Research On Raspberry Pi 3b+ For Load Balancing On 2 Web Servers

1st Alfarabi Ivandi Noor
School of Electrical Engineering
Telkom University
Bandung, Indonesia
alfarabiin@student.telkomuniversity.ac.id

2nd Sofia Naning Hertiana
School of Electrical Engineering
Telkom University
Bandung, Indonesia
sofiananing@telkomuniversity.ac.id

3rd Arif Indra Irawan
School of Electrical Engineering
Telkom University
Bandung, Indonesia
arifirawan@telkomuniversity.ac.id

Abstract— Augmenting the web server's capacity may lead to system failures or excessive load, ultimately resulting in server downtime. In order to address this issue, a load-balancing strategy is employed. This technique involves the distribution of incoming requests over multiple servers simultaneously, thereby reducing the workload absorbed by each individual server. The scalability of the system enables the equitable distribution of server traffic loads across numerous connection lines, optimizing throughput, enhancing reaction time, and mitigating the risk of overload on any single connection line. The Nginx load balancer program will be utilized, employing the Round Robin and Least Connection algorithms. These methods will be connected to two Django web servers. In the analysis of the Throughput, Delay, and Response time metrics for the two algorithms, it was observed that there were no statistically significant differences, with the exception of the tests conducted with 5000 and 6000 requests. Therefore, it can be inferred that the utilization of Round-Robin and Least Connection algorithms is appropriate for a website experiencing traffic volumes ranging from 500 to 4000 requests per minute, with content sizes that are not excessively large, in order to prevent any potential issues during the server load sharing procedure.

Keywords — Raspberry Pi 3B+, Load Balancing, Nginx, Django, Round Robin, Least Connection.

I. INTRODUCTION

The use of the internet is increasing due to the digitalization of technology, making traffic on a web server increase significantly [1]. According to Polling Indonesia, in collaboration with the Association of Indonesian Internet Service Providers, or APJII (2018), the overall number of internet users in Indonesia has reached 171.17 million individuals, representing approximately 64.8% of the country's population [2], and from 2022 until 2023, there will be 215.63 million internet users. If not resolved, it will result in crashes or overload, which will cause the server to go down. Increasing traffic will basically make server performance even harder. Website services are often provided via a server, commonly known as a web server [3]. The server's operating system is specific in order for the server to work properly. In general, the more traffic that enters a web server, the more the website service cannot run optimally or even be accessed because the web server cannot handle the large amount of incoming traffic [3]. When dealing with issues on this web server, load balancing, or a procedure that splits the traffic load into two or more connection lines evenly amongst numerous computers, network cables, hard drives, or other resources, is used to handle difficulties on this web server [4]. In this load-balancing process, the web server traffic paths are divided using a round-robin algorithm so that the workload of the web server will be reduced and the quality of the services provided will increase.

Research on load-balancing on web servers has also been carried out a lot before, such as in “RESEARCH ON THE IMPLEMENTATION OF LOAD BALANCING ON WEB SERVERS USING FUZZY ALGORITHMS IN SOFTWARE DEFINED” by Miftah Nur Fauziah. In addition, research on load balancing has also been carried out by Ariyani Abdullah regarding “THE IMPLEMENTATION AND ANALYSIS OF LOAD BALANCING ON WEB SERVERS”. Closer research on load balancing has also been carried out by Dhuka Dwi Cahyanto regarding “THE IMPLEMENTATION AND ANALYSIS OF WEB SERVER LOAD BALANCING USING THE RECEIVER PULL BASED DISPATCHER METHOD”. However, these studies have significant differences from the current study, where this study uses the Raspberry Pi 3B+ as a router used for media load balancing, which proves that the Raspberry Pi 3B+ can be used for various functions, including load balancing.

This research is important because it provides an update in load balancing research, specifically the use of the Raspberry Pi 3B+ as a router for media load balancing using Nginx, which is also used as the main media web server and backup web server, proving that the Raspberry Pi 3B+ function can be used in a variety of situations. This research is essential for load balancing to make it more effective and adaptable, minimizing overload or server downtime, and other Raspberry Pi 3B+ tasks. Based on these considerations, research on "RESEARCH ON RASPBERRY PI 3B+ FOR LOAD BALANCING ON 2 WEB SERVERS" is feasible.

