Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 3, July 2024
https://doi.org/10.47709/cnapc.v6i3.4383

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

Performance Analysis of an E-commerce Website Using Distributed Servers (Case Study: Ecommerce Bumdes Sarining Kukuh Winangun)

I Kadek Susila Satwika[1)*], **I Gede Andika**[2)]
[1)*2)]Program Studi Rekayasa Sistem Komputer, Institut Bisnis dan Teknologi Indonesia, Indonesia
[1)*]susila.satwika@instiki.ac.id, [2)]gdandika@instiki.ac.id

ABSTRACT

Digital transformation has significantly altered our interactions by leveraging the internet for fast global information exchange and collaboration. This shift impacts businesses, individuals, organizations, and governments by increasing the demand for fast, reliable web services. To meet these demands, many organizations are turning to distributed server infrastructures, which help handle large storage needs and avoid performance issues. This research focuses on creating a distributed web server system that uses Round-Robin load balancing to evenly distribute traffic and enhance performance. The Round-Robin algorithm's simplicity and effectiveness in balancing loads, combined with the advantages of virtualization, such as cost efficiency, improved performance, and better resource management are central to this approach. Virtualization also offers improved scalability, accuracy, and security, further enhancing overall system efficiency and effectiveness in data centers. This research evaluates and contrasts the performance of websites utilizing distributed servers against those using single servers. The findings indicate that websites with distributed servers significantly outperform those with single servers. Specifically, distributed servers offer response times that are 5.8 times faster, achieve 2.2 times more successful responses, and transfer 2.1 times more data than single servers. Additionally, single servers experience a much higher rate of timeouts, with 14.2 times more occurrences compared to distributed servers.

Keywords: Distributed; Server; Website; Virtualization; Round-Robin

1. INTRODUCTION

Digital transformation has fundamentally changed how we interact with the world. This shift impacts not only business operations but also how individuals, organizations, and governments adapt in the digital age (Rino Subekti et al. 2024). The internet, in particular, has become the backbone of this transformation, enabling rapid global information exchange and collaboration without geographic limitations. The performance of web servers is crucial for ensuring a satisfying user experience. As online applications and internet users grow rapidly, the demand for fast, reliable, and responsive web services increases. To meet these challenges, organizations and web service providers are optimizing their infrastructure, often by using distributed servers.

Many organizations now prefer distributed server infrastructure to boost the performance and availability of their web services. Strategically placing servers in a distributed manner within the web server infrastructure helps manage large storage demands, especially for multimedia files that need permanent storage (Wunarso et al. 2013). By spreading the workload across multiple servers, this architecture can prevent bottlenecks and enhance the overall quality of web services (Eludiora et al. 2010). However, implementing and managing such infrastructure is complex, requiring a thorough understanding of distributed architecture, load balancing strategies, fault management, and resource efficiency.

This research aims to create a distributed web server system utilizing Round-Robin load balancing to distribute traffic to backend servers. Studies indicate that the Round-Robin algorithm, known for its simplicity, effectively balances the load using time-sharing principles, where each task is allotted a time slot or quantum (Abdulkarim et al. 2019). The algorithm's straightforwardness and equitable load distribution make it a reliable and efficient option for load balancing in cloud environments (Ekre, Nimbarte, and Balamwar 2018).

Additionally, the research will incorporate the concept of virtualization, which offers substantial benefits in terms of efficiency and scalability. These advantages include cost savings, energy efficiency, simplified deployment,

* Corresponding author

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

enhanced performance, and support for multiple platforms (Arfeen et al. 2022). Virtualization enhances scalability and accuracy in host node emulation, thereby improving overall system performance (Song et al. 2020). In data centers, virtualization ensures excellent resource utilization, scalability, and high availability, leading to better efficiency and cost-effectiveness (Du, He, and Meng 2014). It also provides improvements in security, reliability, scalability, and resource management, further highlighting its efficiency benefits (Cuzzocrea, Mumolo, and Corona 2015).

2. LITERATURE REVIEW

Previous research has examined the use of distributed servers. One study, titled 'Resource Utilization Analysis of Hadoop Clusters Using Virtualization,' focused on measuring the utilization of Hadoop servers in a cluster with virtualization, using four distributed servers for the process. The findings indicated that increasing the number of virtual machines (VMs) reduced processing time (Satwika, Handika, and Swari 2022; Swari, Satwika, and Handika 2020).

