



Performance Analysis of Mobile Learning Systems on Cloud Computing Using Load Testing Methods

Yuri Ariyanto¹, Budi Harijanto², Atiqah Nurul Asri³, A. Yahya Hudan Permana⁴, Muhammad Nuruddin Ismail⁵ and Sofyan Noor Arief⁶

^{1,2,3,4,5,6}State Polytechnic of Malang, Malang 65142, Indonesia

¹yuri@polinema.ac.id

²budi.harijanto@polinema.ac.id

³atiqah.nurul@polinema.ac.id

⁴yahyahudan19@gmail.com

⁵nurudin_2@live.com

⁶sofyan@polinema.ac.id

Abstract. Along with developing the Mobile Learning System, Mobile Learning requires adequate facilities, such as a server as a storage medium and as a provider of an API-based web service, especially when there are more and more Mobile Learning users. And to ensure the quality of the web service, a system test is needed, namely Load Testing. The load testing method uses Apache JMeter tools to check and test the performance of a server. Load testing is done by testing the existing web services on the cloud server where there are 4 web servers, 1 database server, and 1 load balancing server. Scenario testing is based on load data of 100, 500, and 1000 within 10 seconds. The results obtained from the test data are the average test time with a consistent value in each scenario, with the throughput value getting a very good value because the higher the request is given. The HAProxy server is running well in sharing requests to each server with an error value of 0%.

Keywords: Apache JMeter, Cloud Server, Load Testing, Mobile Learning

1 Introduction

Mobile Learning is a technology developed for learning media or e-learning [1]. Mobile learning itself is also an alternative method to solve educational problems in the world, such as learning materials, practice questions, trying out exams, and so on [2] [3].

However, in its development, mobile learning must have adequate requirements such as API (Application Programming Interface)-based Web Service services to facilitate data processing and is developed with other devices. In addition to requiring a web service, the mobile learning system also requires adequate facilities, such as a server that is used for data storage media and can also be used to manage the web service itself [2].

To overcome this, technology is applied so that it can make it possible to procure server infrastructure without buying hardware. It can be done by creating a server virtualization system through cloud computing technology itself [4]. The cloud computing used in this research is Proxmox VE, which has a license under GPL v2. In other words, it is open source and free, so in its development, it can save additional costs [5].

The system that has been designed must have a qualified service quality, significantly mobile learning users will increase and the features that exist in mobile learning will be more complex. Therefore, a test is needed in the form of Load Testing on the Server and the Web Service itself to determine the stability and reliability of the existing system [6].

The system that has been designed must have a qualified service quality, significantly since mobile learning users will increase and the features that exist in mobile learning will be more complex. Therefore, a test is needed in the form of Load Testing on the Server and the Web Service itself to determine the stability and reliability of the existing system [7].

Based on the above reasons, thus, Web Services and Servers on a Cloud Computing-based Mobile Learning System need to be tested using the Load Testing Method to optimize the system's effectiveness and efficiency.

2 Literature Review

This part discusses the results and discussion of Web Service Testing on a Cloud computing-based Mobile Learning System using Apache JMeter tools and analyzes the results of the tests that have been carried out. The results and discussion of this research are as follows.

2.1 API (Application Programming Interface)

API or Application Programming Interface is an integration of two parts of an application system. It consists of elements of functions, protocols, and other tools developers use to create applications. The Web API also supports CRUD functions that work through several HTTP protocols, namely the GET, POST, PUT and DELETE methods [8].

2.2 Proxmox VE

Proxmox is a Debian-based (64-bit) virtualized Linux distribution that supports OpenVZ and KVM. Proxmox allows for centralized management of multiple physical servers. A Proxmox consists of at least one master and multiple nodes (at least one master and one node) [5].

2.3 Load Testing

Load testing is a performance testing technique that measures a response under various load conditions. This test helps determine how the software behaves when multiple users access the software simultaneously. Load testing is needed to simulate access to a mobile application or a website simultaneously. This method is better than having to invite several people at once to access an application or website [9][10].