II. BASIC CONCEPT

A. Raspberry Pi

The Raspberry Pi is a single-board computer (SBC) the size of a credit card that can run office applications, video games, and media players. Raspberry Pi is made up of parts that resemble those of most computers. such as CPU, GPU, RAM, HDMI, Ethernet, USB Port, Audio Jack, and GPIO.

The Raspberry Pi operating system employs a Micro SD card with a size of at least 4 GB instead of a hard disc drive (HDD) for data storage, and the power source is micro-USB, which has a recommended power source of 5V and a minimum current of 700 mA.

B. Switch

An OSI (Open Systems Interconnection) reference model layer 2 (Data Link Layer) network device is a switch. Its job is to link up network devices in a local area network (LAN) and transfer data among them. The switch can read a device's MAC (Media Access Control) address and use that information to direct data to the desired device.

C. Load Balancing

Load balancing is a fundamental principle in the field of information technology, wherein the objective is to equitably balance the workload or traffic among several computing resources, such as servers, networks, or services. The primary objective of load balancing is to enhance system performance, reliability, efficiency, and availability by assuring equitable distribution of workload across resources, hence preventing resource overutilization and underutilization [2].

In the context of servers or networks, load balancing is frequently employed to effectively handle user traffic or requests directed towards a service, application, or website. Load balancing encompasses a variety of techniques, with five algorithms currently in widespread popularity [6]. The following discourse presents an enumeration of the five most prevalent algorithms employed in load balancing, with several others which are Round Robin, Least Connection, Least Respond Time, Weighted Round Robin, and IP Hash.

D. Nginx

Nginx is a versatile software application that may operate as a web server, reverse proxy server, load balancer, and fulfil other network-related functions. Nginx is specifically engineered to deliver superior levels of performance, scalability, and efficiency while managing web traffic [7].

E. Wireshark

M Wireshark is a software application that operates as an open-source network analysis tool, specifically designed to serve as a network protocol analyzer. Its primary function is to monitor and thoroughly analyze network traffic. Wireshark is a software tool that enables the capturing, analysis, and inspection of data packets transmitted and received by network devices. Wireshark is a software tool that provides support for a wide range of network protocols, enabling users to get insights into the interactions occurring between devices inside a network environment [9].

F. Apache Jmeter

Apache JMeter is a freely available software that is utilized for the purpose of conducting performance, load, and functionality assessments on online applications or services. The utilization of load scenarios in testing system response and performance is a highly effective technique. Apache JMeter enables the replication of workloads generated by numerous virtual users, hence facilitating the evaluation of system performance under such conditions [10].

G. Quality of Service

Quality of service (QoS) is a way or method of measurement used to determine the capabilities of a network such as; host, network application or router with the intention of providing planned and much better network services so as to meet the needs of a service.

III. SYSTEM DESIGN

A. System Topology

The topology below depicts the procedure of the Nginx load balancer using the Round Robin and Least connection Algorithm approach from the client to the Raspberry Pi 3B+ as a media load balancer

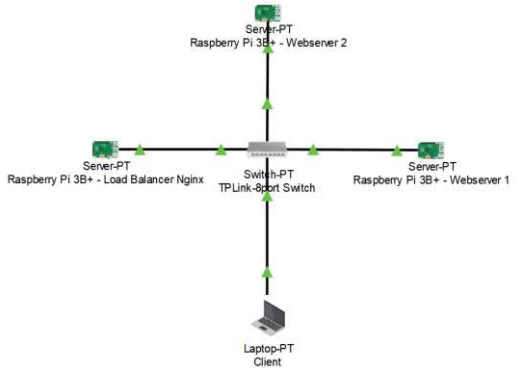

FIGURE 3.1.
System Topology

B. Raspberry Pi 3B+ Specification

TABLE 3.1.
Raspberry Pi 3B+ Specification

Specification	Raspberry Pi 3B+
SOC	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @1.4 Ghz
RAM size	1GB LPDDR2 SDRAM
Memory	Micro SD port
OS	Raspian version 11 (Bullseye)
Ethernet Speed	300Mbps
Power	5V/2.5A DC power input

