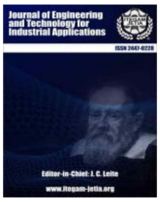
Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.9 n.44, p. 25-34. Nov/Dec, 2023

DOI: https://doi.org/10.5935/jetia.v9i44.903



RESEARCH ARTICLE OPEN ACCESS

SINGLE SERVER-SIDE AND MULTIPLE VIRTUAL SERVER-SIDE ARCHITECTURES: PERFORMANCE ANALYSIS ON PROXMOX VE FOR E-LEARNING SYSTEMS

Yuri Ariyanto*[1]

[1] Department of Information Technology, Informatics Engineering, State Polytechnic of Malang, Malang, Indonesia.

[1] http://orcid.org/0000-0001-8678-5184

Email: *yuri@polinema.ac.id

ARTICLE INFO

Article History
Received: September 09th, 2023
Revised: November 06th, 2023
Accepted: November 13th, 2023
Published: December 30th, 2023

Keywords:
Performance Test,
Virtualization,
Reverse Proxy,
Proxmox VE,
E-Learning.

ABSTRACT

At this time, almost all educational institutions use the e-learning system to support learning. The factor that plays the most role in the comfortable use of a system is the performance aspect. This type of single-server architecture is very commonly applied in building a system, but this system is actually less efficient because it does not pay attention to the level of availability. In this study, a server architecture was built in the form of multiple virtual servers utilizing virtualization on Proxmox by applying reverse proxy techniques and storage clustering to increase system availability in building an e-learning Moodle system. The design of multiple virtual server architectures uses prepare, plan, design, implementation, test and optimization methodologies. The results of the research on the testbed and test plan show that the multiple virtual server architecture has superior availability compared to the single server architecture. Based on the User Behavior Modeling Performance (UBMP) test results, the multiple virtual server architecture is also superior, with a maximum value of 100 concurrent users on the multiple virtual server-side architecture, where the availability level is 80.25%, while on the single server, it is lower with a value of 80 users and an availability level of 78.4%.

I. INTRODUCTION

An educational institution can have more than one e-learning system, which in practice generally uses a single server architecture type [1]. A single server architecture is indeed not a problem if it is used for only one system, but if several systems are active simultaneously on it it will be risky, especially when an error occurs on the server then all services will stop meaning a low level of system availability [2]. Another factor that needs to be considered is the aspect of performance and the efficient use of resources, both software and hardware so that they can support the performance and level of system availability [3].

To overcome the problems that exist in the single server architecture, multiple virtual server architecture types are used combined with the application of reverse proxy techniques for efficiency in terms of application software on the Proxmox server [4][5]. The reverse proxy allows the use of a domain name and SSL certificate for multiple applications or e-learning systems at once [6]. Reverse proxy also has load balance and failover features which can increase server availability [7]. Virtualization is the basis for creating cloud computing with the type of Infrastructure as a Service (IAAS) [8]. Server virtualization can also be implemented for cost efficiency in terms of hardware. Virtualization can optimize the use of hardware and avoid wasting electricity so that it is cost-effective [8][9]. Proxmox Virtual Environment (PVE) is a high-availability hypervisor that can be used to create and manage virtual machines [10]. PVE can also be used as a virtual server infrastructure in making practical web-based e-learning combined with noVNC technology [11]. In addition to virtualization, storage clustering techniques can also be utilized to increase hardware scalability and availability, especially storage media by placing several storage media in a redundant and distributed manner [12][13].

The use of a single server architecture has various limitations, as previously described [14]. The multiple virtual server architecture is a server architecture designed by breaking a physical computer into several virtual computers so that it can make efficient use of hardware and increase system availability by optimizing interactions between existing virtual servers. The application of a reverse proxy has various functions, such as cost effectiveness and increasing system availability by utilizing load balancing [15]. Likewise with storage clustering, which is able to increase availability in terms of hardware [16]. In building an e-learning system, the use of Moodle is very effective from a development standpoint and easy to use from a production standpoint [17]. Moodle in software already has a guaranteed performance benchmark based on various test results that have been carried out by many people, so there is little possibility of a software bottleneck when building an e-learning system using Moodle [18].

Related to some of these things, this research applies Moodle as an e-learning application that is deployed in a multiple virtual server architecture on Proxmox with the implementation of storage clustering using GlusterFS. The use of a reverse proxy with HAproxy is expected to support aspects of the effectiveness and efficiency of resource use and system availability on the server [19]. Various advantages of the application made in building e-learning systems in research need to be validated in terms of performance in terms of server architecture using a performance analysis method because the performance aspect is very vital in a system. To perform performance analysis, the User Behavior Modeling Performance (UBMP) method is used [20]. UBMP is a performance test model that is effective, comprehensive, and broad so that it can be applied to various server architectures, especially in cloud environments [21]. In testing this method, we comprehensively reviewed the performance of the server architecture models tested.

The purpose of this study is to use User Behavior Modeling Performance (UBMP) with the aim of obtaining effective, comprehensive, and accurate performance assessment results for the architecture being tested [22]. Then a single server-side architecture was also built as a comparison to multiple virtual architectures to analyze its performance and determine the level of reliability of each server architecture. The application of the test environment was carried out by the Proxmox server for server implementation on each architecture using the Debian 8 operating system.

II. THEORETICAL REFERENCE

II.1 MOODLE

Moodle is an e-learning platform designed to provide educators, students, and administrators with one integrated system [23]. E-Learning is the principle of direct learning, and in its application, it promotes independent learning, namely web-based distance learning that can be accessed via the internet network [24]. Moodle provides digital classrooms to access material or anything related to learning that is freely accessible to anyone, anytime, anywhere [25]. The advantage of using Moodle is that it is open source, so someone with programming skills can adjust and develop existing features according to their needs and desires [26].

II.2 REVERSE PROXY

The reverse proxy method is a technique that places a proxy server as a middle layer between the client (the frontend layer) and the server (the backend layer). The use of reverse proxies is generally used as load balancing to share server load and can also run caching features to avoid sending the same content repeatedly so as to reduce web server load [27].

Load balancing on the reverse proxy can minimize downtime so that the system has high availability. Haproxy is software that can be used for reverse proxy and load balancing with reliable performance [28]. In addition, its application can also be used in various cases. Among them is to save on the allocation of domain names with the domain forwarding technique or the SSL termination technique by only placing encrypted communications at the frontend layer so that you can save on the use of SSL certificates.

II.3 VIRTUALIZATION

Virtualization is a computer device where the amount of hardware resources can be created, managed, and allocated as needed through software, and various operating systems can be installed in it [29]. The resources used in virtualization come from the resources owned by the physical computer where the virtual computer is created (the host). Resource allocation can be done because of the hypervisor technology. The hypervisor plays the role of managing, running, and monitoring virtual machines [30].