Another study by Roisul Setiawan and colleagues, titled 'Implementation of High Availability Web Server Using Web Cluster and High Availability Proxy Methods on Single Board Computers (Case Study: UD Sendang Putra),' tested a web server with a Raspberry Pi Zero W. The results showed that the cluster server had an availability rate of 98.53% and was optimal for 110 users, maintaining resource usage within a safe limit of 80% (Setiawan et al. 2020).

However, there has been no specific measurement of application performance following the deployment of these distributed servers. This study aims to measure the performance of an e-commerce website in handling numerous simultaneous responses. Additionally, this research will apply virtualization systems, which offer benefits in terms of efficiency and scalability.

3. METHOD

Figure 1 illustrates the various stages of the research focused on optimizing the sales data forecasting computing process through parallel computing in a cloud environment.

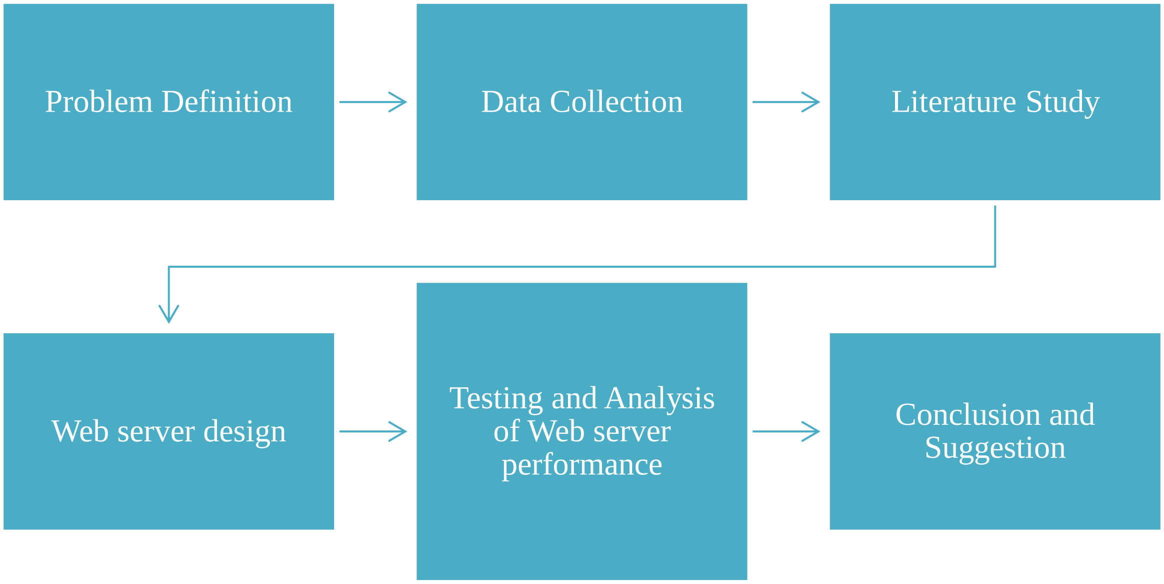

Problem Definition
Data Collection
Literature Study
Web server design
Testing and Analysis of Web server performance
Conclusion and Suggestion

Fig. 1 Research Stages

The researcher begins by defining the problem, observing issues in the field. Next, data collection and a literature review are conducted to support the research. Following this, the researcher designs the web server utilizing a distributed system and virtualization.

