Journal of Information Systems and Informatics
Vol. 7, No. 1, March 2025 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v7i1.993 Published By DRPM-UBD

Enhancing Web Performance for E-learning Platform using Content Delivery Network (CDN) and Varnish Cache

Muhammad Nur Yasir Utomo[1], Eddy Tungadi[2], Widya Khartika[3]

[1,2,3]Department of Informatics and Computer Engineering, State Polytechnic of Ujung Pandang, Makassar, Indonesia
Email: [1]yasirutomo@poliupg.ac.id, [2]eddy.tungadi.poliupg.ac.id, [3]widyakk14@gmail.com

Abstract

Along with the development of technology, the web has become a very popular platform for providing information services and digital content, especially in education sector. The popularity of web services such as e-learning is directly proportional to the increasing number of users. The increase in the number of users is often a problem because it can lead to decreased web performance and potential downtime. To overcome this problem, this study proposes Content Delivery Network (CDN) and Varnish Cache as solutions. Web performance evaluation was carried out in a campus internal network using Apache JMeter with a load of 1,000 users. Based on the evaluation, there was a 175.5% increase in throughput, from 51.9 to 142.9 requests per second. In terms of response time, it improved by 54.3%, decreasing from 16,476 ms to 7,526 ms. Additionally, latency was reduced by 82.4%, from 3,555.8 ms to 624.8 ms. The error rate also decreased from 31.4% to 17.2%. These results indicate that CDN implementation can effectively improve web server performance and provide an optimal user experience, especially under high load conditions.

Keywords: Content Delivery Network, CDN, Varnish Cache, Web Performance, E-learning

1. INTRODUCTION

The rapid development of information technology has driven increasing demand for web-based services, especially in the education sector [1]. The web has now become important for providing real-time information and services, especially on e-learning platforms, which have an important role in education after the Covid-19 pandemic era [2]. As the number of users and web content continues to grow, the demand for web performance also increases [3]. The inability of a web to meet user demands results in a decline in web server performance, negatively impacting user experience and overall system effectiveness [4].

A web server plays a crucial role as the central hub for managing and delivering information to users. It is responsible for processing user requests and responding

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

with website content [5]. However, a standalone web server often encounters challenges in handling high request loads. A significant surge in users, especially in e-learning environments, can overload the server, causing slower access times, increased latency, and a heightened risk of downtime [6]. E-learning platforms, such as Moodle, often face specific performance bottlenecks due to their high number of concurrent users, extensive database queries, and the need to serve large multimedia files. Peak usage periods, such as online exams or mass access to course materials, can overwhelm the web server, leading to slow response times, increased latency, and even service interruptions which can disturb exams of the student. In such situations, the inability of the server to optimally handle all requests becomes a critical issue that needs to be addressed to ensure reliable and efficient web server performance.

Several previous studies have attempted to address this issue by improving web performance. Research by Arun [7] focused on enhancing web performance through file minification. File minification is the process of removing unnecessary elements, such as spaces, comments, and redundant characters from CSS, JavaScript, or HTML files, to reduce the overall file size [8]. This reduction helps improve the loading speed of a website. While minification can contribute to faster load times, its effectiveness diminishes when large files need to be repeatedly downloaded by new users or when content is not cached efficiently [9]. In these cases, the benefits of minification are limited, as the server still needs to handle the heavy load of delivering large files for every request.

Compression techniques have also been explored in previous research. Compression, using algorithms such as Gzip or Brotli [10], helps reduce the size of data transmitted from the server to the user. This method accelerates data transfer by minimizing the amount of data that needs to be sent [11]. However, while compression improves transfer speeds, it does not fully resolve the issue when the web server must handle a sudden surge in a large volume of requests. In these cases, the server may still struggle to efficiently manage the increased load, which can affect overall performance and user experience.

Another research by Dalia [12] applied Load Balancing as a solution. Load balancing is a technique that distributes network traffic across multiple servers to ensure an even workload distribution [13]. This solution is effective in preventing overload on a single server, but its success depends on the number of available servers [14]. In situations with extremely high loads, this approach requires a larger infrastructure investment to maintain optimal performance.

These three techniques can improve website performance, but their limitations become more apparent when the number of users accessing the website increases significantly. In such situations, the web server still faces the risk of overload, leading to slow access times and potential downtime.

A different solution was explored which implemented a Content Delivery Network (CDN) by Gupta and Garg [15]. This research demonstrated that CDN can distribute popular content through surrogate servers, significantly reducing the load on the web server. Gao and Zhu [16] found that the implementation of CDN accelerates access times and improves website scalability, especially when handling a surge in user traffic. Additionally, Yang [1] discover a significant increase in website access speed after the CDN implementation. Study by [17] also revealed that CDN can reduce latency by efficiently caching content, ultimately enhancing the user experience and minimizing the risk of downtime.

Recent studies have shown that CDN can effectively address performance issues on websites with large numbers of users and content. CDN presents a more holistic approach by distributing website content through a network of geographically dispersed servers, meaning that user requests do not always have to be served by the web server [18]. CDN servers act as intermediaries between the web server [19] and users by storing copies of content from the web server [20]. With this approach, CDN not only reduces the load on the web server but also accelerates access times by bringing content closer to the user's location [21]. However, the implementation of CDN in e-learning platforms requires further research to optimize configurations and assess its impact on the overall system performance (throughput, response time, latency, and error rate) which this research aims to address.

