Indonesian Journal of Electrical Engineering and Computer Science
Vol. 44, No. 1, October 2026, pp. 108~117
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v44.i1.pp108-117

Improving LMS scalability using cloud native architecture with auto-scaling and load balancing

Endy Sjaiful Alim[1], Tirta Anhari[2], M. Asep Rizkiawan[3]
[1]Department of Informatics Engineering, Muhammadiyah University Prof. Dr. Hamka, Jakarta, Indonesia
[2]Department Information Systems and Technology, Muhammadiyah University Prof. Dr. Hamka, Jakarta, Indonesia
[3]Department of Electrical Engineering, Muhammadiyah University Prof. Dr. Hamka, Jakarta, Indonesia

Article Info

Article history:

Received May 20, 2026
Revised Aug 27, 2026
Accepted Sep 10, 2026

Keywords:

Cloud technology
Learning management system
Online education
Scalability
System performance
User congruence

ABSTRACT

The increasing adoption of online learning systems has significantly increased the workload of learning management systems (LMS), particularly during online examinations involving large numbers of concurrent users. Traditional on-premises LMS infrastructures often experience performance degradation, service interruptions, and limited scalability under peak access conditions. This study proposes and experimentally validates a cloud-native LMS architecture using Google Cloud Platform (GCP) services, including load balancer, managed instance groups with auto-scaling, Cloud SQL, Redis Memorystore, and Cloud Filestore, to improve system scalability and reliability in higher education environments. The proposed architecture was implemented and evaluated in a real-world university LMS environment during online examination sessions involving more than 3,000 concurrent users. Experimental results demonstrated that the cloud-based architecture successfully maintained stable system performance, reduced response times, prevented service downtime, and maintained CPU utilization below critical thresholds during peak workloads. The integration of load balancing and auto-scaling mechanisms enabled dynamic resource allocation and efficient traffic distribution, while Redis caching improved application responsiveness by reducing database access overhead. The main contribution of this research lies in the practical experimental validation of a scalable cloud-native LMS architecture under real-world high-concurrency conditions. The findings demonstrate that cloud computing provides an effective and reliable solution for supporting large-scale online learning and digital examination activities in higher education institutions. This study also provides practical implementation insights that can serve as a reference for universities seeking scalable and resilient LMS infrastructures.

This is an open access article under the CC BY-SA license.



Corresponding Author:

Endy Sjaiful Alim
Department of Informatics Engineering, Muhammadiyah University Prof. Dr. Hamka
Jakarta, Indonesia
Email: endy@uhamka.ac.id

1. INTRODUCTION

The rapid development of information and communication technology (ICT) has significantly transformed the higher education sector, particularly through the widespread adoption of online learning systems. Learning management systems (LMS) have become essential platforms for managing digital learning activities, including course materials, assignments, examinations, and communication between instructors and students [1]-[5]. One of the key tools in online learning is a LMS, which is used to manage learning materials, assignments, exams, and interactions between instructors and students [6]-[12]. Despite

their widespread adoption, LMS platforms continue to face major technical challenges related to scalability and performance, especially during periods of high concurrent access. In large universities, thousands of students may access the LMS simultaneously during online examinations, assignment submissions, or course registration periods. Under these peak load conditions, traditional LMS infrastructures frequently experience performance degradation, including increased response time, login failures, service interruptions, and even complete system crashes [13]-[18]. These issues directly affect the quality of online learning and reduce the reliability of digital academic services.

Most conventional LMS deployments in higher education still rely on on premises server infrastructures with limited scalability and resource flexibility. Increasing local server capacity often requires substantial hardware investment, complex maintenance, and extended deployment time. Consequently, traditional infrastructures are frequently unable to dynamically adapt to sudden increases in user traffic. In recent years, cloud computing technology has emerged as a promising alternative solution due to its elastic scalability, resource virtualization, and automated infrastructure management capabilities [19], [20]. Cloud based systems allow computing resources to be adjusted dynamically according to workload demands, thereby improving service availability and operational efficiency.

Several studies have explored cloud adoption and LMS scalability in education, but most focus on conceptual integration, general performance optimization, or small-scale implementations. Experimental evidence of cloud-native LMS architectures under real-world university examinations with thousands of concurrent users remains limited, particularly regarding the integration of auto-scaling and load balancing on Google Cloud Platform (GCP). Therefore, this study implements and evaluates a cloud-native LMS architecture using GCP services, including load balancing, managed instance groups with auto-scaling, Cloud SQL, Redis Memorystore, and shared storage. The novelty of this research lies in its real-world experimental validation under examination conditions involving approximately 3,000 concurrent users, demonstrating the practical scalability and performance of a cloud-native LMS in a university environment.

This study aims to evaluate the effectiveness of cloud technology in improving LMS scalability, response time, and service reliability under high concurrent user loads. An experimental approach was employed by comparing LMS performance before and after the implementation of the cloud based architecture. Performance testing was conducted using various concurrent user scenarios to analyze system stability, scalability, and resource efficiency. The results of this study are expected to provide practical contributions for universities seeking reliable and scalable LMS infrastructures to support large scale online learning and examination activities.