* Corresponding author

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

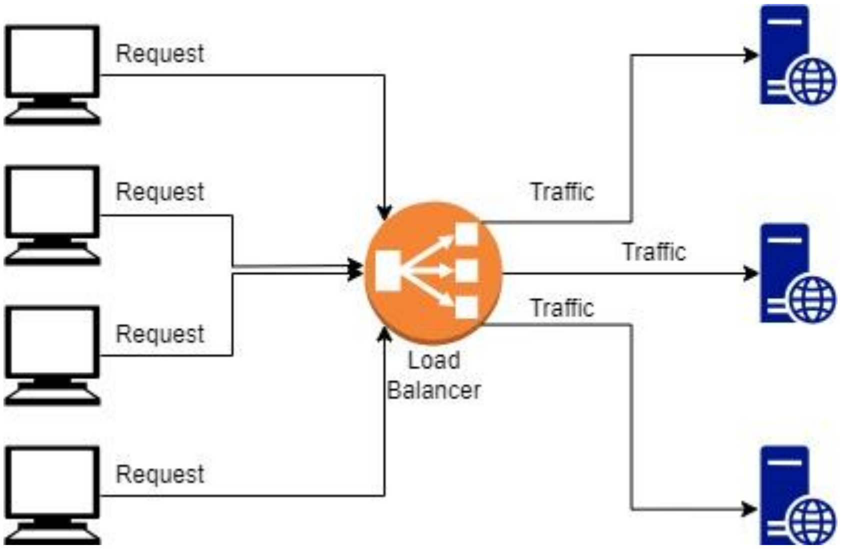


Fig. 2 Distributed Web Server Topology

In Figure 2, shows a topology that uses more than one separate web server. Then there is a server that serves as a traffic distributor. This server will use the round-robin load balancing method. This algorithm functions by assigning requests to each server in a rotating sequence, thereby ensuring an even distribution of the load among all available servers (Deshmukh, 2023). It is widely utilized in different computing contexts, such as cloud computing, software-defined networks, and virtual server clusters, to improve resource use and boost system performance (Alankar et al., 2020; Ekre et al., 2018; Motohashi et al., 2022). Opting for the round-robin load balancing algorithm offers several benefits according to existing literature. This straightforward and efficient method distributes incoming requests or workloads evenly across multiple servers in a rotating sequence (Deshmukh, 2023; Ramadhan & Wardhana, 2021). It guarantees equitable load distribution among servers, avoiding situations where one server becomes overloaded while others are underused (Alankar et al., 2020; Ma'arifah, 2024).

With this distributed server topology, it is expected to share traffic so that the web server will be able to receive more and more optimal requests. At this stage testing will be carried out on the load balancing server side and also the web server. This test is done using the loader.io application. loader.io is a load testing service that allows to stress test the web-apps & APIs with thousands of concurrent connections. This test involved 300 users accessing the system simultaneously for 1 minute. The loader.io application will provide results such as response time, the number of responses, and the volume of data processed. After collecting the test data, a comparison is made between a single server and a distributed server.

Table 1
Server Spesification

	CPU (core)	**Memory (GB)**	**Disk (GB)**
Single Server	4	2	32
Distributed Server	1	2	32

Table 1 illustrates that a single server is equipped with a VM containing 4 CPU cores, 2 GB of RAM, and 2 GB of disk space. In contrast, the distributed server setup involves 4 VMs, each with 1 CPU core, 2 GB of RAM, and 32 GB of disk space. The distributed servers use fewer cores per VM (1 core each) to maintain a consistent total of 4 cores

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 3, July 2024
https://doi.org/10.47709/cnapc.v6i3.4383

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

across the single server and the distributed server setup, facilitating a fair comparison of CPU usage between the two configurations.

In this web server test, one virtual machine (VM) is used to represent a single web server, while five VMs are used for a distributed web server setup. In the distributed configuration, one VM functions as a load balancer, while the remaining four VMs act as web servers.

4. RESULT

In this section, the researcher will explain the results of the research obtained. Researchers can also use images, tables, and curves to explain the results of the study. These results should present the raw data or the results after applying the techniques outlined in the methods section. The results are simply results; they do not conclude.