C. Support Infrastructure

This journal incorporates the use of both hardware and software components, together with frameworks, to facilitate the comprehensive load balancing system

TABLE 3.3.
Software and Framework

No	Name	Type
1	Python 3.9	Programming Language
2	Django	Framework
3	Nginx	Software
4	Round Robin	Algorithm
5	Least Connection	Algorithm
6	Apache Jmeter	Software
7	Wireshark	Software

D. System Design

The following figure is a flowchart depicting a comprehensive outline of the system that is intended to be constructed in the course of this study. The diagram illustrates the complete sequence, commencing with the initiation of the system and concluding with the outcomes derived from the established parameters.

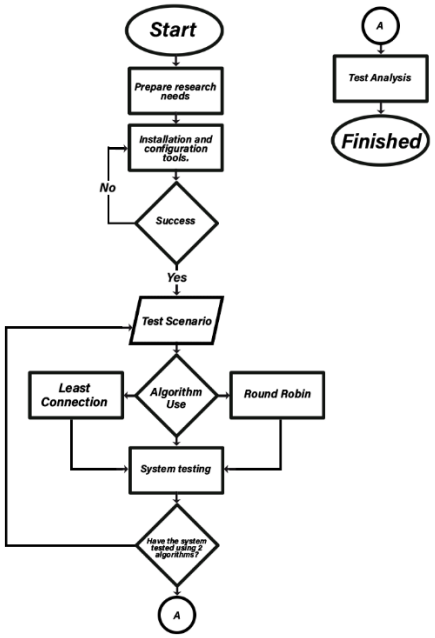

FIGURE 3.2.
System Design Flowchart

E. Prepare research needs

At present, it is necessary to ensure the meticulous preparation of both hardware and software components involved in the development of this load balancing system. The subsequent follow as 3 Raspberry Pi 3B+, 1 Laptop, 1 Switchport, 4 LAN cable.

F. Installation and configuration tools

This process will elucidate the installation procedure for each requirement on the load balancer and web server, along with the configuration steps for the load balancer

G. Implement load balancing algorithm

Load balancing is a widely employed strategy in computer systems and networks that involves the equitable distribution of traffic or tasks over numerous servers or computing resources. The primary objectives of load balancing are to optimise performance, prevent excessive strain on individual components, and enhance overall system availability. Algorithms play a crucial role in load balancing by facilitating the efficient and effective distribution of workload.

H. Test Scenario using Round Robin algorithm.

The experimental procedure involves the initiation of a predetermined quantity of requests towards a domain load balancer. This load balancer has been configured using the Apache Jmeter application on the client side. In addition, the monitoring of requests will be conducted through the analysis of logs obtained from individual web servers. The objective of this analysis is to examine the effective management of request distribution to web servers by a load balancer through the utilization of the Round Robin algorithm.

I. Test Scenario using Least Connection algorithm.

In this experiment, the load balancing capability of the nginx server will be evaluated using the round robin technique. A notable distinction is in the utilization of the least connection algorithm by the algorithm implemented on the Nginx load balancer. Next, we will observe the manner in which the load balancer handles the request to distribute the workload between the two web servers

J. Test analysis

Analysis tests serve as valuable tools for evaluating the performance, functionality, and efficacy of the nginx load balancing system. The final project encompasses an analytical test that encompasses various topics.