Virtualization is the basis for cloud computing providers Infrastructure as a Service (IAAS) services [31]. Several cloud service providers, such as Amazon, also use virtualized machine technology in the cloud services they offer. Generally, the purpose of virtualization is to optimize hardware use and avoid wasting electricity. One of the hypervisors with high availability capabilities is Proxmox, which can manage virtual machines with KVM and containers with LCX [32][33]. The proxmox architecture can be seen in Figure 1.

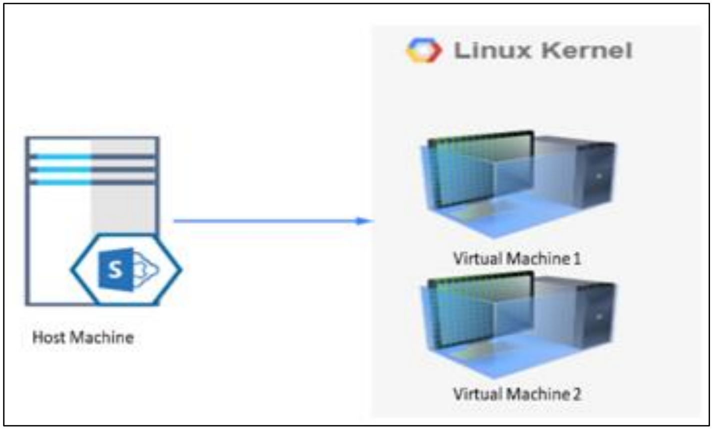


Figure 1: Hypervisor Proxmox Architecture.
Source: Author, (2023).

II.4 TOOL FOR TESTING

There are currently a number of open-source tools that can be used for stress testing web servers or web services. In this article, we focus on tools suitable for generating HTTP and HTTPS loads (GET, POST, and PUT) and FTP loads (PASV and RETR file transfers). Only tools with distribution open licenses (GNU, Apache License, etc.) and executables on the Linux kernel operating system were selected for performance analysis using Apache JMeter [34]. Apache JMeter is open-source software used to perform load testing, performance testing, and functional testing on web applications and other web services [35].

II.5 USER BEHAVIOR MODELING PERFORMANCE (UBMP)

The scope of performance testing is very broad, and the differences in architecture and resources used in it also affect the aspects tested [36]. UBMP testing will produce a performance test model that is effective, comprehensive, and broad so that it can be applied to various server architectures, especially in cloud environments. The optimum value and maximum concurrent users are important indicators to evaluate system performance. In traditional performance testing, there are problems with high costs and limited resources, so the maximum concurrent users cannot be achieved, and the result is that the maximum access load on the system cannot be estimated.

UBMP overcomes these problems using the inflection point performance index, which is an analysis of the performance index based on three performance indicators, namely CPU usage, throughput, and availability. UBMP divides system performance into three parts based on the optimum value and maximum concurrent users, namely light load, smooth load, and heavy load. The larger the user, the greater the increase in CPU and throughput at a certain point and a decrease in the level of availability. The optimum value for concurrent users is obtained when the throughput is at its maximum. The maximum concurrent user value is obtained when the system availability level drops to 70%.

The light load condition occurs when the number of users is below the user's optimum value, the level of availability is high, and the throughput and CPU start to gradually increase. Smooth load conditions are in the range of optimum values and maximum concurrent users. The heavy load condition occurs when the number of concurrent users is above the maximum concurrent user value.

Testing with UBMP uses several user behavior parameters to simulate testing under real-world conditions. The parameters used are thinking time, ramp-up period, and functional test plan. The UBMP test flow to obtain optimum values and maximum concurrent users to determine light, smooth, and heavy load conditions on the system can be seen in Figure 2.

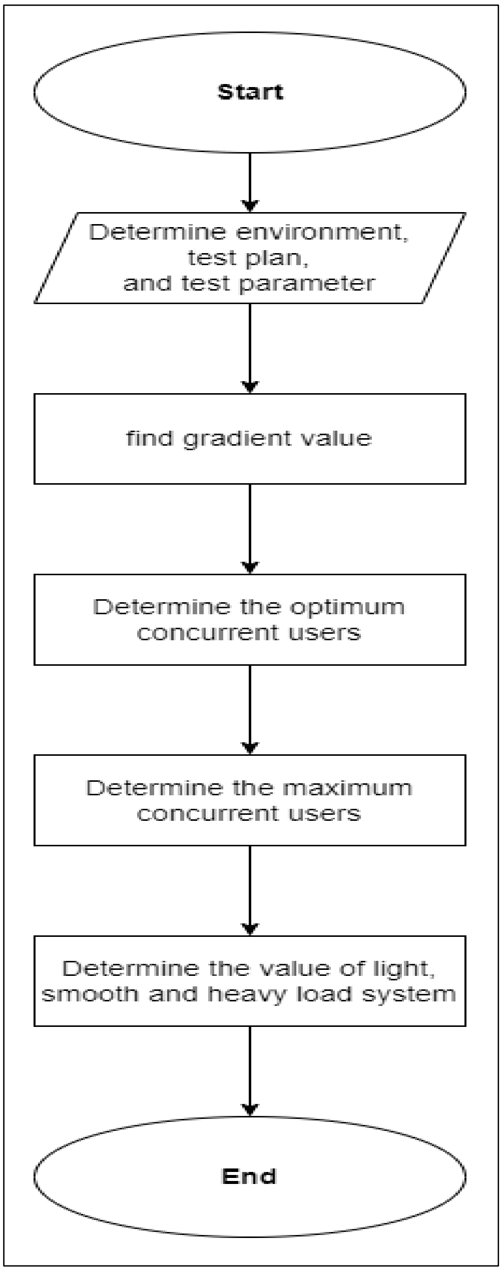

Start
Determine environment, test plan, and test parameter
find gradient value
Determine the optimum concurrent users
Determine the maximum concurrent users
Determine the value of light, smooth and heavy load system
End

Figure 2: UBMP flowchart.
Source: Author, (2023).

II.5 STORAGE CLUSTERING

Storage clustering is a method of grouping several storage media into a logic storage so that the stored data is distributed among the storage media that are joined in a group [37]. This method is widely applied for the scalability and availability of data storage. If a storage medium is damaged, the data stored is still safe because the data is replicated on other storage media in the cluster. One of the Replication Cluster High-Availability Storage technologies is GlusterFS [38]. The application of storage clustering with glusterFS in this study is used to store user data in Moodle applications on two replicated web servers. The storage clustering scheme used can be seen in Figure 3.

