2024 3rd International Conference on Creative Communication and Innovative Technology (ICCIT) | 979-8-3503-6749-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCIT62134.2024.10701232

Performance Evaluation of Monolithic Architectural Design in Online Ornamental Plant Sales Platform

Daniel Yeri Kristiyanto
dept. of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025232003@student.its.ac.id

Riyanarto Sarno
dept. of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@its.ac.id

Kelly Rossa Sungkono
dept. of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelly@its.ac.id

Rini Handayani
dept. of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025232001@student.its.ac.id

Andreas Schmidt
dept. of Engineering
Eduaward Incorporation
Lancaster, United Kingdom
schmidt.andre@eduaward.co.uk

***Abstract*—System architecture design is critical to system construction, with monolithic architecture offering a simple and centralized development pattern. This research examines its application to an online platform selling ornamental plants, integrating several menus into one integrated system. Performance was evaluated through load testing on two types server, namely Linux and Windows using Grafana K6 tools. Quantitative assessment using AHP and MOORA, considering criteria, HTTP failure (HF), P95 response time (RT), peak RPS (PR), and requests made (RM). The results show significant differences between testing on Linux and Windows. Test results on Windows server show reliability of around 40%, while Linux server test results achieve access effectiveness of 90% with an average response time of 2654 ms. Testing on Windows server recorded an average connection time of 1.78 ms and an HTTP failure rate of 2.16%. Testing on Linux server shows better performance with fewer HTTP failures and improved scalability. The complexity of monolithic architectures can cause bottlenecks and maintenance challenges in large-scale systems. These findings provide insight into the reliability and effectiveness of websites, offering implications for system architecture design and performance optimization in similar contexts.**

***Keywords*—Monolith, MCDM, System Evaluation**

I. Introduction

The current era of digitalization has shown significant progress in both design and utilization. Information systems have received significant focus on increasing their performance levels, recognizing the importance of the need for information across all user groups, from developers to the wider community of service users [1]–[4]. In the process of innovation and system development, compliance with established guidelines is very important, including choosing the right framework or application system architecture [5]–[9]. Monolithic architectural design consolidates all system components into one integrated whole, emphasizing centralization in its development path [10], [11]. Integration is key in this approach because the database, logic, and interface design combine seamlessly [12]. Monolithic architectural design aims to simplify usability and ensure consistency in system changes without complex service integration considerations [7], [13]–[16]. This research focuses on the official Sipentas (Ornamental Plant Farmer System) website which adopts a dynamic and monolithic architectural design. As a forum for optimizing sales of ornamental plants for associations such as Sekarsari and Magersari in Magelang City, Central Java, this website has various functions that facilitate sales and transactions between sellers and consumers. The main goal is to expand business promotions by accessing a wider consumer base via the internet. Although the website system requires further development for optimal functionality, its effectiveness needs to be tested through load testing using the Grafana K6 tool, which includes customizable Virtual Users [17], [18]. This tool is widely used to test the performance of websites, APIs and databases and will evaluate the effectiveness and performance of the system. Testing was carried out on two types of server, Suse Linux Enterprise Server (SLES) 9 and Windows server 2003. This testing focuses on assessing system performance and identifying initial problems, which serve as parameters for future development [19]. Apart from that, this test also aims to measure the system's resilience and efficiency in carrying out user commands. The results of this test can provide insight into the performance of the Sipentas website. The results of the two tests can be compared quantitatively to determine the quality of each system [20]. Quantitative analysis uses Multi-Criteria Decision Making (MCDM) with the Analytical Hierarchy Process (AHP) method which is used to identify criteria and carry out consistency testing, while determining the best effectiveness alternative is carried out through quantitative calculations using Multi Objective, Optimization Method Based on Ratio Analysis (MOORA) [21]–[23]. Analytical Hierarchy Process and Multi-Objective Optimization by Ratio Analysis are decision support system methods [24]. This research aims to establish a foundation and benchmark for future system development, with test results outlining the system's challenges and shortcomings in the given task. This research provides a unique contribution compared to previous studies by specifically evaluating the performance of a monolithic architectural design in an e-commerce context using both Linux and Windows server environments [25]. Unlike earlier research that often focuses on microservices architecture or general performance testing, this study delves into the practical implications of server environment selection for monolithic systems. By employing both the Analytical Hierarchy Process (AHP) and Multi-Objective Optimization on the Basis of Ratio Analysis (MOORA) methods, this research offers a comprehensive multi-criteria decision-making approach to performance evaluation. This dual-method analysis not only highlights the strengths and weaknesses of each server environment but also provides a robust framework for future studies aiming to optimize system performance based on specific criteria. Additionally, the focus on a real-world application—the Sipentas e-commerce platform—adds practical value and relevance, demonstrating how the findings can be directly applied to enhance system performance and scalability in similar contexts.