2. METHOD

This study involves several stages designed to address the issue of frequent spikes in user traffic on university LMS platforms; its objective is to design and implement effective technical solutions. By selecting the appropriate load balancing and auto scaling methods, the study aims to create an LMS system that is more robust, scalable, and capable of optimally meeting user needs. The development method used in this study is the network development life cycle. The stages involved include problem identification, design, implementation, and testing. This cyclical approach allows for continuous improvement until results consistent with the research objectives are achieved [21].

2.1. Load balancing

In an increasingly complex digital age, the need for robust and always available web applications is growing. One of the main challenges in building such applications is managing fluctuating traffic and ensuring that all user requests are handled effectively. This is where load balancing plays a crucial role, as the load balancer acts as a coordinator to distribute traffic evenly across multiple virtual server instances, thereby improving the application's performance, availability, and scalability [22], [23]. Auto scaling is a feature that allows computing resources in a cloud environment, such as GCP, to automatically adjust to changes in workload. With auto scaling, the number of virtual instances can increase or decrease dynamically based on predefined metrics, such as CPU usage, memory, or HTTP requests [24]-[26]. The load balancing diagram is shown in Figure 1 and Scale diagram is shown in Figure 2.

2.2. Design and construction

The architecture of the LMS, as shown in Figure 3, consists of several entities, including the GCP Load Balancer, MIGs (Managed Instance Groups), Compute Engine, Cloud SQL, Cloud Memorystore Redis, Cloud Filestore, and Unoconv Job, each of which has its own specific function. The researchers designed each entity with the following specifications:

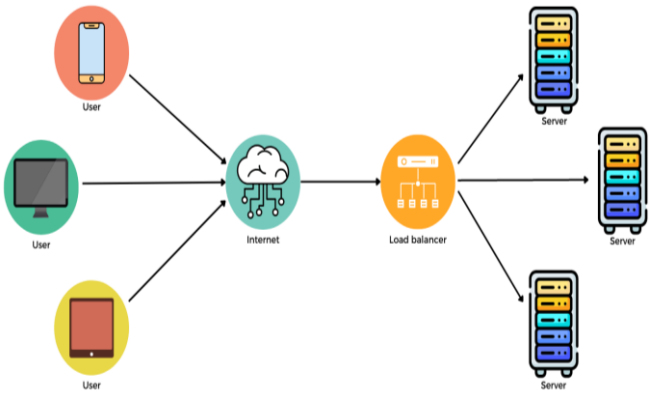

Figure 1. Load balancing

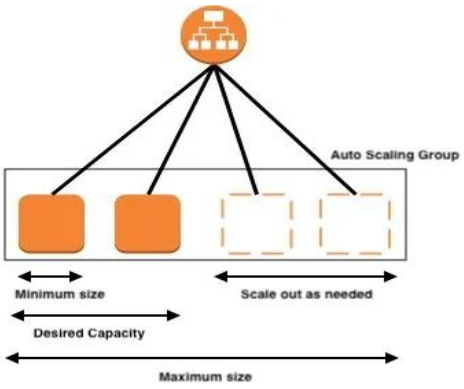

Figure 2. Auto scale

Figure 3 illustrates the GCP-based LMS architecture, integrating load balancing, Managed Instance Groups (MIGs), Compute Engine, Cloud SQL, Redis Memorystore, Cloud Filestore, and Unoconv Job. The Load Balancer distributes incoming traffic and redirects HTTP requests to HTTPS before forwarding requests to the application instances in the MIG. The MIG dynamically adjusts the number of Compute Engine instances based on workload. Cloud SQL provides the database layer, while Redis Memorystore reduces database access through caching. Cloud Filestore provides shared storage for Moodledata, and Unoconv Job handles document conversion processes. This integrated architecture supports scalable and reliable LMS operation under varying workloads.