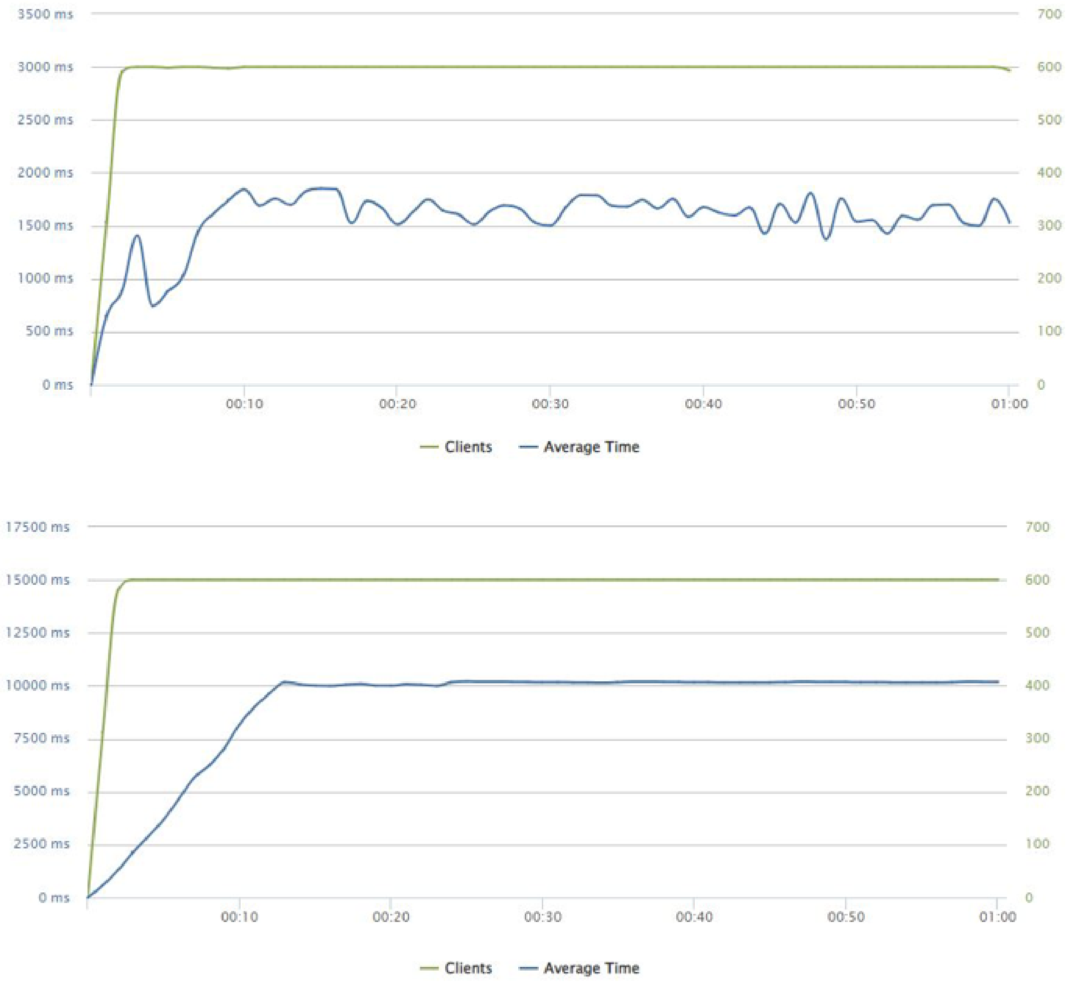


Fig. 2 Response Time Measurement Result Graph

"Figure 2 illustrates the response time measurements for accessing a web server. The upper graph shows the results for a web server using a distributed setup, while the lower graph shows the results for a single-server setup. The data reveals that the response time with a distributed server was variable, fluctuating between 1500ms and 2000ms.

* Corresponding author

Conversely, the single-server setup yielded a consistent response time of around 10,000ms. This suggests that websites utilizing a distributed server can handle requests more quickly compared to those using a single server."