II. Methodology

This study employs a direct observation approach through performance testing of the Sipentas website system. Quantitative analysis is conducted using the Multi-Criteria Decision Making (MCDM) method to assess the accuracy of the testing data and explain the effectiveness of the system on two types of servers. In this study, AHP is used to identify criteria such as HTTP failures (HF), P95 response time (RT), peak requests per second (PR), and requests made (RM). These criteria are assigned weights based on their importance, which are used in consistency testing to ensure reliable results. MOORA is used to determine the best alternative by evaluating the criteria weights identified in the AHP method. The final ranking results indicate the most effective server setup based on the normalized values of the criteria. Figure 1 illustrates the research flow, detailing the steps of data collection, system execution, and quantitative analysis using AHP and MOORA methods.

Data collection involves several procedures, including test setup, execution, Multi-Criteria Decision Making (MCDM) calculations, consistency testing, Multi-Objective Optimization on The Basis of Ratio Analysis (MOORA) evaluation, and prioritization.

A. System Execution

This phase marks the initial step in preparing the website system for testing, involving a thorough inspection of the entire web system structure. Each element in the website system undergoes manual inspection to detect and rectify potential errors before testing.

B. Preparation For Load Testing

The testing conducted on the Sipentas website system involves load testing, which aims to identify the system's.

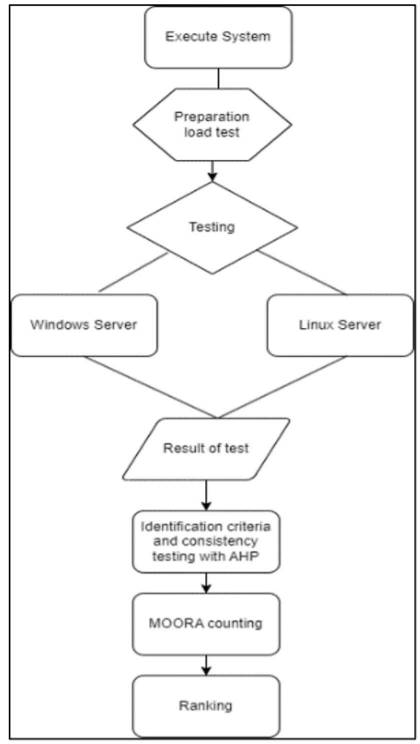

Fig. 1. Compilation of quiz players utilizing the Wordwall platform.

```
import http from 'k6/http';
import { sleep } from 'k6';

export const options = {
  vus: 10,
  duration: '30s',
};

export default function() {
```

Fig. 2. Syntax Testing.

performance and reliability levels in handling access loads performed by Virtual Users (VUs). Below are the test script formats.

C. Testing

Testing was carried out on two server groups, Linux and Windows. Testing was carried out using the HTTP address of the sipentas.com website. The import syntax in figure 2. functions as load testing code that facilitates HTTP requests from the k6 library, and the import command provides a pause with the 'sleep' function. Export options set test parameters, such as the number of Virtual Users (VUs) and access duration. Testing can be started with the command k6 run –vus (number of VUs) –duration (access time) script.js, directing script.js to run the test according to the conditions and duration of the specified VUs. The default export function, along with the 'get' function, is executed by k6 to access each server address.

D. Test Result

Test The test outcomes will be categorized into two types based on the conducted tests within each test group. The average results vary in quantity across different test categories, encompassing both durability and performance aspects.

E. MCDM Counting

Based on the test results obtained, criteria and weights can be determined to be used in the Multi-Criteria Decision Making (MCDM) process using the Analytical Hierarchy Process (AHP). To determine the best alternative, this is done through the MOORA calculation results.

1) Identifying Criteria and AHP Weighting
   a) Identify the criteria and level of importance relevant to the testing topic with AHP.

TABLE I
TESTING CRITERIA

C	Information
C1	(HF) HTTP Failures
C2	(RT) P95 Response Time
C3	(PR) Peak RPS
C4	(RM) Request Made

   b) Identify options for conducting tests.

TABLE II
TESTING ALTERNATIVES

Alternatives
SLES 9 Linux testing
Windows server 2003 testing

   c) Creating a hierarchy to map the relationship between criteria and alternatives.

