Jurnal Informatika dan Rekayasa Perangkat Lunak ISSN 2656-2855
Vol. 8, No. 2, September 2026, Page. 207-212 e-ISSN 2685-5518

Web Server Performance Comparison Analysis: Apache and Nginx on Ubuntu Server-based Docker Container

Hanya Abriananta[1], **Hery Mustofa**[2], **Nur Cahyo Hendro Wibowo**[3]
[1,2] Prodi Teknologi Informasi , Fakultas Sains dan Teknologi, Universitas Islam Negeri Walisongo Semarang
[*]Email: hanyaananta@gmail.com

Abstract

This study aims to analyze and compare the performance of two widely used web servers, Apache and Nginx, operating within Docker container environments on Ubuntu Server. The evaluation focuses on determining the efficiency of each web server in handling client requests, particularly in terms of throughput, response time, latency, resource utilization, and error rate. An experimental method was employed, where both servers were configured in separate containers and tested using Apache Benchmark (ab) and Wrk Benchmark under various load scenarios. The results indicate that Nginx outperforms Apache across most performance parameters. Nginx achieved a throughput of 1316.86 requests per second, which is more than twice the throughput of Apache at 631.56 requests per second. Nginx also demonstrated higher transfer rates and lower latency, with both servers recording a 3% error rate during testing. Furthermore, Nginx exhibited more efficient CPU and memory usage compared to Apache. The findings conclude that Nginx is more suitable for high-load environments requiring fast response and efficient resource consumption, while Apache remains relevant for applications that rely on dynamic content processing and flexible module support.

Keywords: *Apache, Nginx, Docker, web server performance, benchmarking.*

INTRODUCTION

Client devices, such as web browsers, obtain content from web servers, which can be located anywhere in the world and connected to the Internet or a local area network. Web servers act as intermediaries between clients and the information they seek, processing and delivering requests for web pages and other content. For client-server connections, web servers use the Hypertext Transfer Protocol (HTTP). They may also support other protocols, including HTTPS (HTTP over SSL/TLS), for secure communication (A.Tedyyana and O. Ghazali, 2021). Content that can be delivered by a web server includes static files such as HTML pages, images, and videos, as well as dynamic files created by scripts or applications running on the server side, such as Python, Java, or PHP (Riswandi et al., 2020).

Due to its reliability and security, Ubuntu Linux, an open-source distribution built on Debian, is commonly used for server applications. Network service customers can benefit from a dedicated, efficient, and flexible Ubuntu server. Installing Linux on a laptop can improve user access to Linux servers and simplify the functionality of various devices and international network connections (Rahmadhani, 2025).

Many large companies use Nginx, an open-source web server technology, to manage their websites. Websites and online applications rely on the Nginx web server, which processes HTTP and HTTPS requests and manages multiple clients simultaneously. Nginx is a web server that can handle multiple requests from clients simultaneously, allocate resources easily and adaptively, act as a load balancer and reverse proxy, and has low resource requirements compared to other platforms. It is also compatible with modern technologies like Docker (Rafli, 2022). Nginx isn't perfect; it requires additional plugins or modules to fully enable dynamic content functionality and has limited support by default.

A large number of countries use Apache, an open-source web server software. The Apache web server manages authentication and access rights, speeds up website load times using caching and proper configuration, and can be integrated with modern technologies such as MySQL, Perl, and PHP (Kurniawan et al., 2021). Apache is a web server that has many advantages, such as being compatible with many additional modules and plugins that make it

more powerful, as well as having a configuration process that even beginners can understand and use (Komaruddin et al., 2020). Compared to Nginx, the Apache web server uses more resources and is not very good at handling multiple requests from multiple clients at once.

Nginx is renowned for its speed and efficiency in handling multiple requests from multiple clients simultaneously, while Apache is highly regarded for its flexibility and multitasking capabilities. The purpose of this study is to compare and contrast these two web server platforms by running tests on each platform and analyzing the results. Examining how Nginx and Apache, two popular web servers, affect website speed and security is the primary objective of this study.