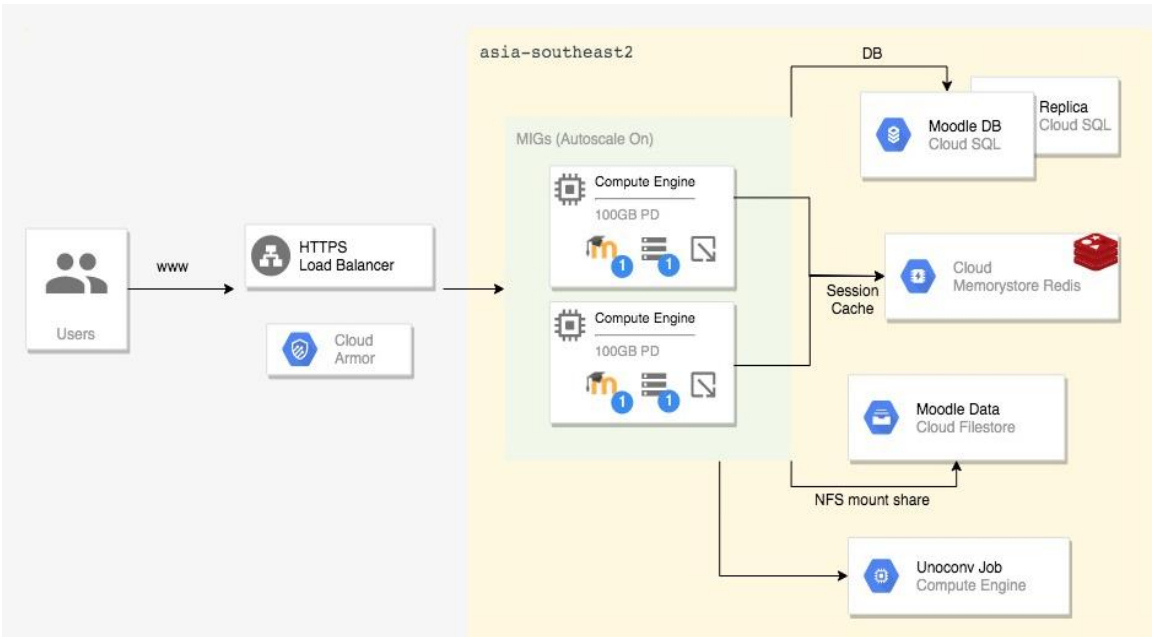

Figure 3. LMS architecture with load balancing and auto scaling

Table 1 summarizes the cloud infrastructure configuration used for the LMS implementation. The Moodle application runs on Compute Engine with 4 vCPUs, 16 GB RAM, and a 50 GB persistent disk, while Cloud SQL provides a MySQL 5.7 database with 4 vCPUs, 8 GB RAM, and 70 GB auto-increasing SSD storage. Redis Memorystore is configured with a 1 GB Basic tier for Moodle caching, and Cloud Filestore provides 1 TB of BASIC_HDD storage for Moodledata. All core services are deployed in the asia-southeast2 region (Singapore). The HTTPS Load Balancer distributes incoming traffic across application instances and redirects HTTP requests to HTTPS, supporting secure and scalable LMS access.

Table 1. Cloud-based LMS infrastructure configuration

Comopnen	CPU	RAM	Persistent disk image	Region	-	-
Compute engine (Moodle APP)	4	16GB	50 GB	asiasoutheast2 a	-	-
Load balancer	Name	Mode	Protocol	Description	-	-
	lb moodle dev http	HTTPS load balancer	HTTP	Untuk redirect HTTP ke HTTPS	-	-
	lb moodle dev HTTPS	Load balancer	HTTPS	Handle traffic	-	-
Filestore (Moodledata)	Tier	Region	Capacity	-	-	-
	BASIC_HDD	asia southeast2 b	1 TB	-	-	-
Cloud SQL (Moodle Database)	CPU	RAM	SSD	Type	Version	Region
	4	8	70 GB (Auto increase)	MySQL	5.7	asiasoutheast2 b
Cloud memorystore redis (Moodle MUC)	Tier	Region	Capacity	Redis version	-	-
	Basic	asia southeast2 b	1 GB	4.0	-	-

2.3. Testing

System testing was conducted to validate the cloud architecture's performance in handling real world workloads (real world load testing). This testing was carried out during the administration of the High School (SMA) Practice Exams throughout the Jakarta Special Capital Region, which utilized the university's LMS platform as the exam medium.

2.3.1. Test Scenarios

The test scenario was designed to simulate extreme peak load conditions. During a single concurrent exam session, 3,000 participants (students) accessed the system simultaneously. These activities included logging in, downloading exam questions, taking interactive quizzes, and submitting answers to the database. These conditions represent a real scalability challenge for the LMS infrastructure. Monitoring results throughout the event showed that the auto scaling mechanism worked very well in responding to the surge in users.

As illustrated in Figure 4, there is a positive correlation between the increase in the number of active users and the addition of computing resources.

- Load detection: as the number of participants began to surge (ramp up) toward 3,000 users, CPU usage on the initial instances was detected to spike sharply, exceeding the specified threshold.
- System response: the auto scaler system responds to this condition in real time by automatically triggering the creation of new instances (servers).
- Service stability: the graph shows that even though user traffic is very high, the workload is successfully distributed to these new instances. This keeps application response times stable and prevents downtime or access failures (crashes) that frequently occur in traditional (non cloud) server infrastructure.

Overall, this test proves that the implemented architecture is capable of handling 3,000 concurrent users stably without significant service disruptions.