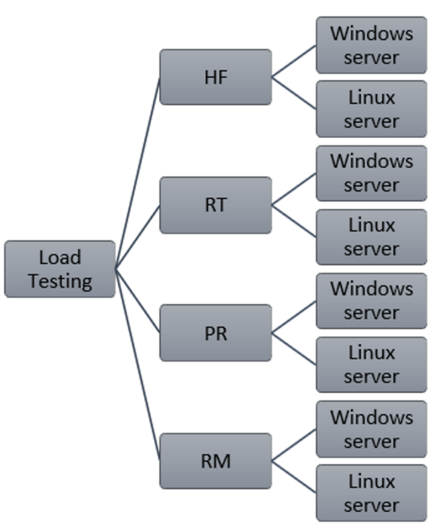

Fig. 3. Hierarchy of Alternatives and Criteria

   d) Provide a relative assessment for each criteria weight.

TABLE III
CRITERIA WEIGHTING

C	Information	Levels
C1	(HF) HTTP Failures	5 (More important)
C2	(RT) P95 Response Time	3 (Quite important)
C3	(PR) Peak RPS	3 (Quite important)
C4	(RM) Request Made	3(Quite important)

F. Consistency Testing

$$\begin{bmatrix} HF \\ RT \\ PR \\ RM \end{bmatrix} \begin{bmatrix} HF & RT & PR & RM \\ a & b & c & d \\ a & b & c & d \\ a & b & c & d \end{bmatrix}$$

$$t = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\text{to element-}i \text{ on } (A)(W^T)}{\text{to element-}i \text{ on } W^T} \right) \tag{1}$$

$$CI = \frac{CI}{RI_n} \tag{2}$$

Consistency testing is done by calculating the overall value of the matrix multiplication divided by the weight of each criterion C obtained through pairwise comparisons. The symbol 'n' in the formula represents the number of alternatives available in the calculation process. The consistency test results should show an index ≤ 0.1 or = 0 for the calculation results to be considered consistent or sufficiently consistent.

G. MOORA Counting

The MOORA method calculation is used to analyze alternatives with superior value levels based on the fulfillment and weighting of criteria identified in the AHP method. To perform calculations using the MOORA method, the following formula is used.

$$\mathbf{X}_{ij} = \frac{X_{ij}}{\sqrt{\sum_{j=1}^{n} X_{ij}^2}} \tag{3}$$

Formula (3) is used to calculate the normalization of values in the decision matrix with the aim of eliminating existing scale differences.

$$y_{ij} = \sum_{j=1}^{g} y_{ij} - \sum_{j=g+1}^{n} y_{ij} \tag{4}$$

Formula (4) can be used to calculate the normalized result by subtracting values on benefit criteria from criteria with cost values.

H. Ranking

Ranking is done based on the results of MCDM calculations using the MOORA method through the identification of criteria and consistency testing in the AHP method. The ranking results will indicate alternatives with superior effectiveness values

III. Result and Discussion

A. Load Testing Result

The results of load testing carried out on the Sipentas website system show quite different results between the two test classes. The following is the conclusion of the load testing results for the two test classes.

1) Windows Testing Testing using Windows server 2003 has been carried out three times involving around 80 virtual users to access the website. Virtual users access the system in stages, the results of this test are in table V. The following is a graph of test results that shows the level of system strength on Windows server in fulfilling access requests. According to the test outcomes conducted on

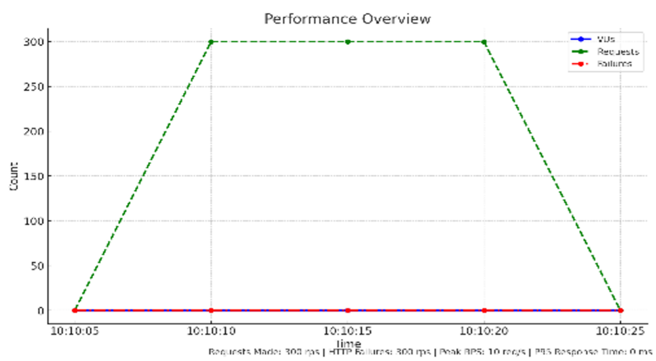

Fig. 4. Windows Server Test Graph

various virtual users with differing access time conditions, the results depicted in the graph above were obtained. The graph indicates that the potential access failure rate could reach up to 300 failed requests. Based on the graph