The success of online businesses depends on the performance and security of their websites, which is why this study examines modern advances in web technology. A web server solution that can handle requests quickly and efficiently is essential for high-traffic e-commerce websites, but a more flexible platform may be better for businesses with lower traffic. Additional factors, including price, security, and ease of use, need to be considered in this study when choosing the right web server platform. Improving website performance and user experience starts with choosing the right web server platform (Saputra et al., 2023).

To help users choose the best platform for their needs, this study will conduct a thorough evaluation of the performance of Nginx vs Apache web servers.

LITERATURE REVIEW

Web servers are fundamental to delivering web-based services. The two main servers used in this context are Apache and Nginx. Apache is recognized as a versatile server due to its implementation of a process-based methodology, where each client request is managed by an individual process or thread. Unlike Apache, Nginx uses an event-driven architecture that allows a single process to manage multiple connections simultaneously. This architectural design allows Nginx to better accommodate both increasing traffic levels and static requests with reduced latency (Fitriana et al., 2023).

The use of Docker as a containerization platform is increasing in prevalence, driven by its lightweight, isolated, and easily replicable environment. Leveraging Docker allows web services to operate on the same operating system without mutual interference, thus improving the consistency of web server performance evaluations compared to conventional environments. Ubuntu Server is often used as a base operating system due to its robustness and wide package availability, making it a good choice to serve as a base Docker image in performance experiments.

Previous research compared the performance of Apache and Nginx using various testing tools and conducted performance analysis using Apache Bench with varying loads ranging from 100 to 1,000,000. The results of the study showed that Nginx consistently took less time than Apache to complete client requests at various load levels, which is early evidence of the superiority of the Nginx architecture in terms of response time efficiency (Chandra, 2019). Meanwhile, another study also deepened the study by using Apache JMeter as a testing tool to measure response time and throughput in multi-client conditions. Test results with variations of 100, 1,000, and 10,000 requests showed that Nginx had an average response time approximately twice as fast and was able to process almost twice as many requests as Apache in the same duration. Furthermore, Nginx did not fail to respond, while Apache experienced an error rate of up to 1% in each test scenario (Ridwan et al., 2020).

Several previous studies have shown that Nginx generally exhibits superior throughput and reduced resource consumption relative to Apache, particularly in scenarios characterized by high concurrent request volumes. Nonetheless, Apache continues to maintain a competitive advantage with regard to module compatibility and its capacity to manage certain dynamic content applications. Consequently, assessing Apache and Nginx performance metrics in containerized environments, such as Docker, is relevant to explaining the operational dynamics of these different architectures on contemporary platforms. Standard performance assessments typically include evaluation of response time, throughput, resource utilization (CPU and RAM), and server stability under high load conditions.

METHOD

A method is a set of techniques used to collect, analyze, and interpret data or

information to answer research questions or evaluate hypotheses. The primary goal of research methodology is to produce authentic, accurate, and reliable knowledge in the domain being studied (Perdana et al., 2025).

A methodological experimental approach was used to evaluate the relative performance of Apache and Nginx web servers implemented on Ubuntu Server. Each web server was configured to run a basic HTML-based website.

Performance testing is performed using tools like Wrk Benchmark to evaluate technical parameters such as response time, throughput, latency, and resource usage (CPU and RAM). Testing is performed under various load scenarios, such as gradually increasing request counts, to observe each web server's ability to handle increasing user numbers.

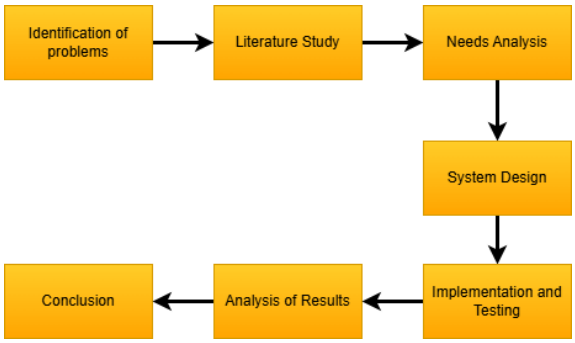


Figure 1. Experimental Method

The following is a flow chart of the stages involved in this research. The research stages involved several important steps, which are explained below:

A. Problem Idenfication

This step aims to identify key performance issues with Apache and Nginx when running in an Ubuntu Server-based Docker environment. The focus is on parameters such as response time, throughput, latency, and resource usage.