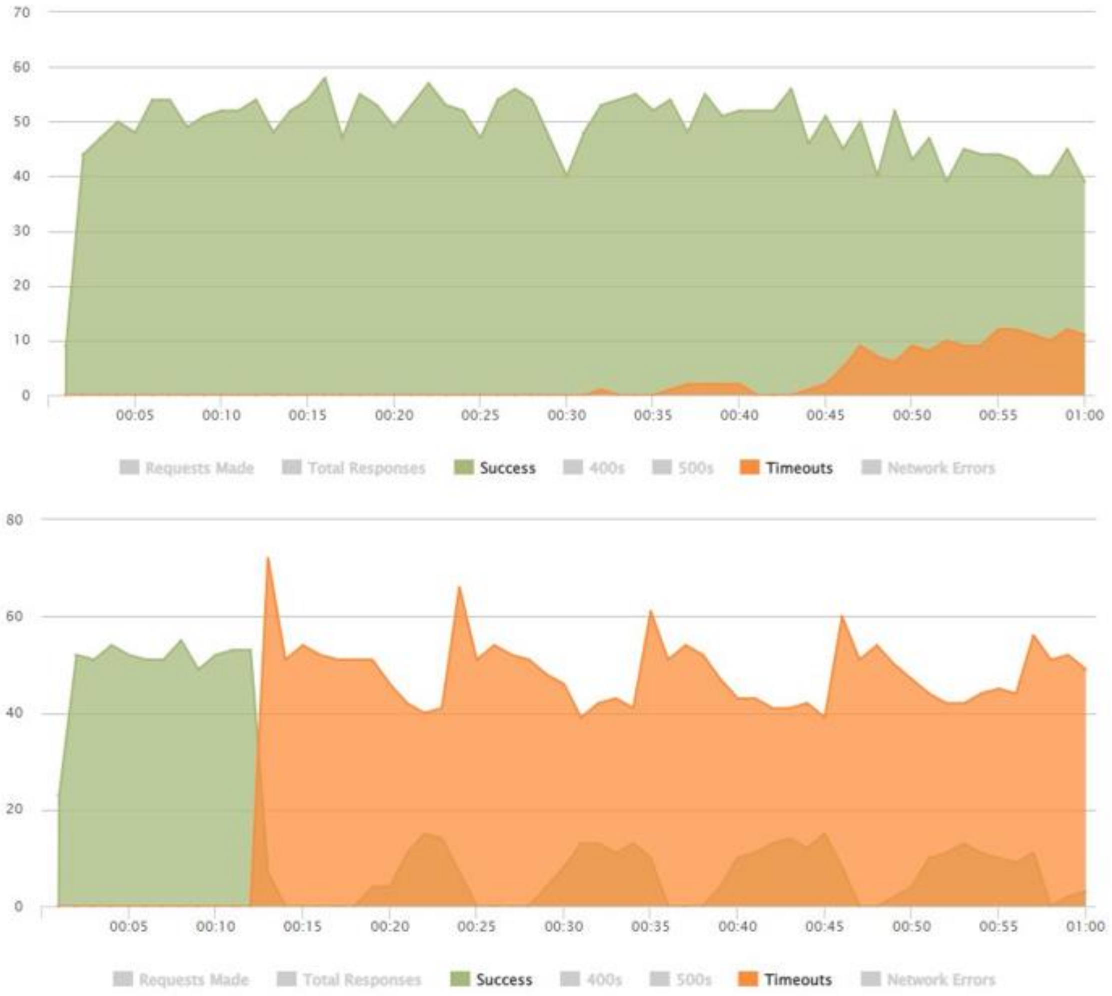


Fig. 3 Response Counts Measurement Result Graph

Figure 3 illustrates the results of measuring the number of responses when accessing a website. The top graph displays measurements taken from a web server using a distributed server, while the bottom graph shows measurements from a single server. The top graph reveals that the average number of successful responses is relatively consistent each second. In contrast, the bottom graph shows a sharp decline in successful responses starting at the 15th second, and they never reach the initial peak again. Similarly, the top graph indicates a minimal number of timeouts beginning at the 30th second, whereas the bottom graph shows a high number of timeouts starting at the 12th second and continuing throughout. This demonstrates that websites using distributed servers handle requests more consistently compared to those using single servers. This stability is due to the fact that requests on a distributed server are spread and processed by multiple servers. In comparison, a single server handles all requests alone, leading to errors and timeouts when the server's resources are overwhelmed.