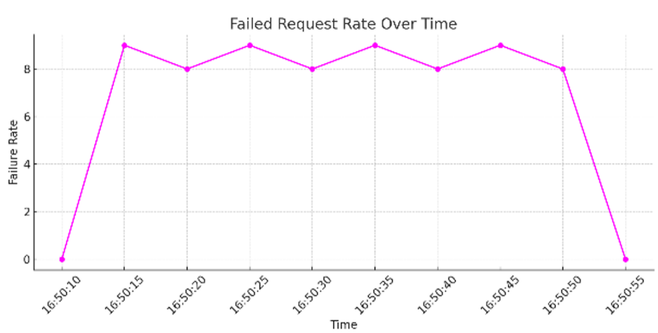

Fig. 5. Request Failure Rate Graph

above, it is clear that the test results on Windows server have a large risk of access failure, because the number of requests is aligned with the number of access failures. Therefore, it can be concluded that access using Windows server to the Sipentas website shows several deficiencies that require improvement.

2) Linux Testing Website Load testing on SLES 9 Linux server also uses 80 virtual users with the same access conditions as testing on Windows server. Based on the results of the tests that have been carried out, load testing data was obtained using 10 VUS with an access time of 30 seconds. The results obtained show that the web server can be reached by 10 VUS users without interruption with 210 iterations or 6.526727/second. The following is a table of test results for the Sipentas web system using Linux server.

TABLE IV
LINUX SERVER TEST RESULTS

Tested	Vus 10	Vus 20	Vus 50
{expected response : true}	Avg = 3.0sms Min = 63.49ms Med = 102.82ms Max = 4.61s P(90) = 1.13s P(95) = 2.11s	Avg = 2.14ms Min = 72.94ms Med = 103.61ms Max = 5.73s P(90) = 3.59s P(95) = 5.1s	Avg = 2.47s Min = 74.78 Med = 1.68s Max = 6.26s P(90) = 5.18s P(95) = 5.31s
Http req failed	0.00% (x110)	0.00% (x 200)	0.00% (x 216)
Http reqs	210 (2.526727/s)	137 (2.048502/s)	871 (3.13326/s)
Vus	1 Min = 1 Max = 10	7 Min = 7 Max = 20	1 Min = 1 Max = 50
Vus max	10 Min = 10 Max = 10	20 Min = 20 Max = 20	50 Min = 50 Max = 50

TABLE V
WINDOWS SERVER TEST RESULTS

Tested	Vus 10	Vus 20	Vus 50
{expected response : true}	Avg = 2.08s Min = 1.61s Med = 1.77s Max = 6.19s P(90) = 2.86s P(95) = 4.89s	Avg = 3.71s Min = 287.61ms Med = 3.88s Max = 5.07s P(90) = 3.99s P(95) = 4.05s	Avg = 13.12s Min = 11.06s Med = 11.23s Max = 26.18s P(90) = 20.37s P(95) = 23.38s
Http req failed	0.00% (x137)	0.00% (x 234)	0.00% (x234)
Http reqs	101 (3.111563/s)	137 (3.048502/s)	234 (3.24666/s)
Vus	2 Min = 2 Max = 10	3 Min = 3 Max = 20	1 Min = 1 Max = 50
Vus max	10 Min = 10 Max = 10	20 Min = 20 Max = 20	50 Min = 50 Max = 50

Testing on 20 Vus with an access time of 30 seconds has almost the same results as testing on the previous 10

Vus, only the request access speed tends to be longer than with 10 Vus. There were also more iterations, reaching 316 iteration processes.

Meanwhile, during the 50 Virtual Users (Vus) test, there was an increase in the volume of received data, access time, and the number of iterations. This suggests that as the load increases, the system's response tends to adapt accordingly.

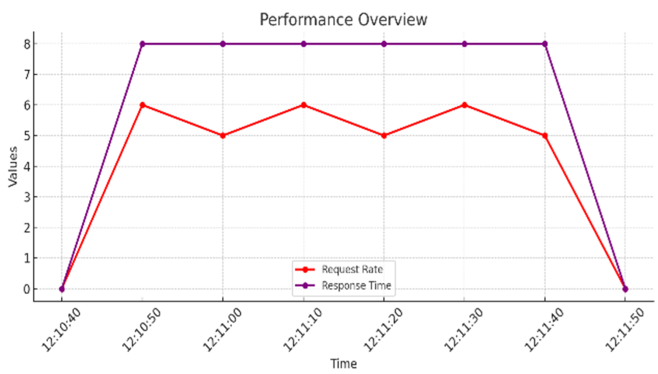

Fig. 6. Linux Server Test Graph

The graph above illustrates the server's performance in handling user requests for accessing the Linux server. It received a total of 148 requests, with an average response time of 2,654 ms. Analysis of the graph indicates that as the number of requests increases, the time required to fulfill them also increases. Additionally, the test results demonstrate a minimal failure rate for this server.