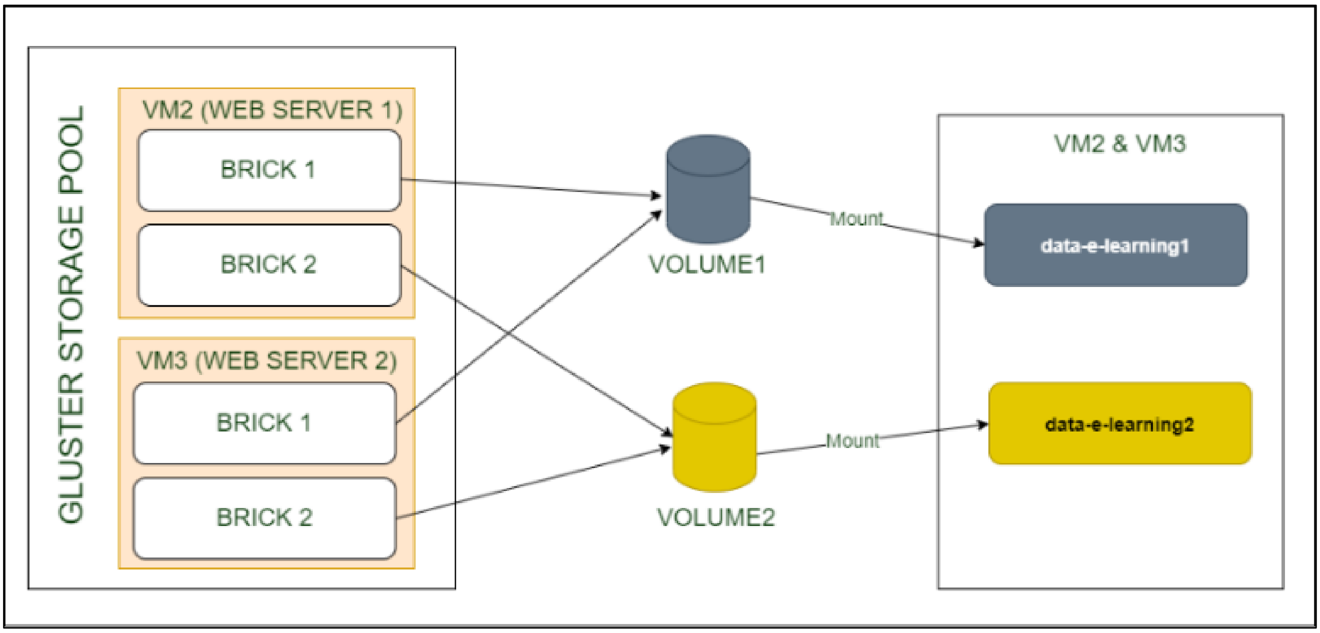


Figure 3: Storage Clustering Architecture.
Source: Author, (2023).

III. MATERIALS AND METHODS

The method used in this study was carried out in the following stages: preparing, planning, designing, implementing, testing, and optimizing. This method results in a standard cycle of network development and management. The flow of the method is shown in Figure 4.

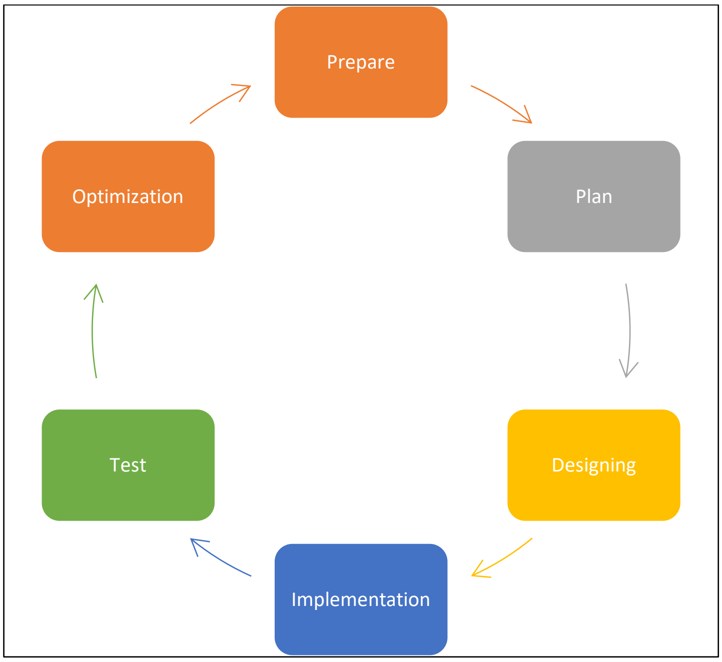


Figure 4: Research Methodology.
Source: Author, (2023).

The stages of the research are as follows:

1. **Prepare**
   At this stage, preparations are made in the form of needs analysis on cloud system services and distribution models, performance test models on cloud systems, Proxmox Virtual Environment systems, e-learning Moodle systems, and JMeter testing.
2. **Plan**
   Identify network architecture designs on Single Server-Side and Multiple Virtual Server-Side Architecture systems.
3. **Designing**
   Designing a network architecture topology system by building an e-learning Moodle system, four virtualization servers, storage clustering, a DNS server, and a web server according to testbed specifications.
4. **Implementation**
   Implement the network architecture design that has been made starting from the installation, network addressing, and service configuration stages until the system is ready for use according to the testbed parameters.
5. **Test**
   Conduct network architecture design tests that are built, which include maintenance, fault detection in the network, and monitoring performance in the implemented network architecture, using testing tools. The testing tool runs using the parameters in the test plan.
6. **Optimization**
   Repairing errors in the network, optimizing network performance, and so on for the suitability and convenience of using the network in the business processes required by the user in accordance with the test plan parameters.

III.1 TESTBED

Multiple virtual server-side architectures were designed using virtualization technology in PVE to build two different e-learning systems using Moodle. Utilizing the second reverse proxy technique, e-learning can be accessed using only a domain name and SSL certificate with the addresses https://lms.edu/lms1 for the first e-learning and https://lms.edu/lms2 for the second e-learning.

There are four virtualized servers on a physical machine, namely VM1 DNS and proxy server, VM2 web server 1, VM3 web server 2, and VM4 database server. VM3 is a replication of VM2 Web Server 1 for load balancing and high availability. Storage clustering is applied to storage media in VM2 and VM3. All virtual servers use the 64-bit Debian 10 (Buster) operating system with a command-line interface. The DNS server uses bind9, the proxy server uses Haproxy, the web server uses NGINX, and the database server uses MariaDB. The multiple virtual server-side architecture testbed design can be seen in Figure 5.