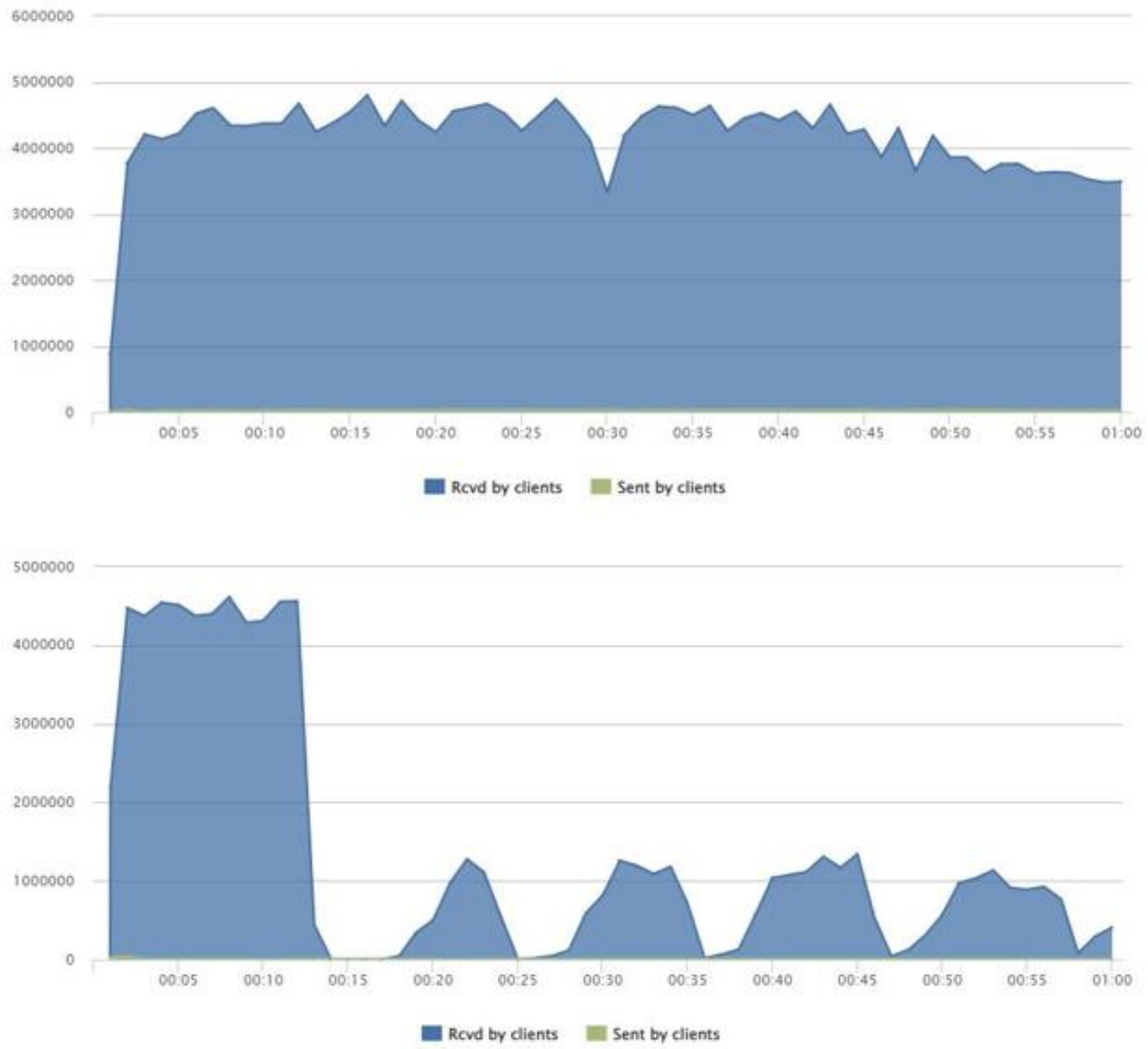


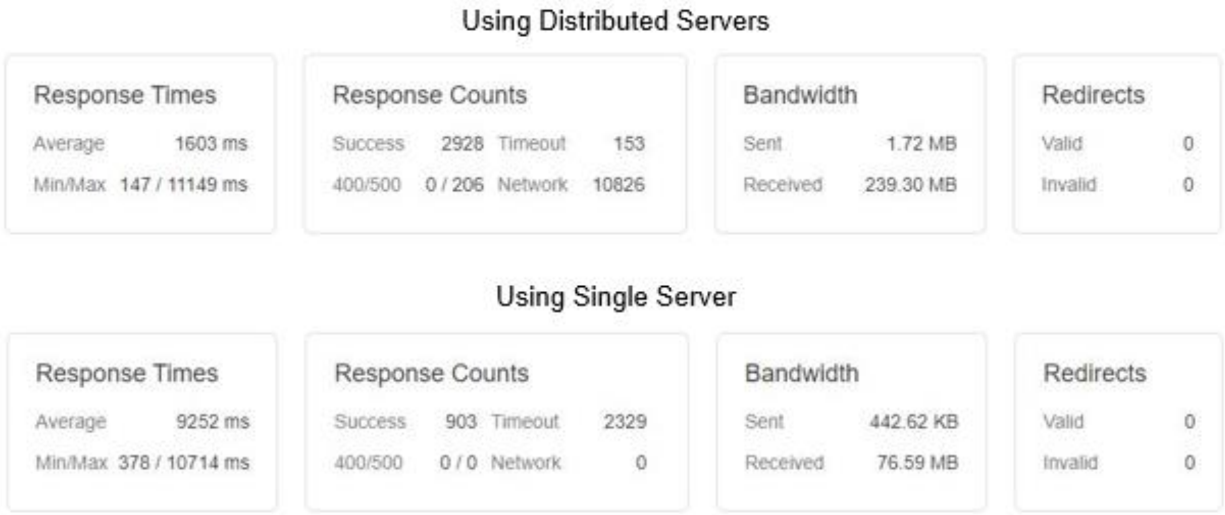
Fig. 4 Data Transfer Measurement Result Graph

