connections [8] as shown in Fig. 6.

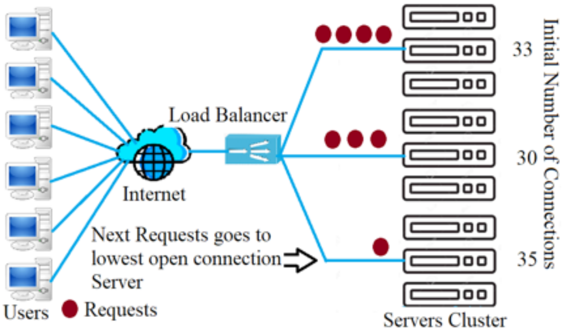

Load Balancer
Internet
Next Requests goes to lowest open connection Server
Users
● Requests
Servers Cluster
33
30
35
Initial Number of Connections

Fig. 6. Least-Connection Algorithm

A least-connection scheduling algorithm distributes the workloads to real servers with the fewest active connections. This algorithm is suitable for network traffic with a high degree of variation in the request load since it applies the dynamic scheduling algorithm. The least-connection algorithm assumes that all servers are identical, so the newly arrived request always goes to the server with the fewest connections. However, one of the disadvantages of this algorithm is the system performance being not ideal when there are various processing capabilities of the servers [18].

3.2.3. Least Loaded Algorithm

In this scheme, the least loaded algorithms assign a new request to the server with the lowest workload, as shown in Fig. 7. The baseline algorithm must identify the service time of the client requests, which is often unknown as the requests arrive. Thus, it is difficult to use the baseline algorithm in practice. However, the baseline algorithm can create an upper bound on the performance. Hence, we refer to it as the baseline algorithm and use it for comparison with the performance of the other schemes [19].

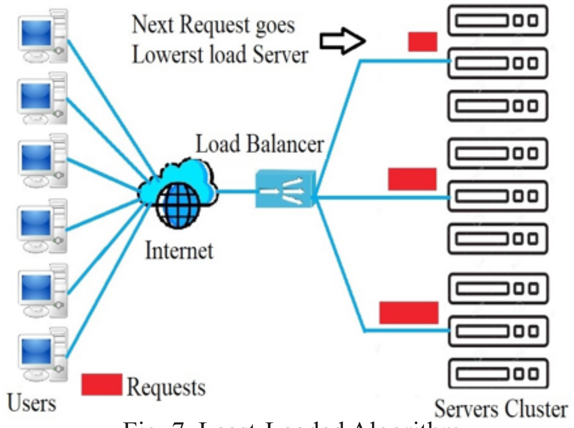


Fig. 7. Least-Loaded Algorithm

4. Related work

There have been various attempts to solve the issue of having a single load balancer in a high traffic computing environment to avoid its drawbacks. [12] developed a low-cost two-tier load balancing model for high availability applications and compared it with the three-tier architecture. They found that the two-tier load balancing model performed at a similar level when comparing with the three-tier architecture in terms of network failover and storage limitation. They only utilized Round-Robin scheduling without mentioning other algorithms. [13] proposed a two-tier load balancing architecture for Wi-Fi networks. Their solution increased the load balancing performance by 34-40%. However, the study only focused on Software Defined Networking (SDN). [14] examined the implementation of a two-tier load balancing architecture in the Internet of Things (IoT) domain. The study results indicated that the two-tier model improved the efficiency of the network when compared with single-tier architecture. [15] developed two-tier architecture to support Mobile Edge Computing and it showed better autoscaling results than single-tier architecture.

5. Problem Statement

Having a single load balancer makes all the load balancing effort goes through a single device as shown in Fig. 8. The probability of failure could increase if there is high traffic. In addition, it will be difficult for the load balancer to distribute the traffic if there is a huge number of servers. Moreover, the administrators could not add more servers to expand the server cluster.