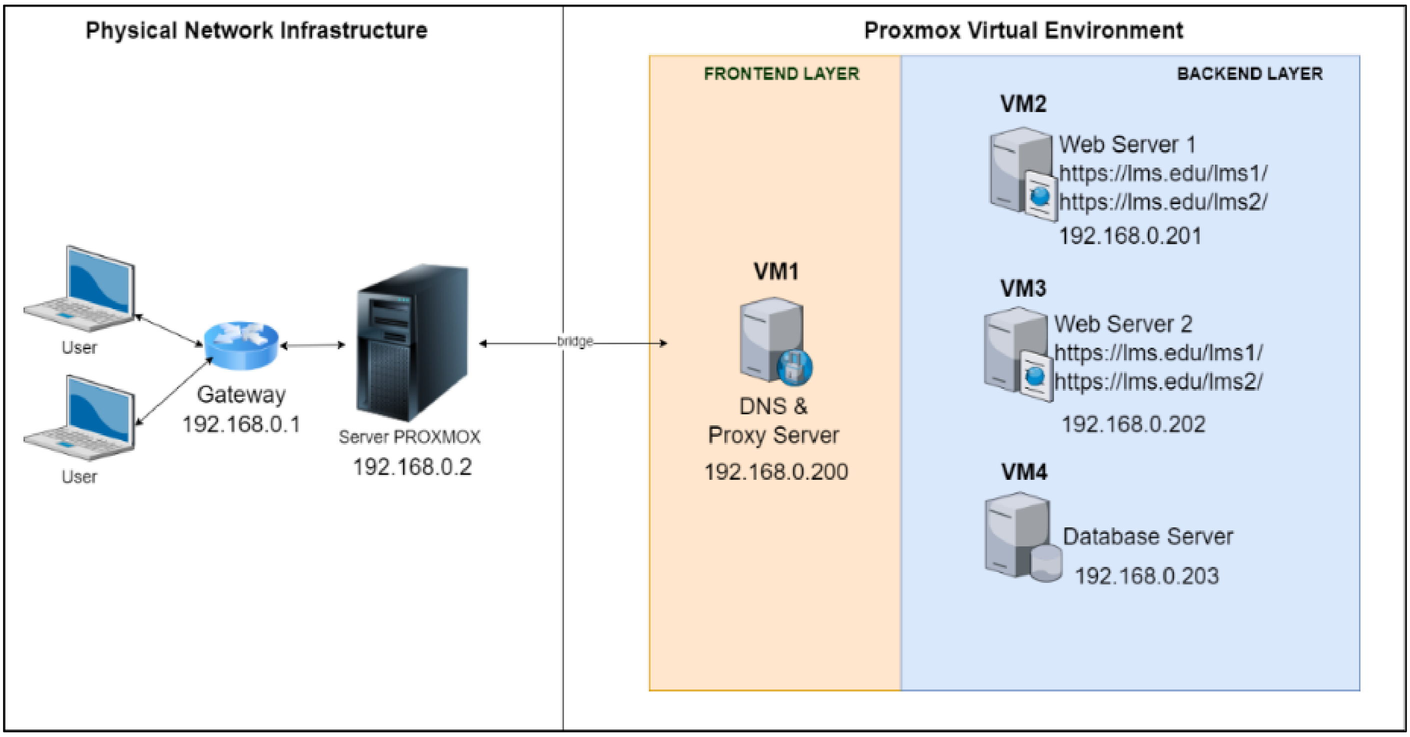


Figure 5: Testbed (Multiple Virtual Server-Side Architecture).
Source: Author, (2023).

The hardware specifications used in the multiple virtual server-side architecture can be seen in Table 1 below.

Table 1: Multiple Virtual Server-Side Architecture Specification.

Server	Processor	RAM	Storage (HDD)
Physical server (Proxmox)	AMD FX (tm)-6350 Six-Core	8GB	1TB
VM1 (DNS & Proxy)	1 Core	1 GB	8 GB
VM2 (Web Server 1)	2 Core	2 GB	8 GB
VM3 (Web Server 2)	2 Core	2 GB	8 GB
VM4 (Database Server)	1 Core	1 GB	8 GB

Source: Author, (2023).

The same case study in the form of deploying two e-learning applications and the same services on multiple virtual server-side architectures built in a testbed in the form of a single server-side architecture that uses specifications in the form of an accumulation of specifications used by virtual servers on multiple virtual server-side architectures as a comparison to analyze its performance. The testbed design on a single server-side architecture architecture can be seen in Figure 6.

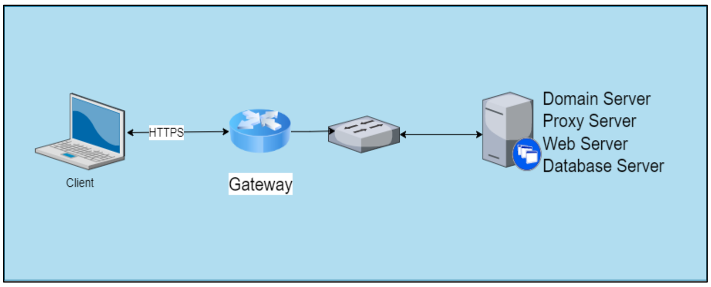


Figure 6: Testbed (Single Server-Side Architecture).
Source: Author, (2023).

The hardware specifications used in the single server-side architecture can be seen in Table 2 below.

Table 2: Single Server Architecture Specification.

Hardware	Specification
CPU	6 Core
RAM	8 GB
Storage (Harddisk)	32 GB
Network Interface Card	Gigabit Ethernet

Source: Author, (2023).

III.2 TEST PLAN

The test uses a load testing technique with the UBMP model. According to the UBMP flowchart in Figure 3, the first step is to determine test plans and test parameters. The flow of the test plan is in the form of accessing the start page, logging in, and accessing a course in e-learning. Tests were carried out on two e-learning courses, namely https://lms.edu/lms1 and https://lms.edu/lms2, simultaneously. The initial test parameters used to find the gradient value are in Table 3 below.

Table 3: Test Parameters.

No	Parameter	Value
1	Concurrent users (Threads)	10
2	Ramp-up periode	5 Threads / 1s
3	Loop	1
4	HTTP request protocol	HTTPS
5	HTTP request host	lms.edu
6	HTTP request host path	/lms1, /lms2
7	Thinking time	1s - 9s (random)

Source: Author, (2023).

The search for gradient values is carried out by adding 5 threads in each test iteration until a 5% increase in CPU is obtained. The gradient value is the difference between the threads in the first iteration and the threads in the nth iteration, which shows a 5% increase in CPU. The gradient value is used as a constant for adding threads at each iteration in the search for optimum and maximum concurrent users.

The search for the optimum value of concurrent users is carried out by adding threads at each test iteration and paying attention to the throughput value. If in the nth iteration the maximum throughput value is obtained, then the threads in the nth iteration are the optimum concurrent users. To get the maximum concurrent users, in the initial iteration of the test, the threads value is filled with the optimum concurrent users value, then updated for the next iteration with a gradient value to obtain a system availability level of up to 70%. Based on the inflection point performance index, the light, smooth, and heavy load conditions of the system can be determined as an accurate and efficient performance index.