B. Literature Study

Collect data from various sources, including journals, books, and government documents, to understand the technical specifications and advantages of each web server (Apache and Nginx). Furthermore, the literature review also includes performance testing tools such as Apache Benchmark (ab) and Wrk Benchmark, as well as statistical methods for data analysis.

C. System Design

The system design in this study was designed to ensure that the performance comparison between Apache and Nginx could be performed in a measurable, standardized, and replicable manner. The test environment was built on a virtualization platform using Ubuntu Server as the base operating system. Docker was used as the container engine to create two separate containers, one running the Apache web server and the other running Nginx. Both containers were configured using the official images of each server and provided with static HTML files.

Each container runs on a different port to ensure simultaneous access without conflict. Next, benchmarking tools like Apache Benchmark (ab) and Wrk Benchmark are used to load requests with specific parameters, including the number of requests, concurrency level, and test duration. The entire performance monitoring process is performed using benchmark output and the docker stats command to measure CPU and memory usage.

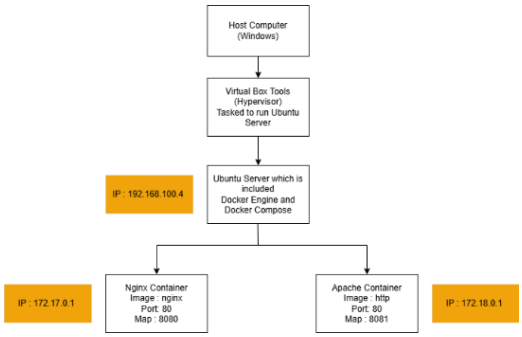


Figure 2. System Design

D. Testing

This test uses Apache Benchmark (ab) and Wrk Benchmark, tools for measuring how a server handles user load, including load and stress tests. The study used 100, 500, 1000, and 2000 connections for 60 seconds. The parameters tested using Apache Benchmark and work were:

1. Response time
2. Throughput
3. Latency
4. Request Per Second
5. Transfer Rate

6. Use CPU and Memory Usage
7. *Error Rate*

Tools and Materials include the hardware and software used to support the research process. The following are the specifications for the software used.

Table 1 Computer Specifications

Component	Specifications
Ram	8 GB
Processor	Inter I3
Storage	SSD 256 GB

Table 2 Software Used

Software	Type
Virtualbox	Virtualization
Apache Benchmark	Testing Tools
Wrk Benchmark	Testing Tools
Apache	Web Server
Nginx	Web Server

RESULT AND DISCUSSION

3.1 Response time

The smaller the number, the faster the time it takes for the web server to respond to a request.

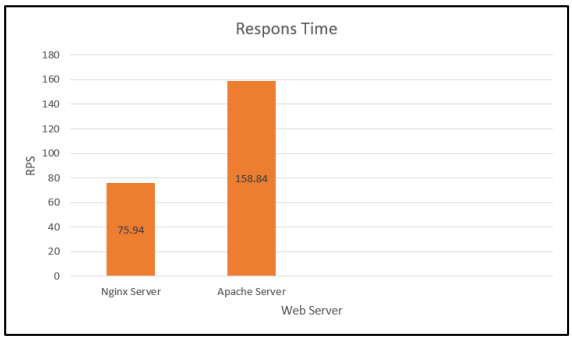


Figure 3. Response Time Comparison Graph

3.2 Throughput

A comparison of data throughput (MB/s) is presented graphically to illustrate each web server's ability to process and transmit data. The higher the number, the more optimal the web server's performance in processing high data volumes.