B. AHP Calculation Result

AHP method in testing will be used as a tool for identifying criteria and testing the consistency of the criteria against the test results. The following data summarizes the test results involving four criteria, namely HTTP Failures (HF), Response Time (RT), Peak RPS (PR), and Requests Made (RM).

TABLE VI
CONCLUSION OF THE RESULT

Alternatives	C (Criteria)			
	HF	RT	PR	RM
WINDOWS	290 reqs	630ms	10 reqs/s	300 reqs
LINUX	0 reqs	253ms	12 reqs/s	348 reqs

$$\begin{bmatrix} 1 & 5 & 3 & 3 \\ 1/5 & 1 & 1/3 & 1/3 \\ 1/3 & 3 & 1 & 1 \\ 1/3 & 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 5 & 3 & 3 \\ 0.2 & 1 & 0.33 & 0.33 \\ 0.33 & 3 & 1 & 1 \\ 0.33 & 3 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1/1.86 & 5/12 & 3/5.33 & 3/5.33 \\ 0.2/1.86 & 1/12 & 0.33/5.33 & 0.33/5.33 \\ 0.33/1.86 & 3/12 & 1/5.33 & 1/5.33 \\ 0.33/1.86 & 3/12 & 1/5.33 & 1/5.33 \end{bmatrix}$$

$$\begin{bmatrix} 0.5376 & 0.4166 & 0.5628 & 0.5628 \\ 0.1075 & 0.0833 & 0.0619 & 0.0619 \\ 0.1774 & 0.25 & 0.1876 & 0.1876 \\ 0.1774 & 0.25 & 0.1876 & 0.1876 \end{bmatrix} \begin{bmatrix} 0.5199 \\ 0.0786 \\ 0.2006 \\ 0.2006 \end{bmatrix}$$

$$[W = 0.5199; 0.0786; 0.2006; 0.2006]$$

$$t = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{\text{to element-}i \text{ on } (A)(W^T)}{\text{to element-}i \text{ on } W^T} \right)$$

$$t = \frac{1}{4}(4,0709 + 4,0063 + 4,0304 + 4,0304)$$
$$= \frac{16,138}{4} = 4,0345$$

$$\frac{CI}{RI_4} = \frac{0,0345}{0,90} = 0,0115 \leq 0,1 \text{ (Quite consistent)}$$

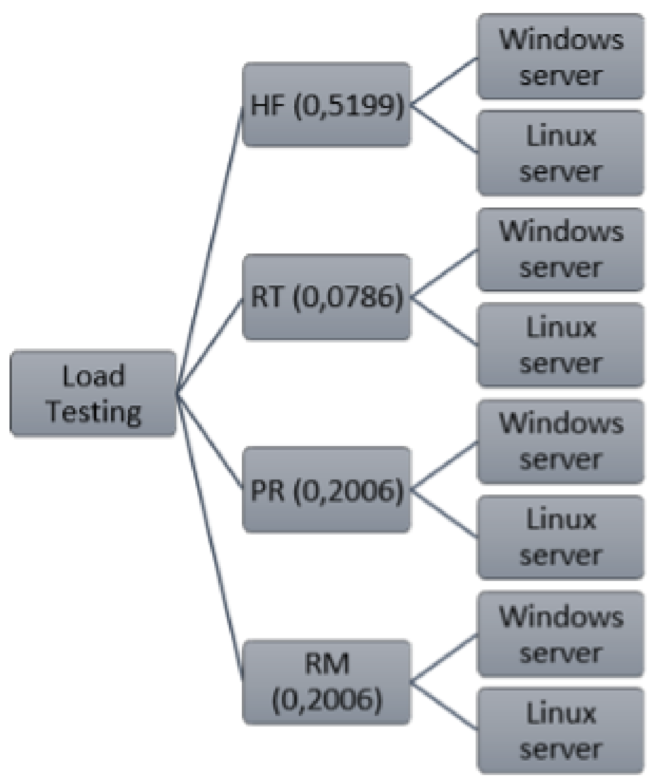

Fig. 7. Criteria Weighting Hierarchy

The criteria and consistency testing using the AHP method showed that the results obtained are quite stable, meaning that the relative comparison between elements will not undergo significant changes from one test to another. Therefore, the calculation can be continued using the MOORA method to determine the test results with the best quality level.