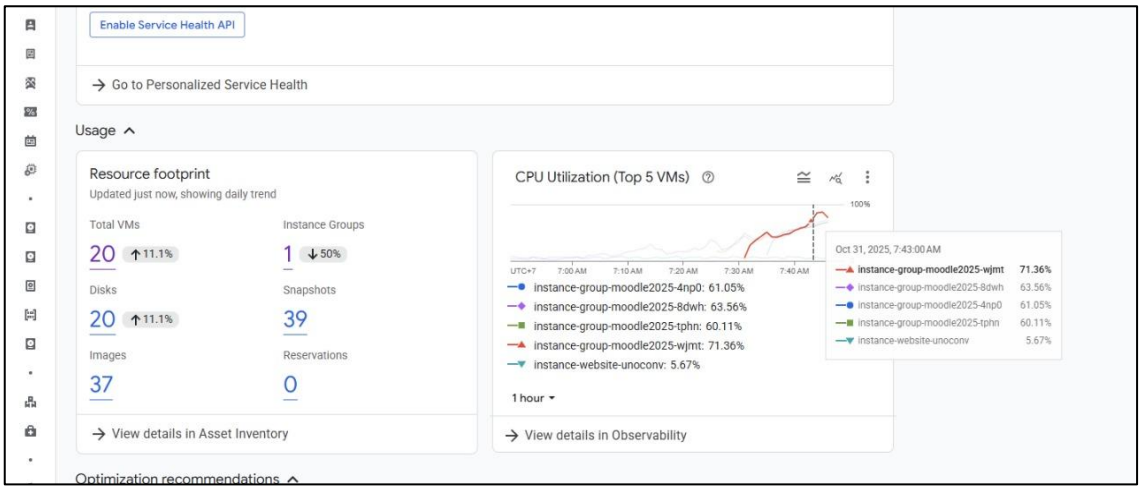

Figure 4. Auto scaling performance chart during a peak load of 3,000 users

2.3.2. Baseline comparison

To validate the effectiveness of the proposed architecture, performance comparisons were conducted between the previous on premises LMS infrastructure and the cloud native LMS implementation on GCP. The comparison focused on:

- response time stability,
- system availability,
- scalability capability,
- concurrent user handling capacity,
- and service interruption frequency during examination sessions.

The experimental results demonstrate that the cloud based architecture significantly improved system stability and reduced the occurrence of service failures during high concurrent access conditions.

3. RESULTS AND DISCUSSION

3.1. Competitor analysis case study at a University

Discussions regarding concurrent users (at any given time) on a university's LMS often pose a critical challenge for IT infrastructure. In traditional infrastructure models using local servers, system capacity is static and limited. When there is a simultaneous surge in access such as during course registration or online exams peak load often exceeds the server's capacity threshold. Figure 5 shows the user monitoring interface in the LMS system.

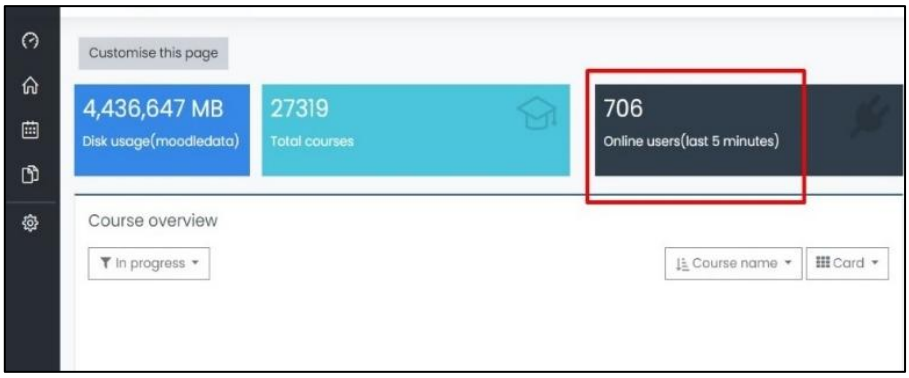

Figure 5. The LMS admin dashboard shows 706 concurrent users online ahead of the exam

Figure 6 demonstrates the effectiveness of the auto scaling mechanism during peak examination sessions. When concurrent user traffic increased significantly, the Managed Instance Group (MIG) automatically provisioned up to five additional virtual machine instances while maintaining 100% instance health status. This behavior indicates that the cloud native architecture was capable of dynamically distributing workloads without causing service interruption. The implementation of auto scaling significantly reduced server overload conditions commonly observed in traditional LMS infrastructures. CPU utilization remained below 40% during peak traffic periods because incoming requests were distributed across multiple active instances rather than being processed by a single server. This horizontal scaling mechanism improved workload balancing and prevented resource bottlenecks.

The transition to cloud technology provides a solution through a two way elasticity mechanism. In a cloud environment, systems no longer rely on a single physical server unit, but rather on a network of virtual resources managed by Managed Instance Groups (MIGs). When load detection indicates an increase in the number of users, Google Cloud Load Balancer will distribute traffic evenly. If the load continues to rise, the auto scaling feature will automatically add new server instances to handle the surge without manual intervention from IT staff, ensuring the load remains safely distributed. Figure 7 shows a spike in concurrent users.

As shown in Figure 8, CPU utilization remained relatively stable even during periods of high concurrent access. The average CPU utilization during peak examination sessions was maintained below 40%, compared to approximately 90–95% in the previous on premises infrastructure. This improvement was achieved through the integration of load balancing and automatic horizontal scaling, which dynamically allocated computing resources based on workload demand.