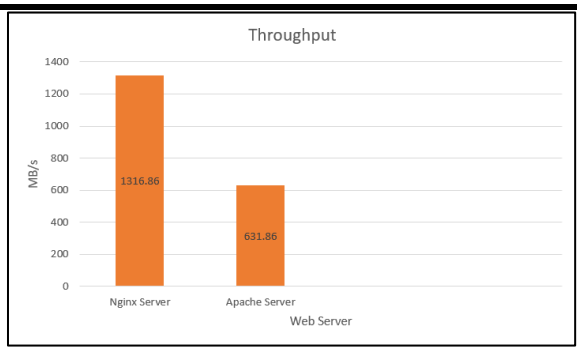


Figure 4. Throughput Comparison Chart

3.3 Latency

The test results show that Nginx achieves significantly greater throughput values than Apache.

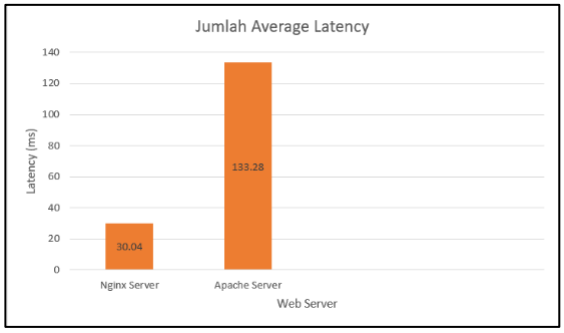


Figure 5. Delay Time Comparison Chart

3.4 Request Per Second

Requests per second is measured in Transactions per Second (TPS), the number of transactions that can be processed in one second. The higher the number, the more requests the web server can process, indicating more optimal performance.

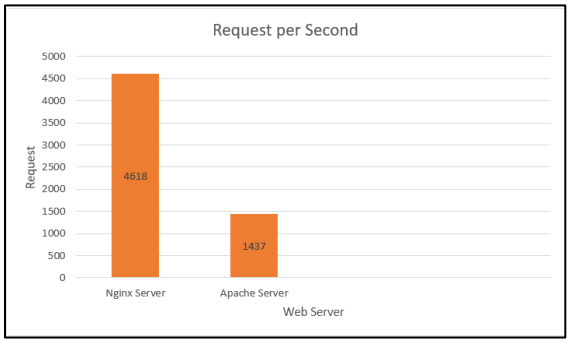


Figure 6. Request Per Second Comparison Chart

3.5 Transfer Rate

Test results show that Nginx achieves significantly higher throughput than Apache. This superior throughput indicates that Nginx offers more optimal performance.

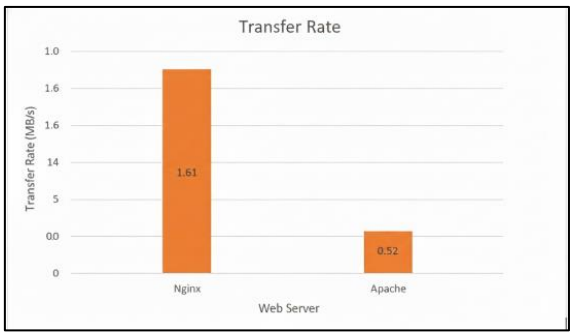


Figure 7. Transfer Rate Comparison Chart

3.6 Use CPU and Memory Usage

A comparison of resource usage is presented graphically to illustrate how efficiently each web server manages CPU and memory resources when processing requests. The lower the number, the more optimal the resource utilization and system performance.

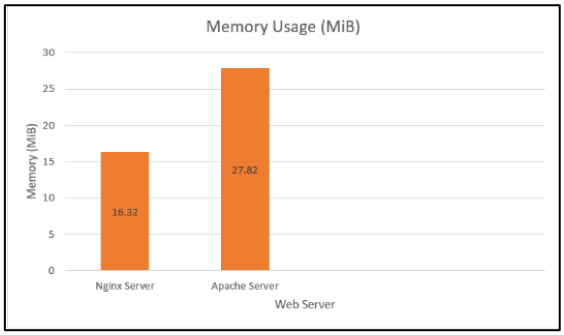


Figure 8. Memory Usage Comparison Chart

3.7 Percentage *Error Rate*

The error rate comparison is presented in graphical form to illustrate the reliability of each web server in processing and sending data. The lower the number, the better the web server's performance in handling requests with fewer errors.