IV. RESULTS AND DISCUSSIONS

Implementation of creating a web-based online laboratory has been carried out[6], in this study using The Networked Control System Lab (NCSLab) framework In our research, the network architecture is implemented in a cloud system using the Proxmox Virtual Environment (PVE), with tests based on the Multiple Virtual Server-Side Architecture testbed and the Single Server-Side testbed. Then a test plan is carried out based on the test parameters according to Table 3. The test is carried out based on the process flow of Availability Testing on Multiple Virtual Server-Side architectures can be seen in Figure 7. In this test, the application of reverse proxy and storage clustering not only saves system operating costs but can also increase the level of system and data availability [40].

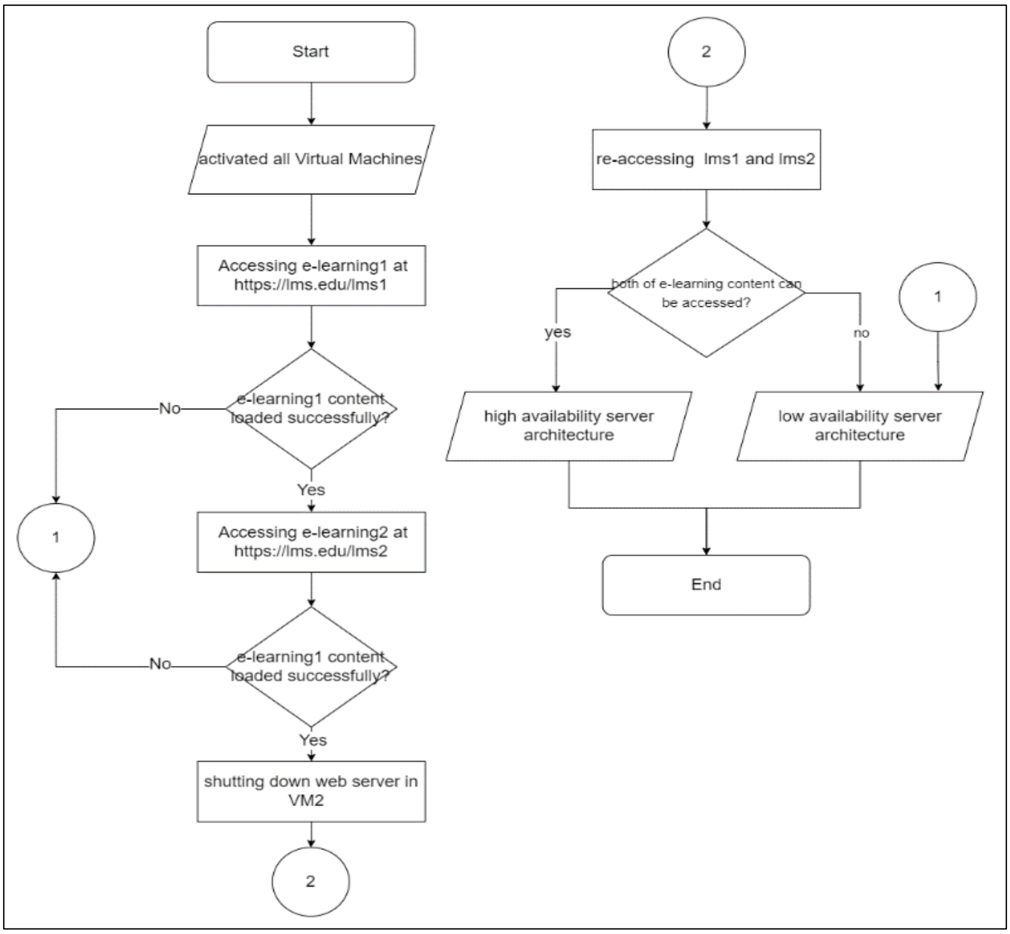

Start
activated all Virtual Machines
Accessing e-learning1 at https://lms.edu/lms1
e-learning1 content loaded successfully?
No
Yes
1
Accessing e-learning2 at https://lms.edu/lms2
e-learning1 content loaded successfully?
No
Yes
shutting down web server in VM2
2
2
re-accessing lms1 and lms2
both of e-learning content can be accessed?
yes
no
1
high availability server architecture
low availability server architecture
End

Figure 7: Availability Testing on Multiple Virtual Server-Side Architecture.
Source: Author, (2023).

Based on the testbed and testplan on the system, the gradient value obtained is 10 concurrent users[41], which is obtained by the difference in threads in the 1st and 3rd iterations with a difference in CPU usage of 5%, which can be seen in Table 4.

Table 4: CPU Usage in Gradient Value Search.

Iterations	Threads	CPU Usage				
		VM1	VM2	VM3	VM4	Average
1	10	2.1%	23.3%	24.8%	17.4%	16.9%
2	15	2.2%	29.7%	29.0%	19.1%	20.0%
3	20	2.3%	30.8%	31.3%	21.3%	21.4%
4	25	2.9%	34.5%	30.7%	27.8%	23.9%
5	30	2.7%	37.4%	38.5%	32.3%	27.7%

Source: Author, (2023).

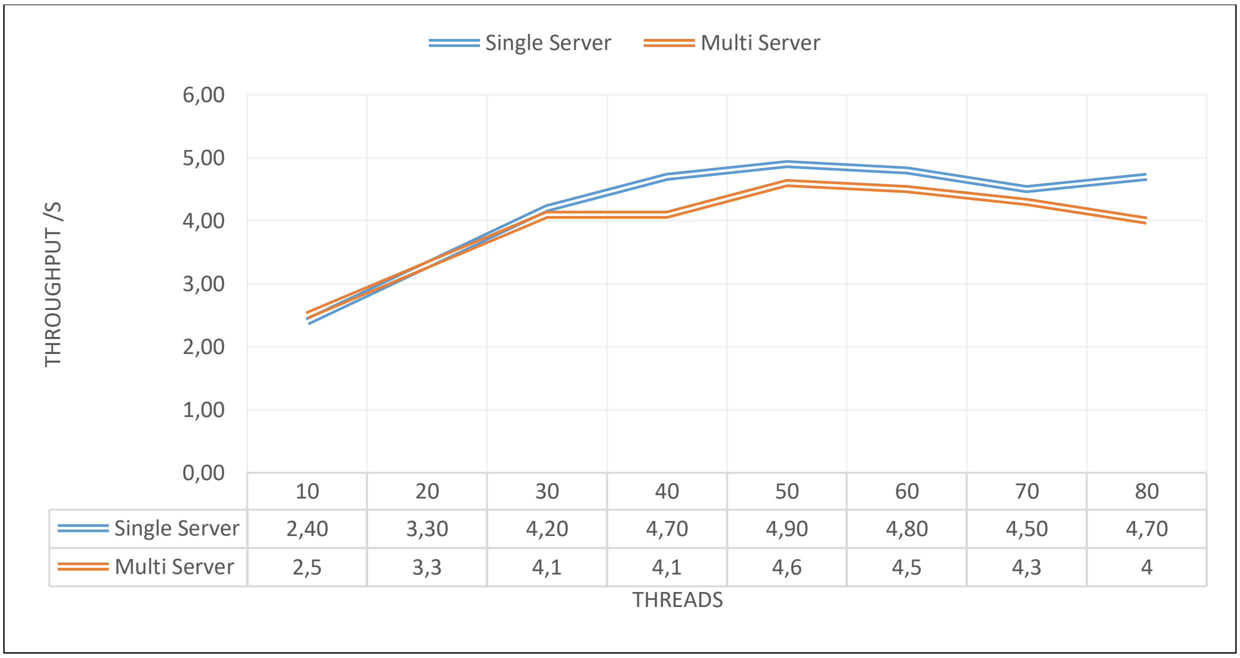
From the results of testbed and testplan testing, it can be concluded that server virtualization with Proxmox Virtual Environment (PVE) is able to optimize hardware performance on physical server machines, then optimize server performance by dividing it into several virtual servers so that it has high efficiency in resource use. From this test, it was found that the Proxmox server was stable [42].