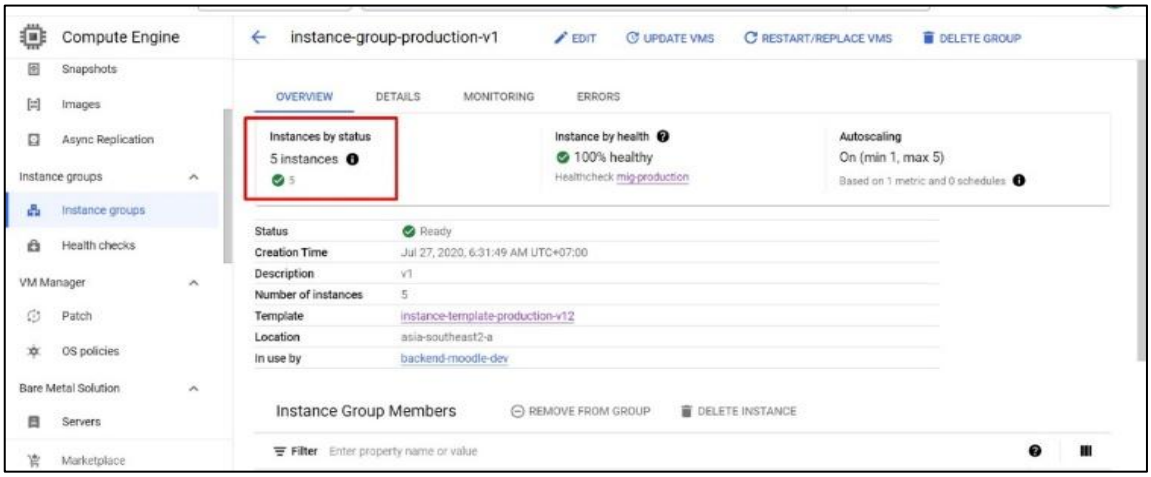

Figure 6. The GCP dashboard successfully auto scaled 5 VMs while they were 100% healthy

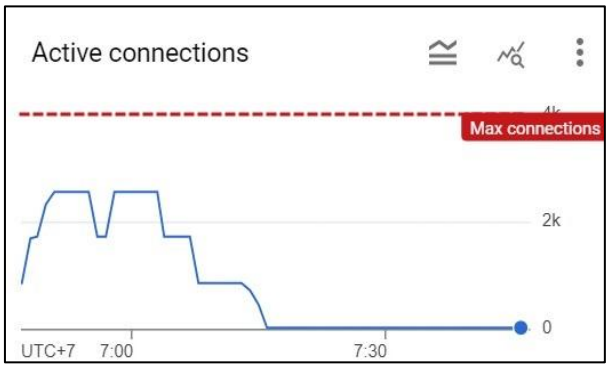

Figure 7. The GCP dashboard showed a surge in concurrent user connections leading up to and at the start of the online exam, which began at 7:00 a.m. WIB

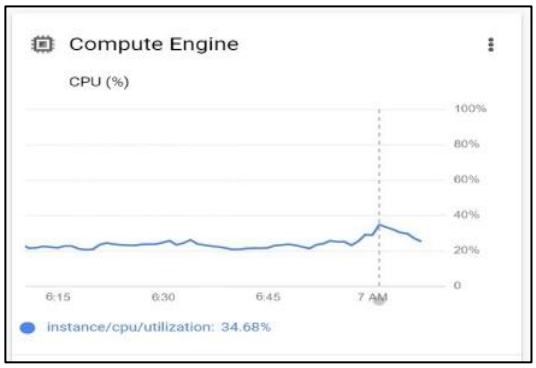

Figure 8. The GCP Dashboard shows that CPU utilization remains below 40% even during spikes in concurrent users

3.2. Scalability

Scalability is a system's ability to handle increasing workloads by adding or adjusting resources. In traditional infrastructure, scalability is often a major obstacle because it is vertical and limited by physical constraints. To increase the capacity of local servers to accommodate more students, universities must procure new hardware, perform physical installations, and reconfigure systems a process that is time consuming and costly. The illustration shows how traditional systems are caught between the demands of a growing student population and rigid local capacity limits. Figure 9 shows an illustration of a traditional server and a cloud-based server.

The scalability results demonstrate a substantial difference between the previous on-premises LMS infrastructure and the proposed cloud-native architecture. The previous infrastructure was capable of supporting approximately 700 concurrent users before experiencing severe response delays and service instability. In contrast, the proposed architecture was experimentally operated under a peak workload of approximately 3,000 concurrent users without significant service disruption. This represents an increase of approximately 328% in concurrent-user handling capacity, as shown in Table 2. The difference between the two environments is primarily related to their resource-scaling characteristics. The on-premises infrastructure depended on a relatively fixed physical capacity, making it difficult to respond immediately to sudden increases in examination traffic. Once the available server resources approached their limits, additional demand resulted in increased response times and a higher probability of service failure. The cloud-native architecture, in contrast, used Managed Instance Groups and auto-scaling to provision additional application instances as workload increased. This allowed computing capacity to expand horizontally rather than relying solely on increasing the resources of a single server. The monitoring results support this interpretation. During the cloud-based examination session, 706 concurrent users were observed in the LMS dashboard

(Figure 5), while the architecture was subsequently validated under a substantially higher workload of 3,000 concurrent users. The distinction between the observed 706-user session and the 3,000-user experimental peak is important because the former represents an actual monitored workload, whereas the latter represents the peak load used to evaluate the scalability of the architecture. Therefore, the 3,000-user result should be interpreted as evidence of the system's tested handling capability rather than as the normal number of concurrent users during the observed examination session.