Figure 9. Error Rate Comparison Chart

CONCLUSION

Based on performance testing between Apache and Nginx web servers running in an Ubuntu Server-based Docker Container environment, it can be concluded that Nginx outperforms Apache in most test parameters. Nginx was able to generate a throughput of 1,316.86 requests per second, significantly higher than Apache's 631.56 requests per second.

Additionally, Nginx also boasts a higher transfer rate, lower latency, and more efficient use of CPU and memory resources. Both web servers recorded a 0% error rate, indicating that they were able to handle all requests without failure in the given test scenario.

Overall, Nginx is more recommended for applications or services that require high-load handling and fast response. Meanwhile, Apache remains relevant for systems that require module flexibility, broad support for server-side technologies, and more complex configuration requirements.

The findings of this study serve as a reference for selecting the right web server based on application needs and operational context. In future research, several developments are possible, such as adding other web servers like Lighttpd, Caddy, or LiteSpeed as comparisons, resulting in a more comprehensive analysis, and adding testing with various hardware to achieve optimal results. Finally, testing can be conducted dynamically, such as using PHP or Python.

REFERENCES

A. Tedyyana and O. Ghazali. (2021). Teler Real-Time Http Intrusion Detection At Website With Nginx Web Server. Int. J. Informatics Vis., 5(3), 327–332.

Chandra, A. Y. (2019). *Analisis Performansi Antara Apache & Nginx Web Server dalam Menangani Client Request*. 48–56. https://doi.org/10.30864/jsi.v14i1.248.

Komaruddin, A. M., Sipitorini, D. M., & Rispian, P. (2020). Load Balancing dengan Metode Round Robin Untuk Pembagian Beban Kerja Web Server. *Siliwangi*, *5*(2), 47–50.

Kurniawan, M. F., Setianto, W., & Ilyas, A. (2021). Implementasi Module Google Pagespeed Pada Apache Web Server Untuk Meningkatkan Performa Aplikasi Web. *IC-Tech*, *16*(2), 47–51. https://doi.org/10.47775/ictech.v16i2.209

Perdana, G. P, Prihanto, A., (2025). *Analisis Perbandingan Performa Web Server Apache ,*. *06*, 1145–1155.

Seimahuira, S., Fitriana, L., A. (2023). *Penerapan Metode SAW dalam Analisa Perbandingan Performa Web server (Apache , Nginx , Lighttpd , Iis) pada Bahasa Pemrograman PHP*. *7*, 409–420.

Rafli, M. (2022). Jurnal Pengujian Kinerja Load Balancing Web Server menggunakan Nginx Riverse Proxy Berbasis OS Centos 7. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(3), 1824–1840. https://doi.org/10.35957/jatisi.v9i3.2185.

Rahmadhani, D. S, (2025). *ANALISIS KINERJA WEB SERVER BERBASIS LINUX*. *5*(1), 35–44.

Ridwan, A. (2020). *ANALISIS PERBANDINGAN PERFORMA APACHE WEB SERVER*. *8*(2), 87–92.

Riswandi, Kasim, & Raharjo, M. F. (2020). Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2. *Journal of Engineering, Technology, and Applied Science*, *2*(1), 19–31. https://doi.org/10.36079/lamintang.jetas-0201.92

Saputra, P. S., Pratama, P. A., Putu, L., Sri, A., Informasi, T., Teknik, F., Sakti, U. P., & Korespondensi, P. (2023). *PERANCANGAN DAN KOMPARASI WEB SERVER NGINX DENGAN WEB SERVER APACHE SERTA PEMANFAATAN REVERSE PROXY SERVER DESIGN AND COMPARISON OF NGINX WEB SERVER WITH APACHE WEB*. *2*(1), 16–21.