2.4 Apache Jmeter

JMeter or The Apache JMeter™ is an open-source application based on 100% Java used for performance testing. For a QA Engineer, jMeter can be used to load/stress testing Web Applications, FTP Applications, and Database server tests. Apache JMeter is a Java-based analytical web application that can run on any operating system or test a web server's stability and performance. The types of the can are done with JMeter are Load Testing, Stress Testing, and Web Test Plan [11][12].

3 Research Methodology

This research methodology describes the flow of the stages of performance analysis on a mobile learning system based on cloud computing, as shown in Fig. 1.



Fig. 1. Stages of performance analysis

3.1 Observation and Cloud Computing Installation

This research started by observing the server requirements that would be created and the selection of tools that would be used for system testing. The tools used in this test were Apache JMeter. From the results of observations, a cloud computing architecture was then designed to be used in performance analysis on mobile learning systems, as shown in Fig. 2.

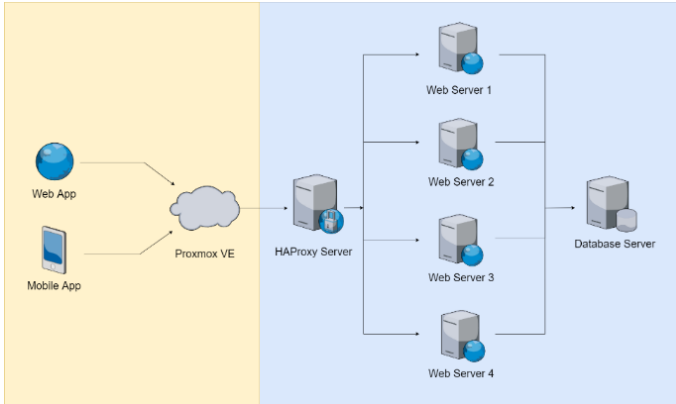


Fig. 2. Cloud computing architecture design

3.2 Load Testing

Load Testing would be carried out using several scenarios, namely providing a stress test by sending 100 requests, 200 requests, and 300 requests

3.3 Analysis of Load Testing Results

After the test was complete, the analysis process was carried out on the test results, namely Web Service analysis includes Average (Average Time), Throughput, Deviation, and Error.

4 Result and Discussion

This part discusses the results and discussion of Web Service Testing on a Cloud computing-based Mobile Learning System using Apache JMeter tools and analyzes the results of the tests that have been carried out. The results and discussion of this research are as follows

4.1 Load Testing

Scenario test data for load testing is done by entering the web service API address for the mobile learning system in the Apache JMeter tool. For the cloud server for testing, there are 4 virtual web servers, 1 database server, and 1 load balancing server. The method used for testing scenarios on the web service API using stress testing for student data, exam data, question data, answer data, and value data, can be seen in Table I.

Table 1. Method Details on the API.

API	Method	Url	Parameter
Student Data	GET	/ API /student/13	-
Exam Data	GET	/ API /exam/7	-
Question Data	GET	/API/getquestion/7	-
Input Answer	POST	/ API /answer	{ "question_id": "test_id": "content_answer": "answer_options": }
Value Data	GET	/ API /value	-

For testing, data from 3 different scenarios, including user, period, and several loops are also used as shown in Table II.

Table 2. Scenarios Load Testing.

Scenario	User	Total	Loop Count	Total
Scenario 1	1	20	10	100
Scenario 2	1	100	10	500
Scenario 3	1	200	10	1000

The test results from the scenarios in table 2 as shown in Fig. 3, 4, and 5.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Data Siswa	20	68	37	274	68,57	0,00%	2,2/sec	1,10	0,28	509,0
Data Ujian	20	49	28	220	41,97	0,00%	2,3/sec	1,72	0,29	775,0
Data Soal	20	40	34	56	5,40	0,00%	2,3/sec	3,03	0,31	1718,0
Input Jawaban	20	164	98	277	61,96	0,00%	2,3/sec	1,13	0,63	512,0
Data Nilai	20	91	45	206	50,47	0,00%	2,3/sec	1,90	0,28	850,0
TOTAL	100	82	28	277	67,27	0,00%	10,8/sec	9,18	1,70	872,8