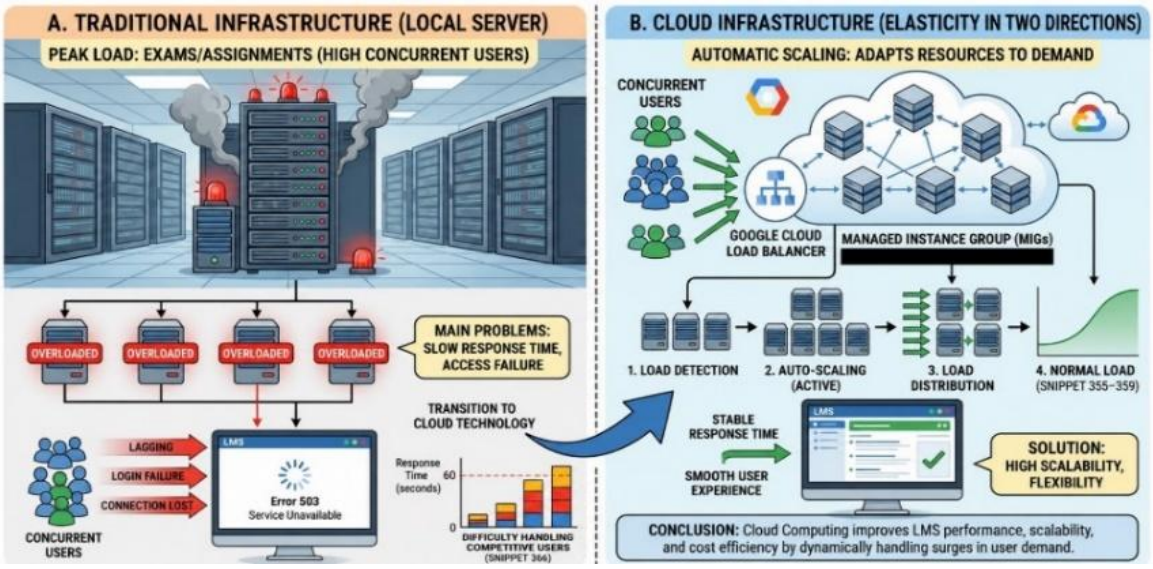

Figure 9. Illustration of the improved scalability of cloud infrastructure compared to traditional infrastructure

The performance metrics in Table 2 further demonstrate the scalability benefit. Average response time decreased from 4.8 s to 1.2 s, representing a 75% reduction. Peak CPU utilization decreased from 95% to 38%, while the error rate decreased from 18.4% to 1.3%. Throughput increased from 120 req/s to 650 req/s, corresponding to an improvement of approximately 441%. These results indicate that the increase in concurrent-user capacity was achieved without a proportional increase in resource saturation or service errors. Redis caching also contributed to the observed performance improvement by storing frequently accessed session data and cache objects in memory. By reducing repeated database queries, this mechanism lowered the processing demand placed on the Cloud SQL backend. However, the results should be interpreted as the combined effect of the cloud-native architecture rather than being attributed to Redis alone, because load balancing, horizontal auto-scaling, database separation, and distributed storage were implemented simultaneously.

Overall, the experimental results demonstrate that the proposed cloud-native LMS architecture provides greater elasticity and high-load handling capability than the previous on-premises infrastructure. The ability to dynamically add application instances enable the system to respond to sudden workload increases without requiring manual hardware expansion. This characteristic is particularly relevant to higher education environments, where extreme LMS workloads are concentrated around examination and other academic activities. The results therefore provide practical evidence that cloud-native infrastructure can improve LMS scalability and service stability under high-concurrency conditions. Table 2 shows a comparison of the performance of traditional LMS and cloud based LMS.

Table 2. Performance comparison between traditional LMS and cloud based LMS

Performance metric	Traditional LMS	Cloud based LMS	Improvement
Max concurrent users	700	3000	+328%
Average response time	4.8 s	1.2 s	75%
CPU utilization peak	95%	38%	60%
Error rate	18.4%	1.3%	92%
Downtime during exams	Frequent	None	Stable
Throughput	120 req/s	650 req/s	+441%

4. CONCLUSION

This study demonstrated that the implementation of a cloud-native LMS architecture using GCP significantly improved system scalability, reliability, and performance under high concurrent user workloads. The proposed architecture integrated load balancer, managed instance groups with auto-scaling, Cloud SQL, Redis Memorystore, and Cloud Filestore to overcome the limitations of traditional on-premises LMS infrastructure. The main contribution of this research lies in the experimental validation of a real-world cloud-based LMS deployment during online examination activities involving more than 3,000 concurrent users. The experimental results showed that the proposed architecture successfully maintained stable system performance, prevented service downtime, and kept CPU utilization below critical thresholds during peak access periods. Compared to the previous traditional infrastructure, the cloud-native implementation demonstrated substantial improvements in scalability, response stability, resource allocation efficiency, and service availability.

The integration of load balancing and horizontal auto-scaling mechanisms enabled the infrastructure to dynamically distribute workloads and automatically provision computing resources based on real-time demand. In addition, the implementation of Redis caching reduced database access overhead and contributed to faster response times during intensive LMS activities. These results confirm that cloud computing provides not only technical advantages but also practical and strategic benefits for higher education institutions conducting large-scale online learning and examination services. This research also provides practical implementation insights for universities seeking scalable and reliable LMS infrastructures. The proposed framework can serve as a reference architecture for educational institutions facing challenges related to concurrent user management, peak-load stability, and digital learning transformation.