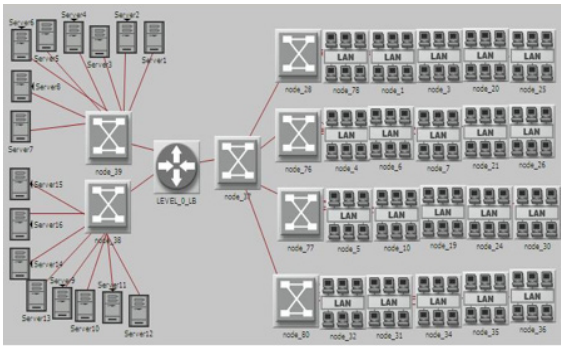


Fig. 8. Single-Tier Load Balancer

6. Proposed Solution

To solve this problem, we added another tier with two load balancers, as shown in Fig. 9. The expected benefits of adding this tier include reducing the load on the primary load balancer and providing high scalability to the server cluster. We added this tier after the primary load balancer to maintain it as the only entry point to the server cluster.

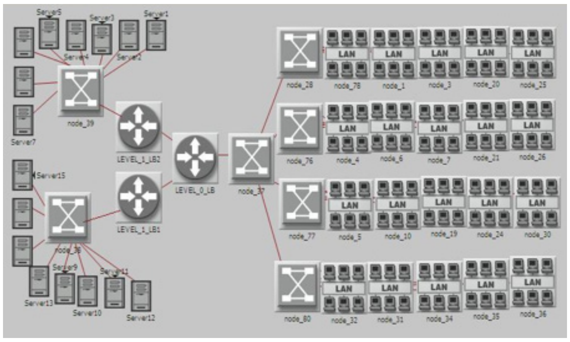


Fig. 9. Multi-Tier Load Balancer

7. Simulation Methodology

The network simulation uses OPNET® Modeler, a simulation software tool with multiple capabilities. It allows the simulation of many types of networks with various protocols [20]. In the first experiment, a network with 16 HTTP servers provides HTTP service to 1800 clients, and there is one load balancer device between the servers and clients, as shown in Fig. 9. The second experiment was conducted with the same number of clients and servers as the first experiment with the addition of another tier containing two load balancers, as shown in Fig. 10. Two experiments were tested with the main three loading balancer algorithms (Round-Robin, Least-Connection, and Least Loaded). Both experiments evaluated these three features:

- Response time to reflect the time required to complete the user request.
- Main load balancer CPU utilization to measure the performance of the main load balancer.
- Servers' CPU utilization to review the server's performance.

8. Results

The results show that the Single-Tier Load Balancing method provided better response time for all three algorithms (see Fig. 10, Fig. 11, and Fig. 12). However, the difference in response time for the Single-Tier Load Balancing method and the Multi-Tier Load Balancing method for all algorithms is low (i.e., less than 0.0001 second). This finding is expected since adding an extra layer will lead to more time for processing activities.