The test results showing that both server architectures are able to optimally handle 50 users with a maximum throughput of 4.6 s on multiple virtual servers and 4.9 s on a single server are shown in Table 5. For optimum results for concurrent users, shown in Figure 8, From this test, it was found that the value of concurrent users was appropriate [42].

Table 5: Finding Optimum Concurrent Users Value Through Maximum Throughput.

Iterations	Threads	Multiple Virtual Server		Single Server	
		Throughput	Error	Throughput	Error
1	10	2.5/s	0.0%	2.4/s	0.0%
2	20	3.3/s	0.0%	3.3/s	0.0%
3	30	4.1/s	0.0%	4.2/s	0.0%
4	40	4.1/s	1.5%	4.7/s	0.0%
5	50	4.6/s	7.0%	4.9/s	0.0%
6	60	4.5/s	11.4%	4.8/s	4.2%
7	70	4.3/s	16.4%	4.5/s	7.1%
8	80	4.0/s	16.2%	4.7/s	21.6%

Source: Author, (2023).



	10	20	30	40	50	60	70	80
Single Server	2,40	3,30	4,20	4,70	4,90	4,80	4,50	4,70
Multi Server	2,5	3,3	4,1	4,1	4,6	4,5	4,3	4

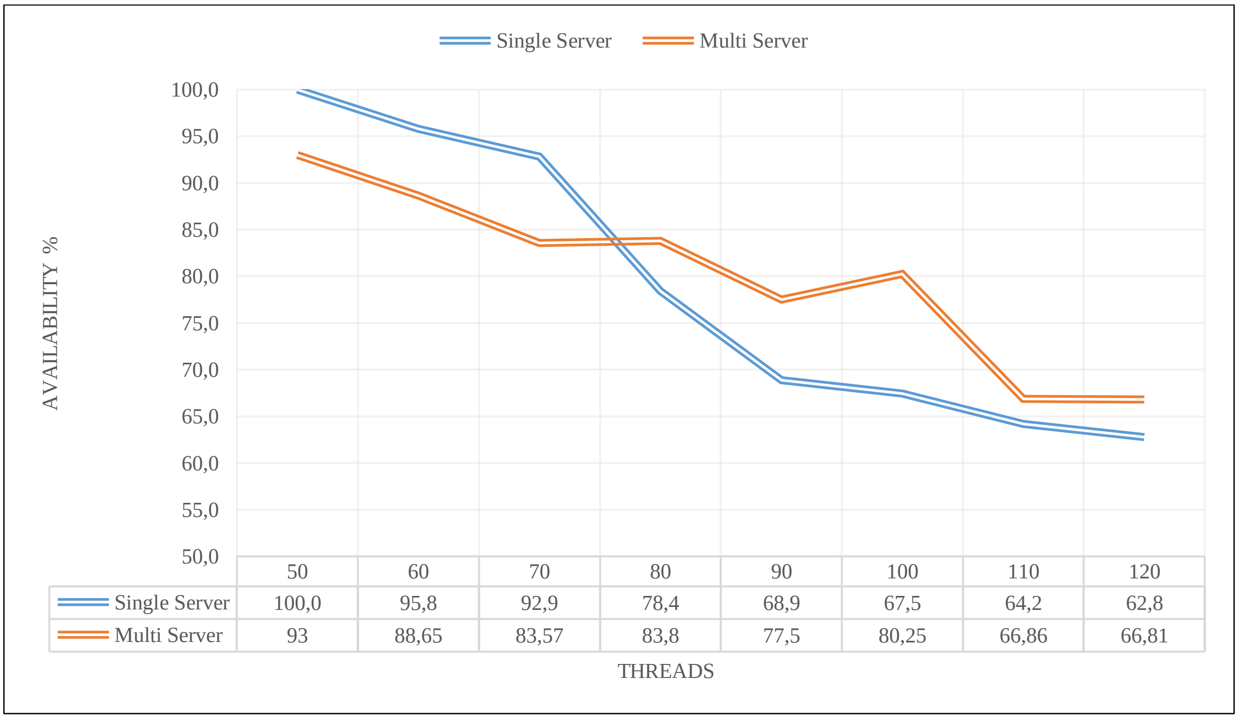
Figure 8: Optimum concurrent users.
Source: Author, (2023).

The maximum concurrent users obtained in the test show the results of the multiple virtual server-side architecture having better performance, which is able to handle a maximum of 100 users with an availability level of 80%, while the single server architecture is only able to handle as many as 80 users with an availability level of 78.4%. can be seen in Table 6. The optimum results for concurrent users are can be seen in Figure 9.

Table 6: Finding Maximum Concurrent Users Value Through The Availability.

Iterations	Threads	Multiple Virtual Server		Single Server	
		Availabilty	Error	Availabilty	Error
1	50	93.0%	7.0%	100 %	0.0%
2	60	88.7%	11.3%	95.8%	4.2%
3	70	83.6%	16.4%	92.9%	7.1%
4	80	83.8%	16.2%	78.4%	21.6%
5	90	77.5%	22.5%	68.9%	31.1%
6	100	80.3%	19.7%	67.5%	32.5%
7	110	66.9%	33.1%	64.2%	35.8%
8	120	66.8%	33.2%	62.8%	37.2%

Source: Author, (2023).



	50	60	70	80	90	100	110	120
Single Server	100,0	95,8	92,9	78,4	68,9	67,5	64,2	62,8
Multi Server	93	88,65	83,57	83,8	77,5	80,25	66,86	66,81

Figure 9: Maximum Concurrent Users.
Source: Author, (2023).

Based on the results obtained from the research, the performance of the multiple virtual server architecture is superior to the single server architecture by handling 20 more users in the system in the test scope when the system is tested for performance.