IV. RESULT AND ANALYSIS

A. Load balancing test result

The evaluation of load balancing is conducted through the implementation of two distinct scenarios, specifically employing the Round Robin and Least Connection algorithms. Each test is conducted within an approximate duration of 60 seconds, and the number of requests will be incrementally augmented till reaching the optimal threshold of the load balancer. The parameters that were tested were Throughput, Delay, and Response time

B. Throughput (rps)

The acquired throughput results exhibit relatively little variations between the two techniques. In the experimental procedure including 500 requests, a marginal discrepancy of 0.05 milliseconds was observed. Subsequently, when doing tests with 1000, 2000, 3000, and 4000 requests, an identical value was consistently achieved. In the experiment involving 5000 queries, a discrepancy of 4.7 milliseconds was seen, while in the experiment involving 6000 requests, a discrepancy of 10.7 milliseconds was observed.

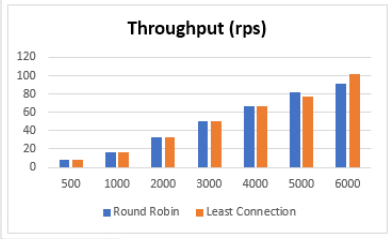

FIGURE 4. 1.
Throughput test Graph

However, a notable disparity in the quantity of requests is observed between 5000 and 6000. This discrepancy arises from an error encountered during the load balancing procedure, resulting in substantial variations in data between the two processes.

C. Delay

The findings obtained in the delay test were quite significant, with the exception of the test conducted on the number of requests, specifically 6000. When employing the Round Robin technique, the mean value observed was 0.754 milliseconds, but the Least Connection algorithm yielded an average value of 0.0005 milliseconds. The occurrence of this event can be attributed to an error that occurred during the load balancing process, specifically in the implementation of the Round Robin algorithm, which accounted for 44.13% of the errors. In contrast, the Least Connection procedure experienced a significantly lower error rate of 11.63%.

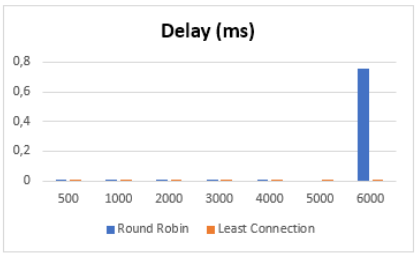

FIGURE 4. 2.
Delay test Graph

D. Response time

The findings of the response time test indicate that there was no substantial difference in values between the two procedures. The values obtained in both load balancing methods were within the range of normalcy.

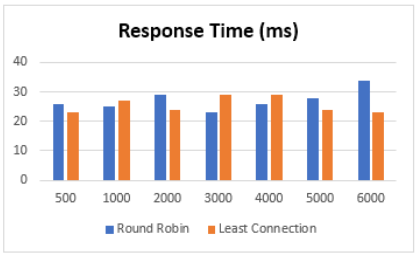

FIGURE 4. 3.
Response time test graph

E. Load balancer performance result

This section will present an analysis of the performance of load balancing hardware through testing, employing two distinct techniques. The characteristics that were examined encompassed the utilization of the central processing unit (CPU) and the random-access memory (RAM).

F. CPU Usage (%)

During the examination of CPU consumption, a noteworthy rise in CPU usage was seen when the number of requests increased from 2000 to 6000 for both algorithms.

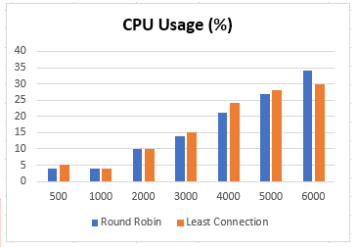

FIGURE 4. 1
CPU Usage test graph

G. RAM Usage (%)

The Raspberry Pi 3B+ is equipped with a modest amount of RAM, specifically 1 GB. In the course of conducting tests, it was observed that a sudden surge of 4000 requests occurred while employing the Least Connection technique. In the processes numbered 5000 and 6000, a decline was observed due to problems encountered in the respective request procedures.