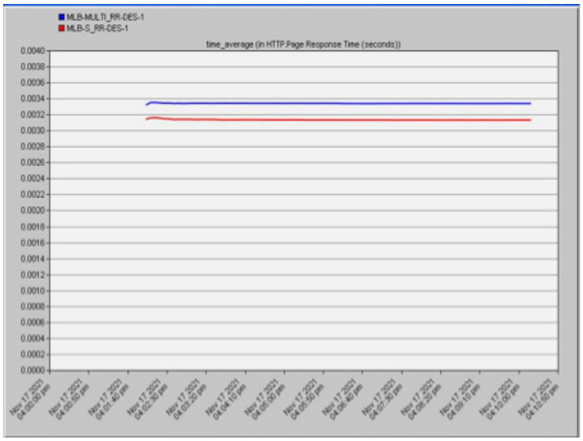


Fig. 10. Round Robin Response Time Average

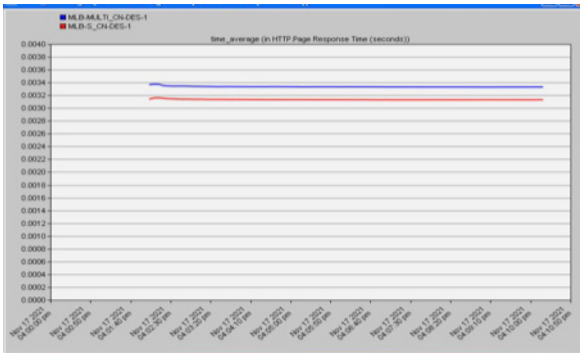


Fig. 11. Number of Connection Response Time Average

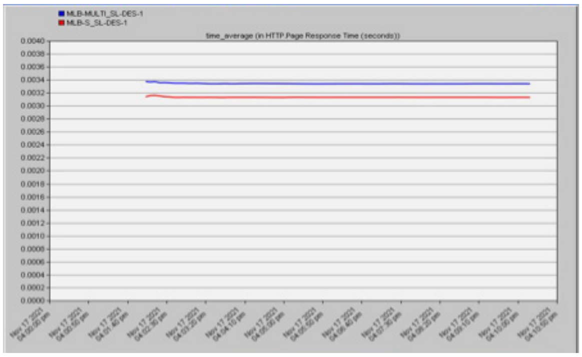


Fig. 12. Server Load Response Time Average

Regarding CPU utilization of the main load balancer, the results indicate that Multi-Tier Load Balancing method offered better CPU utilization than the Single-Tier Load Balancing method for the Round Robin and Server Load algorithms (see Fig. 13 and Fig. 15). For the Number of Connection algorithm, both methods showed similar performance (see Fig. 14). The reason for such results is that both

Round Robin and Server Load algorithms depend on the number of nodes connected to the main load balancer. In the experiments, the researchers investigated the servers' CPU Utilization for the three algorithms and each method. The results show that the Single Tier method using the Number of Connection Algorithm and Server Load Algorithm had balanced CPU utilization for all servers (see Fig. 17 and Fig. 18). However, this is not the case for the Round Robin Algorithm since the CPU utilization is high for one server (See Fig. 16). For the Multi-Tier method, using the Round Robin Algorithm and Server Load Algorithm balanced CPU utilization for all servers (see Fig. 19 and Fig. 21). However, this is not the case for the Number of Connection Algorithm since the CPU utilization is high for two servers (see Fig. 20). The findings indicated that the Multi-Tier balancing method could improve the performance of server cluster architecture. 9.