V. CONCLUSIONS

Based on research data, the UBMP performance test model is a testing model that is comprehensive, effective, and accurate and can be applied to systems with web-based applications implemented in Proxmox Virtual Environment (PVE). Multiple virtual server-side architectures combined with the application of reverse proxy and storage clustering techniques in research are able to optimally use existing resources and are more efficient, effective, and superior in performance than single server architectures. Multiple virtual server-side architectures can not only be applied to e-learning systems but can also be applied to other web-based application systems. Further development is needed regarding the network architecture design used in research to obtain more optimal performance and efficiency in the implementation of Proxmox Virtual Environment (PVE).

VI. AUTHOR’S CONTRIBUTION

Conceptualization: Yuri Ariyanto.
Methodology: Yuri Ariyanto.
Investigation: Yuri Ariyanto.
Discussion of results: Yuri Ariyanto.
Writing – Original Draft: Yuri Ariyanto.
Writing – Review and Editing: Yuri Ariyanto.
Resources: Yuri Ariyanto.
Supervision: Yuri Ariyanto.
Approval of the final text: Yuri Ariyanto.

VII. REFERENCES

[1] S. P. Chung, Y. J. Lu, and Y. C. Lai, “Cloud computing with single server threshold and double congestion thresholds,” ICT Express, vol. 4, no. 3, pp. 119–123, 2018, doi: 10.1016/j.icte.2017.03.002.

[2] I. Alharkan, M. Saleh, M. A. Ghaleb, H. Kaid, A. Farhan, and A. Almarfadi, “Tabu search and particle swarm optimization algorithms for two identical parallel machines scheduling problem with a single server,” J. King Saud Univ. - Eng. Sci., vol. 32, no. 5, pp. 330–338, 2020, doi: 10.1016/j.jksues.2019.03.006.

[3] P. Jain, Y. Munjal, J. Gera, and P. Gupta, “Performance Analysis of Various Server Hosting Techniques,” Procedia Comput. Sci., vol. 173, no. 2019, pp. 70–77, 2020, doi: 10.1016/j.procs.2020.06.010.

[4] R. Entezari-Maleki, L. Sousa, and A. Movaghar, “Performance and power modeling and evaluation of virtualized servers in IaaS clouds,” Inf. Sci. (Ny)., vol. 394–395, pp. 106–122, 2017, doi: 10.1016/j.ins.2017.02.024.

[5] E. Ali, Susandri, and Rahmaddeni, “Optimizing Server Resource by Using Virtualization Technology,” Procedia Comput. Sci., vol. 59, no. Iccsci, pp. 320–325, 2015, doi: 10.1016/j.procs.2015.07.572.

[6] Z. Lei, H. Zhou, S. Ye, W. Hu, and G. P. Liu, “Cost-effective server-side re-deployment for web-based online laboratories using nginx reverse proxy,” IFAC-PapersOnLine, vol. 53, no. 2, pp. 17204–17209, 2020, doi: 10.1016/j.ifacol.2020.12.1748.

[7] Y. Tao and G. Chen, “An extensible universal reverse proxy architecture,” Proc. - 2016 Int. Conf. Netw. Inf. Syst. Comput. ICNISC 2016, pp. 8–11, 2017, doi: 10.1109/ICNISC.2016.45.

[8] A. Rashid and A. Chaturvedi, “Virtualization and its Role in Cloud Computing Environment,” Int. J. Comput. Sci. Eng., vol. 7, no. 4, pp. 1131–1136, 2019, doi: 10.26438/ijcse/v7i4.11311136.

[9] R. Kaur and S. Chopra, “Virtualization In Cloud Computing : A Review,” Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 3307, pp. 01–05, 2020, doi: 10.32628/cseit20641.

[10] V. Tiwari, A. A. Waoo, D. Akhilesh, A. Waoo, and B. Garg, “Study on Virtualization Technology and Its Importance in Cloud Computing Environment See Profile Study on Virtualization Technology and Its Importance in Cloud Computing Environment,” Int. J. Creat. Res. Thoughts, vol. 8, no. 2, pp. 2320–2882, 2020, [Online]. Available: www.ijcrt.orgwww.ijcrt.org.

[11] M. K. L. Singh, W. A. T. Quijano, and G. Koneru, “Evaluation of network performance-type1 open source virtualization platforms,” 2015 Int. Conf. Comput. Commun. Informatics, ICCCI 2015, 2015, doi: 10.1109/ICCCI.2015.7218073.

[12] H. Mao, “High-density information security storage method of big data center based on fuzzy clustering,” Microprocess. Microsyst., vol. 81, no. November 2020, p. 103772, 2021, doi: 10.1016/j.micpro.2020.103772.

[13] A. Pérez, G. Moltó, M. Caballer, and A. Calatrava, “Serverless computing for container-based architectures,” Futur. Gener. Comput. Syst., vol. 83, pp. 50–59, 2018, doi: 10.1016/j.future.2018.01.022.

[14] A. B. Prasetijo, E. D. Widianto, and E. T. Hidayatullah, “Performance comparisons of web server load balancing algorithms on HAProxy and Heartbeat,” Proc. - 2016 3rd Int. Conf. Inf. Technol. Comput. Electr. Eng. ICITACEE 2016, pp. 393–396, 2017, doi: 10.1109/ICITACEE.2016.7892478.

[15] J. E. C. De La Cruz and I. C. A. R. Goyzueta, “Design of a high availability system with HAProxy and domain name service for web services,” Proc. 2017 IEEE 24th Int. Congr. Electron. Electr. Eng. Comput. INTERCON 2017, 2017, doi: 10.1109/INTERCON.2017.8079712.

[16] V. A. M and P. Sonpatki, ReactJS by Example: Building Modern Web Applications with React. 2016.

[17] L. L. Salekhova, K. S. Grigorieva, and T. A. Zinnurov, “Using LMS moodle in teaching CLIL: A case study,” Proc. - Int. Conf. Dev. eSystems Eng. DeSE, vol. October-20, no. May 2022, pp. 393–395, 2019, doi: 10.1109/DeSE.2019.00078.

[18] A. Villar-Martinez, L. Rodriguez-Gil, I. Angulo, P. Orduna, J. Garcia-Zubia, and D. Lopez-De-Ipina, “Improving the Scalability and Replicability of Embedded Systems Remote Laboratories through a Cost-Effective Architecture,” IEEE Access, vol. 7, pp. 164164–164185, 2019, doi: 10.1109/ACCESS.2019.2952321.

[19] S. Kanthavar, S. Simeen, M. Chawla, and S. Sarayu, “Design of an Architecture for Cloud Storage to Provide Infrastructure as a Service (IaaS),” 2017 14th IEEE India Counc. Int. Conf. INDICON 2017, 2018, doi: 10.1109/INDICON.2017.8488000.