Future research may focus on the integration of artificial intelligence-based predictive auto-scaling, container orchestration technologies such as Kubernetes, multi-region cloud deployment for higher fault tolerance, and advanced cybersecurity mechanisms to further improve infrastructure resilience, scalability, and operational security in cloud-native educational systems.

ACKNOWLEDGMENTS

We would like to express our gratitude to the Information Technology Development Agency for facilitating this research.

FUNDING INFORMATION

This research was not funded by any party.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Endy Sjaiful Alim	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Tirta Anhari		✓				✓		✓	✓	✓	✓	✓		
M. Asep Rizkiawan	✓		✓	✓			✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing **O**riginal Draft
E : Writing Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

None of the authors have any conflicts of interest.

DATA AVAILABILITY

Data availability does not apply to this paper because no new data were generated or analyzed in this study.

REFERENCES

[1] S. Desmaryani *et al.*, "The role of digital leadership, system of information, and service quality on e-learning satisfaction," *International Journal of Data and Network Science*, vol. 6, no. 4, pp. 1215–1222, 2022, doi: 10.5267/j.ijdns.2022.6.012.

[2] V. Dhameria, A. I. M. Muazeib, K. M. S. Blhaj, S. Sugiyarsih, and R. A. Rosadah, "The impact of digital transformation in higher education management: integrating online learning and educational applications for efficiency and accessibility," *International Journal of Educational Qualitative Quantitative Research*, vol. 4, no. 1, pp. 15–24, Jul. 2025, doi: 10.58418/ijeqqr.v4i1.135.

[3] M. Alenezi, "Digital learning and digital institution in higher education," *Education Sciences*, vol. 13, no. 1, p. 88, Jan. 2023, doi: 10.3390/educsci13010088.

[4] V. J. García-Morales, A. Garrido-Moreno, and R. Martín-Rojas, "The transformation of higher education after the COVID disruption: emerging challenges in an online learning scenario," *Frontiers in Psychology*, vol. 12, Feb. 2021, doi: 10.3389/fpsyg.2021.616059.

[5] E. S. Alim and H. Jin, "Deployment of cloud computing for higher education using google apps," in *Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017*, IEEE, Nov. 2017, pp. 73–77. doi: 10.1109/ICITISEE.2017.8285563.

[6] S. Brown, L. Murphy, and K. Hammond, "Learning management system adoption by academics: a perspective following the forced lockdown of NZ Universities due to COVID-19 in 2020," *Journal of Open, Flexible and Distance Learning*, vol. 25, no. 2, pp. 55–65, 2022, [Online]. Available: https://jofdl.nz/index.php/JOFDL/article/view/477

[7] I. Maslov, S. Nikou, and P. Hansen, "Exploring user experience of learning management system," *The International Journal of Information and Learning Technology*, vol. 38, no. 4, pp. 344–363, Aug. 2021, doi: 10.1108/IJILT-03-2021-0046.

[8] D. Liu, L. Carter, and J. Lin, "Towards connectivism: exploring student use of online learning management systems during the COVID-19 pandemic," *Online Learning Journal*, vol. 28, no. 2, Jun. 2024, doi: 10.24059/olj.v28i2.4047.

[9] A. K. Alhazmi, A. Imtiaz, F. Al-Hammadi, and E. Kaed, "Success and failure aspects of LMS in E-learning systems," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 11, pp. 133–147, Jun. 2021, doi: 10.3991/ijim.v15i11.20805.

[10] N. N. M. Kasim and F. Khalid, "Choosing the right learning management system (LMS) for the higher education institution context: A systematic review," *International Journal of Emerging Technologies in Learning*, vol. 11, no. 6, pp. 55–61, Jun. 2016, doi: 10.3991/ijet.v11i06.5644.

[11] H. Allam, J. Dempere, F. Kalota, and D. Hua, "Enhancing educational continuity: exploring factors affecting the success of learning management systems in Dubai higher education," *Frontiers in Education*, vol. 9, Jun. 2024, doi: 10.3389/feduc.2024.1382021.

[12] M. N. Ayubi and A. Retnowardhani, "Optimizing learning experiences: a study of student satisfaction with LMS in higher education," *APTISI Transactions on Technopreneurship*, vol. 7, no. 2, pp. 527–541, Jul. 2025, doi: 10.34306/att.v7i2.501.

[13] J. Slamet and Y. Basthomi, "Assessing gamification-based LMS for EFL students: a self-directed learning framework," *Studies in Linguistics, Culture and FLT*, vol. 12, no. 2, pp. 100–122, Aug. 2024, doi: 10.46687/CVHT3942.

[14] K. Samaila, M. N. M. Khambari, J. A. Kumar, and M. Masood, "Factors influencing postgraduate students' intention to use learning management system," *Tuning Journal for Higher Education*, vol. 9, no. 2, pp. 151–176, May 2022, doi: 10.18543/tjhe.2177.

[15] A. Al-Hunaiyyan, S. Al-Sharhan, and R. AlHajri, "Prospects and challenges of learning management systems in higher education," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 12, pp. 73–79, 2020, doi: 10.14569/IJACSA.2020.0111209.

[16] E. B. Tuğtekin, "Scrutinizing learning management systems in practice: an applied time series research in higher education," *International Review of Research in Open and Distributed Learning*, vol. 24, no. 2, pp. 53–71, Mar. 2023, doi: 10.19173/irrodl.v24i2.6905.