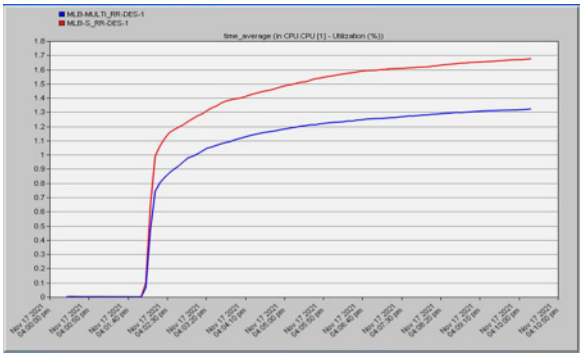


Fig. 13. Round Robin CPU Utilization

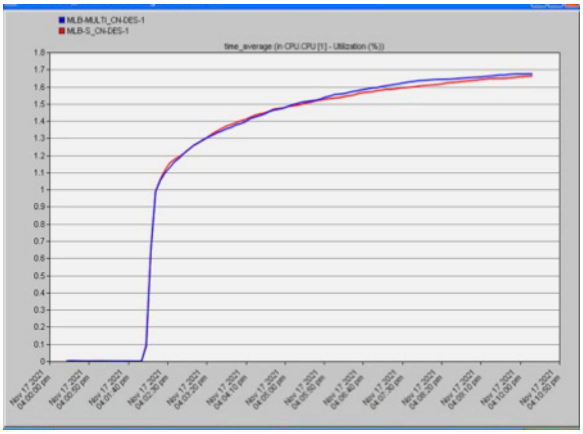


Fig. 14. Number of Connection CPU Utilization

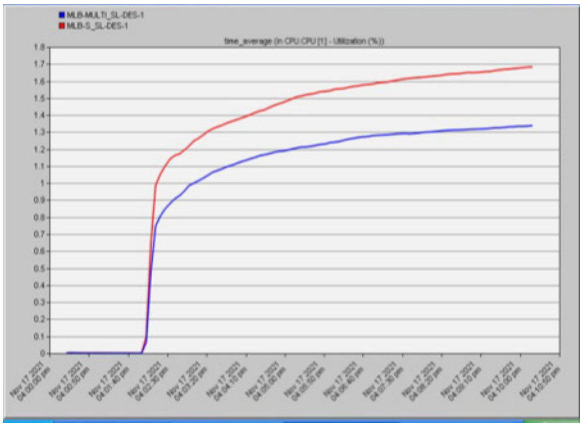


Fig. 15. Server Load CPU Utilization

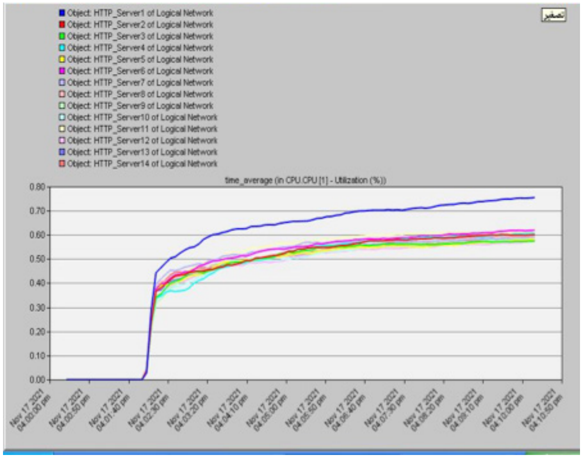


Fig. 16. Single Tier Round Robin Algorithm Servers CPU Utilization

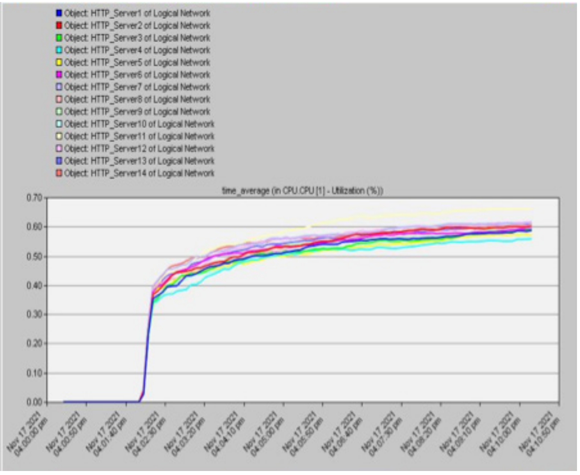


Fig. 17. Single Tier Number of Connection Algorithm Servers CPU Utilization

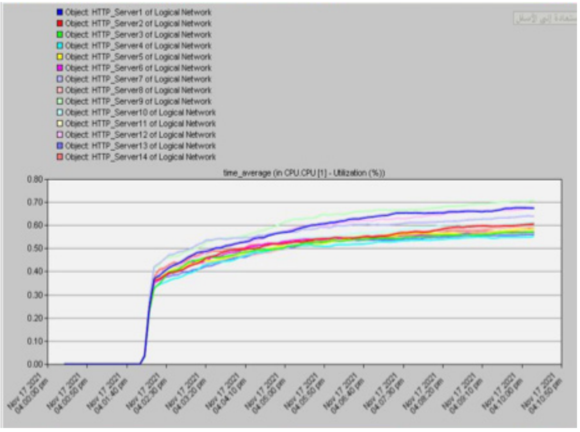


Fig. 18. Single Tier Server Load Algorithm Servers CPU Utilization

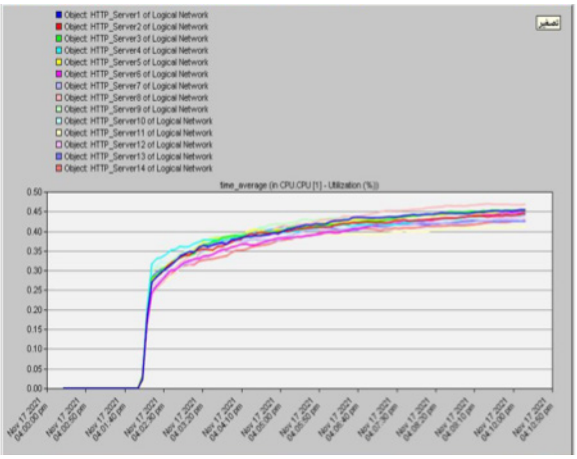


Fig. 19. Multi-Tier Round Robin Algorithm Servers CPU Utilization

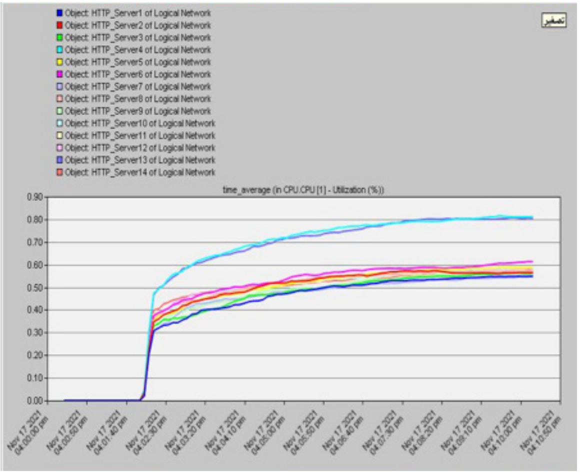


Fig. 20. Multi-Tier Number of Connection Algorithm Servers CPU Utilization

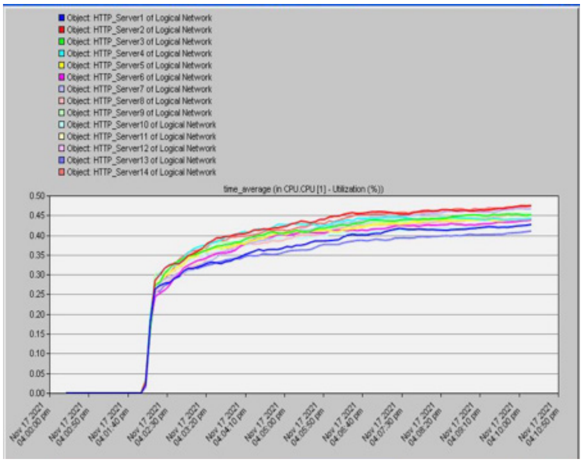


Fig. 21. .Multi-Tier Server Load Algorithm Servers CPU Utilization

Conclusion

The increased volume of internet traffic requires innovative solutions. The current server cluster architecture depends on the Single Load balancer tier. In this paper, the researchers introduced a new Multi Load balancer tier to reduce the load at the main Load Balancer. The researchers conducted two experiments to test the performance of Single Load balancer tier and Multi Load balancer architectures. The experiments showed that the multi-tier solution reduced the main load balancer CPU Utilization because it distributed the traffic to two devices rather than directing the load to 16 servers. A problem with the Multi-Tier solution could be the delay in Response Time Average. The implication of this study could be the need to think about innovative architecture that supports emerging technologies, such as Cloud Computing and the Internet of Things. Future research could include the investigation of the impact of architecture with heterogeneous servers.

References

[1] A. K. Arahunashi, Gourav G Vaidya,

Neethu S, K. Viswavardhan Reddy, 2018. “Implementation of Server Load Balancing Techniques Using Software-Defined Networking,” 3rd IEEE International Conference on Computational Systems and Information Technology for Sustainable Solutions.