[20] F. Zhang, Y. Liu, J. Mao, M. Zhang, and S. Ma, “User behavior modeling for Web search evaluation,” AI Open, vol. 1, no. March, pp. 40–56, 2020, doi: 10.1016/j.aiopen.2021.02.003.

[21] H. H. Li, X. R. Li, H. Wang, J. Zhang, and Z. X. Jiang, “Research on cloud performance testing model,” Proc. IEEE Int. Symp. High Assur. Syst. Eng., vol. 2019-Janua, no. January 2019, pp. 179–183, 2019, doi: 10.1109/HASE.2019.00035.

[22] A. F. Wicaksono and A. Moffat, “Metrics, User Models, and Satisfaction,” pp. 654–662, 2020.

[23] P. He, J. Qiu, and B. Zhai, “Study on the integration of cloud computing and moodle learning platform,” Proc. 2015 IEEE Int. Conf. Commun. Softw. Networks, ICCSN 2015, pp. 367–371, 2015, doi: 10.1109/ICCSN.2015.7296185.

[24] C. De Medio, C. Limongelli, F. Sciarrone, and M. Temperini, “MoodleREC: A recommendation system for creating courses using the moodle e-learning platform,” Comput. Human Behav., vol. 104, p. 106168, 2020, doi: 10.1016/j.chb.2019.106168.

[25] M. Zabolotniaia, Z. Cheng, E. M. Dorozhkin, and A. I. Lyzhin, “Use of the LMS Moodle for an effective implementation of an innovative policy in higher educational institutions,” Int. J. Emerg. Technol. Learn., vol. 15, no. 13, pp. 172–189, 2020, doi: 10.3991/ijet.v15i13.14945.

[26] W. Fernando, “Moodle quizzes and their usability for formative assessment of academic writing,” Assess. Writ., vol. 46, no. September, p. 100485, 2020, doi: 10.1016/j.asw.2020.100485.

[27] D. Arnaldy and T. S. Hati, “Performance Analysis of Reverse Proxy and Web Application Firewall with Telegram Bot as Attack Notification on Web Server,” 2020 3rd Int. Conf. Comput. Informatics Eng. IC2IE 2020, pp. 455–459, 2020, doi: 10.1109/IC2IE50715.2020.9274592.

[28] L. H. Pramono, R. C. Buwono, and Y. G. Waskito, “Round-robin algorithm in HAProxy and nginx load balancing performance evaluation: A review,” 2018 Int. Semin. Res. Inf. Technol. Intell. Syst. ISRITI 2018, no. July, pp. 367–372, 2018, doi: 10.1109/ISRITI.2018.8864455.

[29] K. Kritikos and P. Skrzypek, “Simulation-as-a-service with serverless computing,” Proc. - 2019 IEEE World Congr. Serv. Serv. 2019, vol. 2642–939X, pp. 200–205, 2019, doi: 10.1109/SERVICES.2019.00056.

[30] L. Hernandez, G. Jimenez, A. Pranolo, and C. U. Rios, “Comparative Performance Analysis Between Software-Defined Networks and Conventional IP Networks,” Proceeding - 2019 5th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Cyber Phys. Syst. ICSITech 2019, pp. 235–240, 2019, doi: 10.1109/ICSITech46713.2019.8987493.

[31] K. Arjunan and C. N. Modi, “An enhanced intrusion detection framework for securing network layer of cloud computing,” ISEA Asia Secur. Priv. Conf. 2017, ISEASP 2017, 2017, doi: 10.1109/ISEASP.2017.7976988.

[32] J. Yang and Y. Lan, “A performance evaluation model for virtual servers in KVM-based virtualized system,” Proc. - 2015 IEEE Int. Conf. Smart City, SmartCity 2015, Held Jointly with 8th IEEE Int. Conf. Soc. Comput. Networking, Soc. 2015, 5th IEEE Int. Conf. Sustain. Comput. Communic, pp. 66–71, 2015, doi: 10.1109/SmartCity.2015.49.

[33] Y. Ariyanto, B. Harijanto, A. Setiawan, S. Adhisuwignjo, and E. Rohadi, “Integration of digital library server with Service Oriented Architecture (SOA) based on cloud computing using proxmox server,” J. Phys. Conf. Ser., vol. 1402, no. 7, 2019, doi: 10.1088/1742-6596/1402/7/077054.

[34] S. Grabovsky, P. Cika, V. Zeman, V. Clupek, M. Svehlak, and J. Klimes, “Denial of Service Attack Generator in Apache JMeter,” Int. Congr. Ultra Mod. Telecommun. Control Syst. Work., vol. 2018-November, pp. 1–4, 2019, doi: 10.1109/ICUMT.2018.8631212.

[35] P. Cika, S. Grabovsky, V. Zeman, and V. Clupek, “Network Emulator of Transmission Parameters of Data Networks,” Int. Congr. Ultra Mod. Telecommun. Control Syst. Work., vol. 2018-November, no. 1, pp. 1–6, 2019, doi: 10.1109/ICUMT.2018.8631254.

[36] S. Pradeep and Y. K. Sharma, "A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Applications," Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019, pp. 399–403, 2019, doi: 10.1109/AICAI.2019.8701327.

[37] O. Hkdylru et al., "Cloud Storage Behavior Analysis Using Time Series Clustering," pp. 90–95, 2018.

[38] Z. Tang, S. Sha, J. Zhang, Z. Li, H. Peng, and H. Du, "WebGlusterFS2.0: A Web-based Administration Tool for GlusterFS with Unattended Deployment and Automatic Configuration of Xlator," 2019 Comput. Commun. IoT Appl. ComComAp 2019, pp. 276–281, 2019, doi: 10.1109/ComComAp46287.2019.9018744.

[39] Q. Ye, W. Hu, and H. Zhou, "Implementation of WebVR-based laboratory for control engineering education based on NCSLab framework," Chinese Control Conf. CCC, pp. 7880–7885, 2017, doi: 10.23919/ChiCC.2017.8028601.

[40] Z. Yang and X. Huang, "Dynamic Configuration of Reverse Proxy Cache based on Multi-Dimensional Time Series Prediction of Visit Traffic," pp. 237–240.

[41] H. H. Li, X. R. Li, H. Wang, J. Zhang, and Z. X. Jiang, "Research on cloud performance testing model," Proc. IEEE Int. Symp. High Assur. Syst. Eng., vol. 2019-Janua, no. 2015, pp. 179–183, 2019, doi: 10.1109/HASE.2019.00035.

[42] L. Chen, W. Huang, A. Sui, D. Chen, and C. Sun, "The online education platform using Proxmox and noVNC technology based on Laravel framework," Proc. - 16th IEEE/ACIS Int. Conf. Comput. Inf. Sci. ICIS 2017, pp. 487–491, 2017, doi: 10.1109/ICIS.2017.7960041.