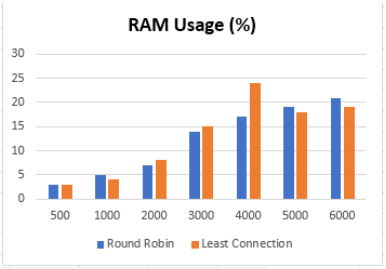

FIGURE 4.5
RAM Usage test graph

V. CONCLUSION AND SUGGESTION

Based on the findings and analysis conducted in this study, it can be inferred that the Load Balancing procedure employed on the Raspberry Pi 3B+ for the purpose of managing two web servers is deemed suitable for implementation on websites characterized by relatively low levels of incoming traffic, typically ranging from approximately 500 to 4000 requests per minute. This load balancing strategy is particularly well-suited for websites that are of a relatively small size, such as blogs or personal websites. The utilization of the Round Robin and Least Connection algorithms in the load balancing procedure

exhibits minimal disparity when evaluated on a website with relatively lightweight content. The examination of the two methods in each test reveals a notable disparity arising from faults encountered during the load balancing procedure.

In summary, the utilization of the Round Robin and Least Connection algorithms in the load balancing procedure for a website experiencing a traffic rate ranging from 500 to 4000 per minute, with relatively lightweight website content, is very appropriate for implementation. In the event that one of these methods is employed for a substantial number of requests over 4000 or for highly utilized content, such as the inclusion of large-sized films or photographs, it is conceivable that an error may arise throughout the load sharing procedure, hence impeding user satisfaction.

BIBLIOGRAPHY

[1] M. Arman, N. Wijaya, and H. Irsyad, "Analisis Kinerja Web Server Menggunakan Algoritma Round Robin dan Least Connection," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 8, no. 2, pp. 55–59, 2017, doi: 10.32736/sisfokom.v6i1.143.

[2] A. I. Fahmi Apriliansyah, Iskandar Fitri, "Implementasi Load BalancingPada Web Server Menggunakan Nginx," *17th Int. Conf. Geoinformatics - Theor. Appl. Asp.*, vol. 6, no. 1, 2018, doi: 10.3997/2214-4609.201801770.

[3] Hanang and A. Nugroho, "Analisis performansi web server menggunakan load balancing server dengan algoritma least connection," *Repos. Institusi ST3 Telkom Purwokerto*, vol. 57, pp. 1–5, 2018.

[4] D. T. Nugrahadi, R. Herteno, and M. Anshari, "Pengaruh Implementasi Load Balancing Dan Tuning Web Server Pada Response Time Raspberry Pi," *Klik - Kumpul. J. Ilmu Komput.*, vol. 6, no. 2, p. 211, 2019, doi: 10.20527/klik.v6i2.249.

Conference Series, 2017.

[5] R. H. Putra and W. Sugeng, "Implementasi Cluster Server pada Raspberry Pi dengan Menggunakan Metode Load Balancing," *J. Edukasi dan Penelit. Inform.*, vol. 5, no. 1, pp. 41–42, 2016, doi: 10.26418/jp.v2i1.14988.

[6] I. G. L. P. E. P. Supramana, "IMPLEMENTASI LOAD BALANCING PADA WEB SERVER DENGAN MENGGUNAKAN Supramana I Gusti Lanang Putra Eka Prismana," *J. Manaj. Inform.*, vol. 5, no. 2, pp. 117–125, 2016.

[7] AMT IT Solution, "Apa Itu NGINX? Ini Fungsi dan Kelebihannya," *Angewandte Chemie International Edition, 6(11), 951–952.*, 2023. https://amt-it.com/blog/apa-itu-nginx/

[8] D. K. Hakim, D. Y. Yulianto, and A. Fauzan, "Pengujian Algoritma Load Balancing pada Web Server Menggunakan NGINX," *JRST (Jurnal Ris. Sains dan Teknol.*, vol. 3, no. 2, p. 85, 2019, doi: 10.30595/jrst.v3i2.5165.

[9] Wireshark, "What is Wireshark?" https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs

[10] A. S. Gillis, "Apache Jmeter," 2021, [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/Apache-JMeter

[11] R. Julianto, W. Yahya, and S. R. Akbar, "Implementasi Load Balancing Di Web Server Menggunakan Metode Berbasis Sumber Daya CPU Pada Software Defined Networking," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 1, no. 9, pp. 904–914, 2017.