TABLE VII
MOORA CRITERIA

Alternatif	Kriteria	Bobot	Kategori
Windows Linux	HF	30	Cost
	RT	25	Cost
	PR	20	Cost
	RM	25	Benefit

TABLE VIII
MOORA WEIGHTING

Alternatif	HF	RT	PR	RM
Windows	1	1	2	1
Linux	1	2	1	2

Table VII. presents the criteria data along with their categories and weights, with a maximum total weight of 100. This weighting is a representation of the AHP method's weighting results. Table VIII presents the mapping results of the criteria values for each alternative, viewed on a scale of 1, indicating a lower value ratio for lower qualifications, and 2, indicating a higher value ratio for higher qualifications.

$$X_{11} = \frac{1}{\sqrt{1^2+1^2}} = 1$$
$$X_{12} = \frac{1}{\sqrt{1^2+2^2}} = 0,4472$$
$$X_{13} = \frac{2}{\sqrt{2^2+1^2}} = 0,8944$$
$$X_{14} = \frac{1}{\sqrt{1^2+2^2}} = 0,4472$$
$$X_{21} = \frac{1}{\sqrt{1^2+1^2}} = 1$$
$$X_{22} = \frac{2}{\sqrt{1^2+2^2}} = 0,8944$$
$$X_{23} = \frac{1}{\sqrt{2^2+1^2}} = 0,4472$$
$$X_{24} = \frac{2}{\sqrt{1^2+2^2}} = 0,8944$$

The calculation is one of the steps in the MOORA method, which aims to determine the optimization value for each criterion based on the weighting value of each alternative. Based on the optimization value calculation results, the normalized matrix is obtained as follows.

$$\begin{bmatrix} 1 & 0,4472 & 0,8944 & 0,4472 \\ 1 & 0,8944 & 0,4472 & 0,8944 \end{bmatrix}$$

Normalized matrix results can be used to proceed to the next step, which involves calculating the criteria values in each category. It is known that in this calculation, there are two categories: three cost criteria and one benefit criterion. To calculate the final result of this method, the value of the benefit criterion can be subtracted from the value of the cost criteria.

$$\begin{aligned} YA1 = (0,4472 \times 0,25) - ((1 \times 0,30) + (0,4472 \times 0,25) + \\ (0,8944 \times 0,20)) = -0,4788 \\ YA2 = (0,8944 \times 0,25) - ((1 \times 0,30) + (0,8944 \times 0,25) + \\ (0,4472 \times 0,20)) = -0,3894 \end{aligned}$$

The calculation results using the MOORA method show that the test results with Linux server have a higher index value compared to Windows server. Test results from both areas produce varied outputs, which serve as indicators for developers and users regarding the system's capability to handle requests. Additionally, the MCDM AHP and MOORA method's calculation results reveal that user access quality tends to be more competent through Sipentas web server access using Linux server.

TABLE IX
COMPARISON OF TEST RESULTS

Component	Windows Test	Linux Test
Percentage of failure	Testing use Windows server shows a fairly high access failure rate, as seen in Figure 5. This can be caused by several factors, one of which is issues with the internal hardware being used.	Load testing results on the HTTP failure component showing the potential for failure <HF in Windows testing.
Request rate	The access speed for Windows server requests tends to be higher, at 10 requests per second.	Meanwhile, request access speed tends to be faster and can accommodate 12 requests per second.
Response time	The response time provided by the Windows server is slower with 630ms/reqs.	The time required to respond to web access tends to be faster, reaching 253ms.
Max request	The maximum number with Windows server testing, requests reaches 300 requests, but with a total of 90/100 failures.	Meanwhile, testing on Linux, the maximum number of requests reached 348 requests, this can be influenced by system performance and the device used.