Figure 4 illustrates the measurement results of the amount of data sent and received by the test server. The upper graph shows data for a web server using a distributed setup, while the lower graph shows data for a single server. The data transfer measurements correspond to the earlier graph in Figure 3, which depicts the number of responses the web server successfully handled. The data shows that the distributed server maintains a more stable transfer rate throughout the test period. In contrast, the single server, as shown in the lower graph, experiences a significant decline in data transfer capacity at various intervals, indicating that a single server processes much less data.

* Corresponding author

Journal of Computer Networks, Architecture and High Performance Computing
Volume 6, Number 3, July 2024
https://doi.org/10.47709/cnapc.v6i3.4383

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024



Using Distributed Servers

Response Times
Average 1603 ms
Min/Max 147 / 11149 ms

Response Counts
Success 2928 Timeout 153
400/500 0 / 206 Network 10826

Bandwidth
Sent 1.72 MB
Received 239.30 MB

Redirects
Valid 0
Invalid 0

Using Single Server

Response Times
Average 9252 ms
Min/Max 378 / 10714 ms

Response Counts
Success 903 Timeout 2329
400/500 0 / 0 Network 0

Bandwidth
Sent 442.62 KB
Received 76.59 MB

Redirects
Valid 0
Invalid 0

Fig. 5 Comprehensive Results of The Website Testing

Figure 5 presents the comprehensive results of the website testing, covering response time, number of responses, and data transfer. The comparison between the two sets of test results is made by comparing them against each other. Equation (1) is used to calculate the comparison values from these test results.

$$Comparison\ Value = \frac{Highest\ Value}{Lewest\ Value} \quad (1)$$

For the comparison value of response time, the following results are obtained:

$$Response\ Times = \frac{Highest\ Value}{Lewest\ Value} \quad (2)$$

$$Response\ Times = \frac{9252}{1603}$$

$$Response\ Times = 5{,}8$$

Using the same equation, the calculation results for the number of responses and data transfer are obtained. Table 1 presents the comparison of all test results.

Table 2
Comparison Results

	Distributed Servers	**Single Servers**	**Comparisons**
Response Times (ms)	1603	9253	5,8
Success Response	2928	903	2,2
Time Out Response	153	2329	14,2
Data Transferred (MB)	239,3	76,59	2,1

According to Table 2, the distributed server has a response time that is 5.8 times faster than the single server. Furthermore, it achieves 2.2 times more successful responses and can transfer 2.1 times more data than the single

* Corresponding author

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

server. The single server experiences a considerably higher timeout rate, with 14.2 times more timeouts compared to the distributed server. These results clearly show that a website using a distributed server performs much better than one using a single server.

DISCUSSIONS

After evaluating and comparing the performance of websites using single versus distributed servers, it was found that websites with distributed servers perform better. This improvement is due to the fact that traffic requests on a distributed web server are handled by multiple servers. Even though the total number of CPUs used is equivalent to that of a single server, the distributed setup processes data more quickly and efficiently. To gain more precise insights, further research is needed on how traffic is distributed among servers and to analyze the queuing processes on each server. Additionally, comparing different load balancing methods could help identify the most effective approach for managing web server requests.

5. CONCLUSION

Distributed servers using virtualization have been successfully tested and show impressive performance improvements. In this study, the first author is in charge of planning, designing the topology, and implementing the system. Meanwhile, the second author handles data analysis and processing during testing.The results reveal that the distributed server responds 5.8 times faster than a single server, handles 2.2 times more successful responses, and transfers 2.1 times more data. Additionally, the single server experiences a significantly higher timeout rate, with 14.2 times more timeouts compared to the distributed server. These findings clearly demonstrate that websites with distributed servers offer substantially better performance than those using a single server. In future research, traffic analysis can be conducted to understand how the request queue process functions on each server. Additionally, comparing different load balancing methods could help identify the most effective approach for managing web server requests.