[17] T. Raghu, S. M. S. Meenatshi, K. Shaik, D. Biswas, N. Goke, and S. S.Surve, "Cloud-based learning management systems for scalable education," in *2025 World Conference on Cutting-Edge Science and Technology, WCCEST 2025*, IEEE, Sep. 2025, pp. 1–7. doi: 10.1109/WCCEST66994.2025.11389973.

[18] E. S. Alim, M. A. Rizkiawan, and A. Subagyo, "Implementing load balancing on google cloud platform to build an online learning system (in *Indonesian*)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 4, pp. 759–770, 2025.

[19] S. Jung and J. H. Huh, "An efficient LMS platform and its test bed," *Electronics (Switzerland)*, vol. 8, no. 2, p. 154, Feb. 2019, doi: 10.3390/electronics8020154.

[20] H. Fawareh, O. Dahham, H. Aljawawdeh, and E. Al Daoud, "Evaluation of cloud computing for advancement LMS through different environments," *International Journal of Advances in Soft Computing and its Applications*, vol. 16, no. 3, pp. 125–148, 2024, doi: 10.15849/IJASCA.241130.08.

[21] A.-E. Guerrero-Roldán, M. E. Rodríguez-González, D. Bañeres, A. Elasri-Ejjaberi, and P. Cortadas, "Experiences in the use of an adaptive intelligent system to enhance online learners' performance: a case study in Economics and Business courses," *International Journal of Educational Technology in Higher Education*, vol. 18, no. 1, p. 36, Dec. 2021, doi: 10.1186/s41239-021-00271-0.

[22] M. Mosayebi and R. Azmi, "Cost-effective clonal selection and AIS-based load balancing in cloud computing environment," *The Journal of Supercomputing*, vol. 80, no. 16, pp. 23271–23310, Nov. 2024, doi: 10.1007/s11227-024-06324-1.

[23] L. S. Khoshaim, "Adaptive threshold tuning-based load balancing (ATTLB) for cost minimization in cloud computing," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 3, pp. 943–958, 2024, doi: 10.14569/IJACSA.2024.0150394.

[24] G. Bharanidharan and S. Jayalakshmi, "Predictive scaling for elastic compute resources on public cloud utilizing deep learning based long short-term memory," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, pp. 73–81, 2021, doi: 10.14569/IJACSA.2021.0121010.

[25] S. Alharthi, A. Alshamsi, A. Alseiari, and A. Alwarafy, "Auto-scaling techniques in cloud computing: issues and research directions," *Sensors*, vol. 24, no. 17, p. 5551, Aug. 2024, doi: 10.3390/s24175551.

[26] D. Ajiga, P. A. Okeleke, S. O. Folorunsho, and C. Ezeigweneme, "Methodologies for developing scalable software frameworks that support growing business needs," *International Journal of Management & Entrepreneurship Research*, vol. 6, no. 8, pp. 2661–2683, Aug. 2024, doi: 10.51594/ijmer.v6i8.1413.

BIOGRAPHIES OF AUTHORS



Endy Sjaiful Alim received his bachelor's and master's degrees in engineering, establishing a solid foundation in computer and electrical engineering. He currently serves as a senior lecturer at the Faculty of Industrial Technology and Informatics, Universitas Muhammadiyah Prof. Dr. Hamka (Uhamka), Indonesia. Alongside his academic role, he holds a strategic structural position as the Head of the Information Technology Development Agency (BPTI) at the university. His primary research interests encompass cloud computing, computer for education, electrical instrumentation, and information systems. In the current study, he contributed significantly to the conceptualization of the system architecture, framework implementation, and data synthesis, leveraging his expertise in mobile and cloud-based computing technologies. He can be contacted at email: endy@uhamka.ac.id.



Tirta Anhari is a lecturer and researcher at the Faculty of Industrial Technology and Informatics, Universitas Muhammadiyah Prof. DR. HAMKA (UHAMKA), Indonesia. He holds a Bachelor's degree in Engineering (S.T.) and a Master's degree in Computer Science (M.Kom.). Currently, he also serves as the Acting Secretary of the Information Technology Development Agency (BPTI) at UHAMKA. His primary research interests include information systems and technology, business intelligence, and business process management, with a particular focus on data visualization and institutional analytics. He can be contacted at email: tirta.anhari@uhamka.ac.id.



M. Asep Rizkiawan he is a lecturer in Electrical Engineering who was born and raised in Lampung, Indonesia. He pursued his undergraduate education at Prof. Dr. HAMKA Muhammadiyah University majoring in Electrical Engineering. After completing his undergraduate studies, he continued his education to the postgraduate level (S2) at the National Institute of Science and Technology (ISTN) Jakarta, with a field of study that is still in line, namely Electrical Engineering, and successfully earned a Masters in Engineering. He has also completed his professional engineering degree at the Indonesian Institute of Technology and is currently pursuing a Ph.D. in Electrical Engineering at Mercu Buana University. Conducting several researches both accredited Sinta and nationally related to the field. Interested in the field of telecommunications electronics, internet of things and also computers. He can be contacted at email: aseprizkiawan10@gmail.com.