Fig. 3. First scenario results

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Data Siswa	100	54	27	146	20,83	0,00%	10,1/sec	5,00	1,28	509,0
Data Ujian	100	44	25	114	16,64	0,00%	10,1/sec	7,65	1,27	775,0
Data Soal	100	42	27	275	27,42	0,00%	10,1/sec	17,02	1,37	1718,0
Input Jawaban	100	474	210	859	138,63	0,00%	9,8/sec	4,91	2,73	512,0
Data Nilai	100	57	30	276	34,92	0,00%	10,0/sec	8,28	1,24	850,0
TOTAL	500	134	25	859	182,41	0,00%	47,9/sec	40,85	7,57	872,8

Fig. 4. Second scenario result

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
Data Siswa	200	391	34	1402	304,08	0,00%	19,1/sec	9,48	2,42	509,0
Data Ujian	200	243	30	1170	179,62	0,00%	18,3/sec	13,82	2,30	775,0
Data Soal	200	249	31	1155	182,88	0,00%	18,2/sec	30,56	2,45	1718,0
Data Nilai	200	253	33	984	160,97	0,00%	18,0/sec	14,91	2,23	850,0
Input Jawaban	200	905	264	2171	350,17	0,00%	17,6/sec	8,79	4,89	512,0
TOTAL	1000	408	30	2171	355,16	0,00%	86,4/sec	73,62	13,65	872,8

Fig. 5. Third scenario results

The test results with three scenarios using apache JMeter as shown in Table III.

Table 3. Test Result.

Parameter	Scenario Apache JMeter		
	100 Thread	200 Thread	300 Thread
Average	82 ms	134 ms	408 ms
Deviation	67.27	182.41	355,16
Throughput	10.8/sec	47.9/sec	86.4/sec
Error	0%	0%	0%

4.2 Analysis of Test Results

Tests were carried out with 3 test scenarios on Apache JMeter resulting in several important points. The critical issue was the average, throughput, deviation, and error, as shown in Fig. 6, 7, 8, and 9.