6. REFERENCES

Abdulkarim, Abdulrahman, Ishaq Muhammed, Lele Mohammed, and Abbas Babayaro. 2019. “Performance Analysis of an Improved Load Balancing Algorithm in Cloud Computing.” American Journal of Networks and Communications 8(2).

Alankar, B., Sharma, G., Kaur, H., Valverde, R., & Chang, V. (2020). Experimental setup for investigating the efficient load balancing algorithms on virtual cloud. Sensors, 20(24), 7342.

Arfeen, Asad, Muhammad Asim Khan, Obad Zafar, and Usama Ahsan. 2022. “Process Based Volatile Memory Forensics for Ransomware Detection.” Concurrency and Computation: Practice and Experience 34(4).

Cuzzocrea, Alfredo, Enzo Mumolo, and Pietro Corona. 2015. “Cloud-Based Machine Learning Tools for Enhanced Big Data Applications.” In Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015,.

Du, Guangyu, Hong He, and Qinggang Meng. 2014. “Energy-Efficient Scheduling for Tasks with Deadline in Virtualized Environments.” Mathematical Problems in Engineering 2014.

Deshmukh, G. (2023). Performance comparison of different load balancing algorithms in cloud computing. International Journal on Recent and Innovation Trends in Computing and Communication, 11(8s), 747-754.

Ekre, A.A., N.M. Nimbarte, and S.V. Balamwar. 2018. “An Empirical Proposition to Load Balancing Effectuate on AWS Hybrid Cloud.” International Journal of Scientific Research in Computer Science and Engineering 6(4).

Eludiora, Safiriyu et al. 2010. “A Load Balancing Policy for Distributed Web Service.” International Journal of Communications, Network and System Sciences 03(08).

Ma'arifah, W. (2024). Performance comparison of zevenet multi service load balancing with least connection and round robin algorithm. Joiv International Journal on Informatics Visualization, 8(2), 882.

Motohashi, H., Nguyen, P., Nguyen, K., & Sekiya, H. (2022). Implementation of p4-based schedulers for multipath communication. Ieee Access, 10, 76537-76546.

Ramadhan, H. and Wardhana, S. (2021). Computer networks optimization using load balancing algorithms on the citrix adc virtual server. Jurnal Online Informatika, 6(1), 103.

* Corresponding author

Submitted : 24 July 2024
Accepted : 28 July 2024
Published : 31 July 2024

Rino Subekti et al. 2024. TRANSFORMASI DIGITAL (Teori & Implementasi Menuju Era Society 5.0). Jakarta: PT. Sonpedia Publishing Indonesia. https://www.researchgate.net/publication/380462238_TRANSFORMASI_DIGITAL_Teori_implementasi_Menuju_Era_Society_50.

Satwika, I Kadek Susila, I Putu Susila Handika, and Made Hanindia Prami Swari. 2022. "ANALISIS UTILISASI RESOURCE CLUSTERS PADA HADOOP MENGGUNAKAN VIRTUALIZATION." Jurnal RESISTOR (Rekayasa Sistem Komputer) 5(1).

Setiawan, Roisul, Bayu Rahayudi, Dany Primanita, and Kartika Sari. 2020. "Implementasi High Availability Web Server Menggunakan Metode Web Cluster Dan High Availability Proxy Pada Single Board Computer (Studi Kasus: UD Sendang Putra)." Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya 4(9): 2548–2964. https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7880.

Song, He, Xiaofeng Wang, Mengdong Zhai, and Guangjie Zhang. 2020. "High-Fidelity Router Emulation Technologies Based on Multi-Scale Virtualization." Information (Switzerland) 11(1).

Swari, Made Hanindia Prami, I Kadek Susila Satwika, and I Putu Susila Handika. 2020. "Performance Analysis of Sales Big Data Processing Using Hadoop and Hive in Cloud Environment." In 2020 6th Information Technology International Seminar (ITIS), IEEE, 162–66. https://ieeexplore.ieee.org/document/9320964/.

Wunarso, Novita Belinda, Andrias Rusli, Michelle Angelica, and Arabella Margaret Salim. 2013. "Implementasi Distributed File Server Pada Apache Web Server." Jurnal ULTIMA InfoSys 4(2).