C. Monolith Architectural Design Analysis

The development of the Sipentas website system (Ornamental Plant Farmer System) employs a monolithic architectural design. This design integrates all existing menu features without distributed services. Each menu feature is consolidated into a unified system, facilitating changes in response to system modifications. Additionally, the database utilized is singular, eliminating the need for separate databases for individual services. While the monolithic architectural design is straightforward, it may be deemed less flexible when applied to large-scale and intricate systems. The findings of this study have important implications for future architectural decisions and real-world applications. The superior performance of the Linux server, demonstrated through lower HTTP failure rates, faster response times, higher peak requests per second, and greater total requests made, suggests that Linux-based environments are more suitable for applications requiring high reliability and scalability. For e-commerce platforms like Sipentas, where user experience and transaction reliability are critical, adopting a Linux server can significantly enhance system performance and customer satisfaction. Furthermore, the cost-efficiency and reduced maintenance requirements associated with Linux servers can lead to substantial operational savings, making them a more economical choice in the long run. These insights should guide future architectural decisions, emphasizing the selection of server environments that align with the performance and scalability needs of the application. By leveraging these findings, businesses can optimize their back-end infrastructure to ensure seamless, efficient, and reliable service delivery. In practice, these results can be leveraged by system architects and IT managers to make informed decisions about server environments for monolithic architectures. For instance, e-commerce platforms and other high-traffic websites can adopt Linux servers to improve reliability and performance, ensuring a better user experience and higher transaction success rates. Moreover, the cost-efficiency and reduced maintenance needs associated with Linux servers can lead to substantial savings, which is particularly beneficial for startups and small businesses with limited budgets. The comprehensive evaluation framework presented in this study can also serve as a benchmark for future performance assessments, enabling organizations to systematically compare different server environments and optimize their IT infrastructure accordingly. By applying these findings, businesses can achieve more scalable, reliable, and cost-effective system architectures, ultimately supporting their growth and operational efficiency

IV. Conclusion

The Sipentas website adopts a monolithic architectural approach, integrating the user interface, database, and functionality into a unified structure where all menu features interact seamlessly. Load testing carried out on the Sipentas website system using Grafana 6 evaluates web performance in the two operating systems applied. The test results show differences, testing on Windows server is considered less effective. The virtual user is having difficulty accessing the HTTP server address, identified by an access failure. In contrast, testing on Linux server shows relatively optimal performance, successfully handling the total number of access requests via HTTP. Differences in test results can be caused by different system configurations and device compatibility. The calculation results of the AHP and MOORA methods show that there is a gap in the quality and effectiveness of system performance. Accessing the system via Windows server showed significant error potential and an effectiveness rate of less than 50%, while on Linux the system demonstrated near maximum test effectiveness, with a total response time of 2,654 ms for 148 requests. Quantitative calculations also provide clear results, showing that the Windows test has a consistency level of $<$ (-0.3), meaning that the value is still below the MOORA calculation results for the area area alternative. Improvements that can be applied to optimize the implementation of the website system on these two types of operating systems can be done through the process of optimizing the web server with the operating system, through IIS configuration, updates and patches.

Acknowledgment

This research was funded by Institut Teknologi Sepuluh Nopember (ITS) under Penelitian Dana Departemen Program, Penelitian Keilmuan, Penelitian Flagship, Penelitian Kolaborasi Pusat and under the Project Scheme of the Publication Writing and Intellectual Property Rights (IPR) Incentive (Penulisan Publikasi dan Hak Kekayaan Intelektual/PPHKI) Program 2024, and the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia under Beasiswa Pendidikan Indonesia (BPI), Pusat Layanan Pembiayaan Pendidikan (Puslapdik), Balai Pembiayaan Pendidikan Tinggi (BPPT), and Lembaga Pengelola Dana Pendidikan (LPDP). BPI Number: 202327091197.

References

[1] E. Berrio-Charry, J. Vergara-Vargas, and H. Umana-Acosta, "A component-based evolution model for service-based software architectures," in *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2020-October, pp. 111–115, 2020.

[2] F. Freitas, A. Ferreira, and J. Cunha, "A methodology for refactoring orm-based monolithic web applications into microservices," *J. Comput. Lang.*, vol. 75, no. February 2022, p. 101205, 2023.

[3] S. W. F. binti Ambo'Miri, B. W. Kusuma, and A. N. Halimah, "Web-based information systems and knowledge management's impact on worker performance," *International Transactions on Education Technology*, vol. 2, no. 1, pp. 11–17, 2023.

[4] A. Lansonia and M. Austin, "The role of information management in enhancing organizational resilience," *APTISI Transactions on Management*, vol. 8, no. 1, pp. 32–39, 2024.

[5] E. Volynsky, M. Mehmed, and S. Krusche, "Architect: A framework for the migration to microservices," in *Proc. - 2022 Int. Conf. Comput. Electron. Commun. Eng. iCCECE 2022*, pp. 71–76, 2022.

[6] N. Santos and A. R. Silva, "A complexity metric for microservices architecture migration," in *Proc. - IEEE 17th Int. Conf. Softw. Archit. ICSA 2020*, pp. 169–178, 2020.

[7] J. Kazanavičius and D. Mažeika, "An approach to migrate from legacy monolithic application into microservice architecture," in *2023 IEEE Open Conf. Electr. Electron. Inf. Sci. eStream 2023 - Proc.*, pp. 1–6, 2023.