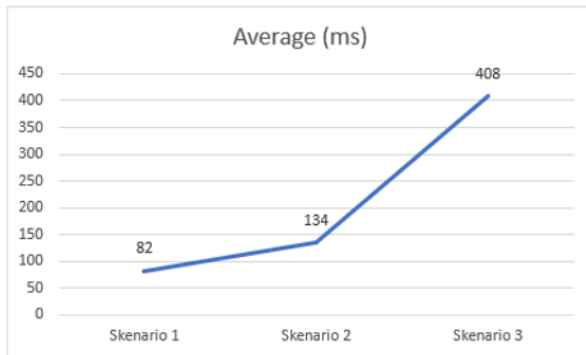


Fig. 6. Graph of results average time

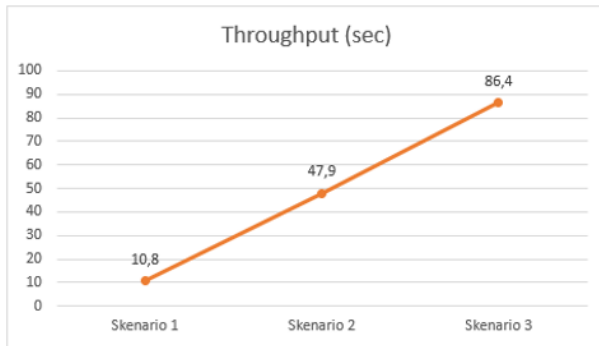


Fig. 7. Graph of throughput test results

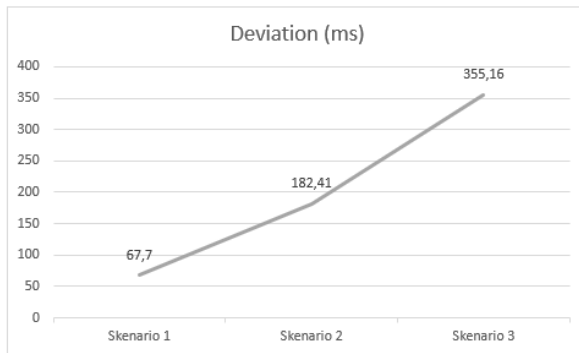


Fig. 8. Deviation test results

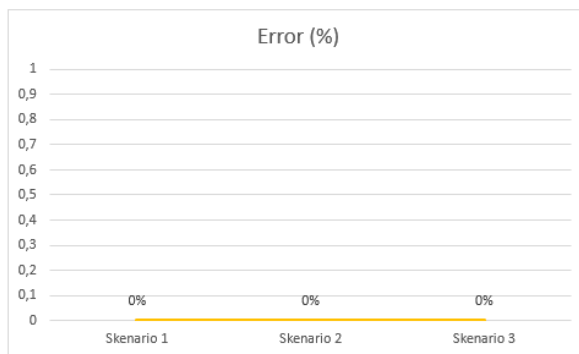


Fig. 9. Error testing results

5 Conclusion

The tests carried out in this study used the load testing method on the web service API using the Apache JMeter tools. The test is carried out on the web service that is on the respective web server. On the load balancer on the HAProxy server, the server can run properly and share the request load to each web server without error with 0% error.

6 Acknowledgment

The authors would like to thank the Ministry of Education, Culture, Research, and Technology (Kemdikbudristek) and the State Polytechnic of Malang who have supported this research project

References

1. Y. Shoaib and O. Das, "Web Application Performance Modeling Using Layered Queuing Networks," in *Electronic Notes in Theoretical Computer Science*, Sep. 2011, vol. 275, no. 1, pp. 123–142, DOI: 10.1016/j.entcs.2011.09.009.
2. G. B. Heimovski, R. C. Turchetti, J. A. Wickboldt, L. Z. Granville, and E. P. Duarte, "FT-Aurora: A highly available IaaS cloud manager based on replication," *Comput. Networks*, vol. 168, Feb. 2020, DOI: 10.1016/j.comnet.2019.107041.
3. Y. Jiugen and X. Ruonan, "Mobile terminal based mobile learning system design," *ICCSE 2016 - 11th Int. Conf. Comput. Sci. Educ.*, no. Iccse, pp. 699–703, 2016, DOI: 10.1109/ICCSE.2016.7581664.
4. M. Nazari Cheraghloou, A. Khadem-Zadeh, and M. Haghparast, "A survey of fault tolerance architecture in cloud computing," *Journal of Network and Computer Applications*, vol. 61, Academic Press, pp. 81–92, Feb. 01, 2016, DOI: 10.1016/j.jnca.2015.10.004.
5. Y. Ariyanto, B. Harijanto, V. A. H. Firdaus, and S. N. Arief, "Performance analysis of Proxmox VE firewall for network security in cloud computing server implementation," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 732, no. 1, 2020, DOI: 10.1088/1757-899X/732/1/012081.
6. A. Sunardi and Suharjito, "MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web-based," in *Procedia Computer Science*, 2019, vol. 157, pp. 134–141, DOI: 10.1016/j.procs.2019.08.150.
7. R. Ramakrishnan and A. Kaur, "Little's law based validation framework for load testing," *Inf. Softw. Technol.*, vol. 108, pp. 88–98, Apr. 2019, DOI: 10.1016/j.infsof.2018.11.007.
8. N. Lukic, N. Teslic, T. Maruna, and V. Mihic, "A java API interface for the search of DTV services in embedded multimedia devices," *IEEE Trans. Consum. Electron.*, vol. 59, no. 4, pp. 875–882, 2013, doi: 10.1109/TCE.2013.6689702.
9. H. Zhu and H. Wu, "Research on web application load testing model," *Proc. 2017 IEEE 2nd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2017*, vol. 2018-January, pp. 1175–1178, 2018, DOI: 10.1109/ITNEC.2017.8284961.
10. B. Vani, R. Deepalakshmi, and S. Suriya, "Web-based testing-An optimal solution to handle peak load," *Proc. 2013 Int. Conf. Pattern Recognition, Informatics Mob. Eng. PRIME 2013*, pp. 5–10, 2013, DOI: 10.1109/ICPRIME.2013.6496439.
11. R. K. Lenka, S. Mangain, S. Kumar, and R. K. Barik, "Performance Analysis of Automated Testing Tools: JMeter and TestComplete," *Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018*, no. 1, pp. 399–407, 2018, DOI: 10.1109/ICACCCN.2018.8748521.

12. 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.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