[2] Rhonda Ascierto, Andy Lawrence, 2020. “Uptime Institute global data center survey 2020”, Utime Institute.

[3] X. Zongyu, Xingxuan Wang, 2014. “A Modified Round-robin Load-balancing Algorithm for Cluster-based Web Servers” Proceedings of the 33rd Chinese Control Conference, Nanjing, China, July 28-30.

[4] A. Lawrence, “Annual outage analysis 2021, The causes and impacts of data center outages,” Utime Institute.

[5] D. Sharma, 2018. “Response Time Based Balancing of Load in Web Server Clusters,” 978-1-5386-4692-2/18/$31.00, IEEE.

[6] Mochamad Rexa Mei Bella, Mahendra Data, Widhi Yahya, 2018. “Web Server Load Balancing Based On Memory Utilization Using Docker Swarm,” 978-1-5386-7407-9/18/$31.00, IEEE.

[7] W. Zhang, 2000. “Linux Virtual Server For Scalable Network Services,” National Laboratory for Parallel & Distributed Processing,

[8] YU SHENGSHENG, YANG LIHUI, LU SONG, ZHOU JINGLI, 2005. “Least-Connection Algorithm based on variable weight for multimedia transmission”.

[9] Krisna Wahyu Murti, Tengku Ahmad Riza, Asep Mulyana, 2020. “Comparative Analysis of Load Balancing Dynamic Ratio and Server Ratio Algorithms” 2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE).

[10] S. Ghosh and C. Banerjee, 2018. “Dynamic Time Quantum Priority Based Round Robin for Load Balancing In Cloud Environment,” Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN).

[11] R. S. Sajjan, B. R. Yashwantrao 2019. “Load Balancing and its Algorithms in Cloud Computing: A Survey,” International Journal of Computer Sciences and Engineering.

[12] A B M Moniruzzaman, Syed Akhter Hossain, 2014. “A Low Cost Two-Tier Architecture Model For High Availability Clusters Application Load Balancing”.

[13] Ying-Dar Lin, Chih Chiang Wang, Yi-Jen Lu, Yuan-Cheng Lai, and Hsi-Chang Yang, 2018. “Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks”.

[14] Pol Serra i Lidón, Giuseppe Caso, Luca De Nardis, Alireza Mohammadpour, Eljona Zanaj, and Maria-Gabriella Di Benedetto, 2019. “Two-tier Architecture for NB-IoT: Improving Coverage and Load Balancing”.

[15] Ying-Dar Lin, Widhi Yahya, Chien-Ting Wang, g, Chi-Yu Li, and Jeans H. Tseng, 2021. ” Scalable Mobile Edge Computing: A Two-tier Multi-Site Mul-

ti-Server Architecture with Autoscaling and Offloading”.

[16] V. R. Kanakala, V. K. Reddy, 2015. “Performance Analysis of Load Balancing Techniques in Cloud Computing Environment,” TELKOMNIKA Indonesian Journal of Electrical Engineering, Vol. 13, No. 3.

[17] Amjad Mahmood, Irfan Rashid , 2011. “Comparison of Load Balancing Algorithms for Clustered Web Servers,” Proceedings of the 5th International Conference on IT & Multimedia at UNITEN (ICIMU 2011).

[18] Brajendra Kumar , Dr. Vineet Richhariya2, 2014. Load Balancing of Web Server System Using Service Queue Length , International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 5.

[19] Y. M. Teo and R Ayani, 2001. “Comparison of Load Balancing Strategies on Cluster-based Web Servers,” Transactions of the Society for Modeling and Simulation (accepted for publication).

[20] S. G. Thorenoor, 2010. “Communication Service Provider’s Choice between OSPF and IS-IS Dynamic Routing Protocols and Implementation Criteria Using OPNET Simulator,” Second International Conference on Computer and Network Technology, Bangkok, 38-42