[8] H. Farsi, D. Allaki, A. En-Nouaary, and M. Dahchour, "Dealing with anti-patterns when migrating from monoliths to microservices: Challenges and research directions," in *2023 Int. Conf. Cloud Comput. Artif. Intell. Technol. Appl. CloudTech 2023*, pp. 1–8, 2023.

[9] D. Apriani, R. Afrijaldi, N. Auliya, and A. A. Darmawan, "Operating system and server integration for business effectiveness," *IAIC Transactions on Sustainable Digital Innovation (ITSDI)*, vol. 5, no. 2, pp. 91–99, 2024.

[10] N. Ivanov and A. Tasheva, "A hot decomposition procedure: Operational monolith system to microservices," in *Int. Conf. Autom. Informatics, ICAI 2021 Proc.*, pp. 182–187, 2021.

[11] A. Poniszewska-Maranda, J. MacIoch, B. Borowska, and W. Maranda, "Mechanisms for transition from monolithic to distributed architecture in software development process," in *Proc. - IEEE Comput. Soc. Annu. Int. Symp. Model. Anal. Simul. Comput. Telecommun. Syst. MASCOTS*, pp. 1–8, 2021.

[12] A. Dudhat *et al.*, "Application of information technology to education in the age of the fourth industrial revolution," *International Transactions on Education Technology*, vol. 1, no. 2, pp. 131–137, 2023.

[13] D. Pianini and A. Neri, "Breaking down monoliths with microservices and devops: An industrial experience report," in *Proc. - 2021 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2021*, pp. 505–514, 2021.

[14] G. Quattrocchi, D. Cocco, S. Staffa, A. Margara, and G. Cugola, "Cromlech: Semi-automated monolith decomposition into microservices," *IEEE Trans. Serv. Comput.*, vol. PP, pp. 1–16, 2024.

[15] E. Yuan, "Architecture interoperability and repeatability with microservices: An industry perspective," in *Proc. - 2019 IEEE/ACM 2nd Int. Work. Establ. a Community-Wide Infrastruct. Archit. Softw. Eng. ECASE 2019*, pp. 26–33, 2019.

[16] N. Lutfiani, N. P. L. Santoso, R. Ahsanitaqwim, U. Rahardja, and A. R. A. Zahra, "Ai-based strategies to improve resource efficiency in urban infrastructure," *International Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 121–127, 2024.

[17] A. Godinho, J. Rosado, F. Sá, and F. Cardoso, "Method for evaluating the performance of web-based apis," in *International Conference on Smart Objects and Technologies for Social Good*, pp. 30–48, Springer, 2023.

[18] M. Dinga, I. Malavolta, L. Giamattei, A. Guerriero, and R. Pietrantuono, "An empirical evaluation of the energy and performance overhead of monitoring tools on docker-based systems," in *International Conference on Service-Oriented Computing*, pp. 181–196, Springer, 2023.

[19] E. Nathanael and W. Sejati, "Effective government management of flood discharge in drainage channels using hec-ras 6.3. 1 application," *APTISI Transactions on Management*, vol. 7, no. 3, pp. 210–220, 2023.

[20] A. Sumanri, M. Mansoer, U. A. Matin, *et al.*, "Exploring the influence of religious institutions on the implementation of technology for stunting understanding," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 6, no. 1, pp. 1–12, 2024.

[21] C. Methods and A. S. Ajrina, "Mining zone determination of natural sandy gravel using fuzzy ahp and," *Int. J. Intell. Eng. Syst.*, vol. 13, no. 5, 2020.

[22] R. P. Setyono and R. Sarno, "Comparative method of supplier selection in abc mining company," *Int. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 2, pp. 890–899, 2020.

[23] K. R. Sungkono, "Optimizing time and cost activity based on discrete event simulation with multi-objective optimization method by ratio analysis (moora)," in *2022 Int. Conf. Adv. Comput. Sci. Inf. Syst.*, pp. 53–58, 2022.

[24] R. P. Setyono and R. Sarno, "Comparative method of moora and copras based on weighting of the best worst method in supplier selection at abc mining companies in indonesia," *Int. J. Electr. Eng. Comput. Sci.*, 2023.

[25] P. A. Sunarya, U. Rahardja, S. C. Chen, Y.-M. Lic, and M. Hardini, "Deciphering digital social dynamics: A comparative study of logistic regression and random forest in predicting e-commerce customer behavior," *Journal of Applied Data Sciences*, vol. 5, no. 1, pp. 100–113, 2024.