Based on this background, this research proposes a novel approach that combines Content Delivery Network (CDN) with Varnish Cache as an HTTP accelerator, specifically tailored for Moodle-based e-learning platforms, an area that has not been extensively explored in existing literature. The performance of the web platform will be evaluated before and after the implementation of the CDN, using Apache JMeter to measure key metrics such as throughput, response time, latency, and error rate. By integrating CDN with Varnish Cache, this research aims to provide a practical solution for optimizing web performance on e-learning platforms, contributing to more reliable and efficient service during high-traffic periods.

This research is structured as follows: Section 2 shows the research method. Section 3 describes the results and analysis. Section 4 concludes this research and the possibilities for further research.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2. METHODS

There are three main stages carried out in this research namely system design, design implementation, and testing. Each stage can be explained as follows.

2.1. System Design

The system design stage is carried out to plan the infrastructure and flow of the CDN system:

2.1.1. Infrastructure

In this stage, the infrastructure design is carried out to ensure that the Content Delivery Network (CDN) system can function optimally in supporting the management and distribution of web content. The designed infrastructure includes key elements such as clients, DNS servers, CDN servers, and web servers.

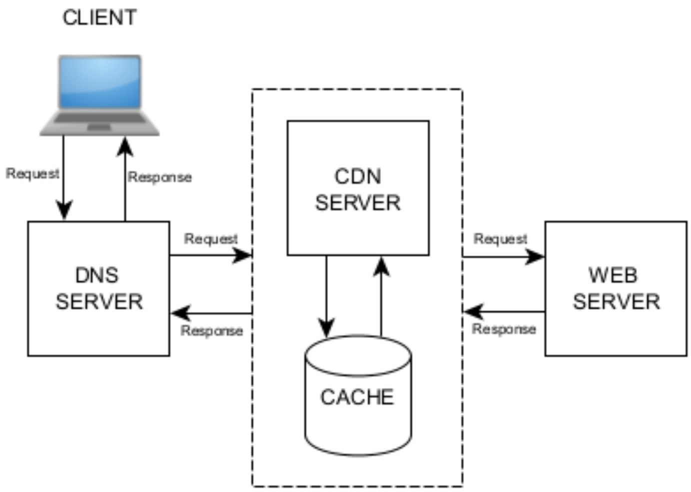

Figure 1. CDN Infrastructure Design

Figure 1 illustrates the data flow from the client request to the response provided by the web server, highlighting the crucial role of the CDN in caching frequently accessed content. The CDN functions as an intermediary that reduces the workload of the web server by distributing copies of content to servers located in various geographic locations. By using this approach, the CDN ensures that clients can access content with lower latency, faster response times, and higher stability, particularly for web serving users on a large scale.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.1.2. CDN Flow

The CDN system flow is designed to ensure efficient content distribution and reduce the workload on the web server. The CDN flow is shown in Figure 2.

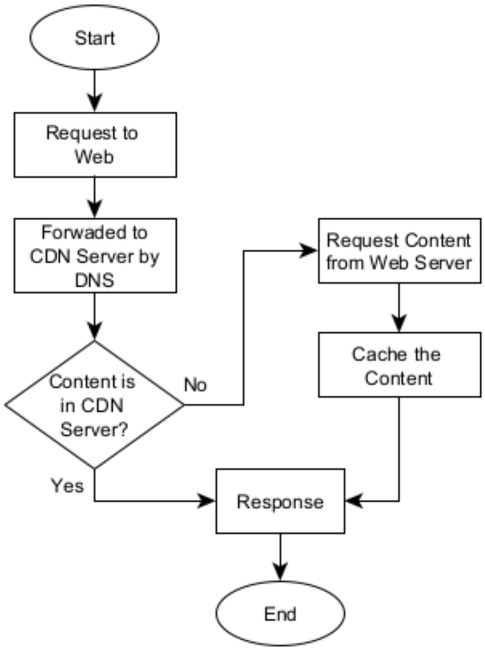

Figure 2. CDN Flow Design

Figure 2 illustrates the CDN process, which begins when the client sends a request to the DNS server. After the DNS server processes and forwards the request to the CDN server, the CDN will check the availability of the requested content in its cache. If the content is available (cache hit), the response is directly provided to the client without forwarding the request to the web server.

Conversely, if the content is not available in the cache (cache miss), the CDN server will forward the request to the web server to retrieve the requested content. Once the content is successfully obtained from the web server, the CDN will store it in the cache for faster response to similar future requests. This flow helps improve content distribution efficiency, accelerate user access times, and optimize the overall system performance.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

2.2. Implementation

The system creation is carried out in Politeknik Negeri Ujung Pandang that uses Proxmox as the virtualization platform. The server is equipped with a 4-core CPU and 8GB RAM, as this specification aligns with the e-learning server infrastructure used in the campus internal network to support large number of concurrent users efficiently. The detail specifications of the Virtual Machine (VM) used are shown in Table 1.

Table 1. Server Specification

Platform	Server	Specification
Proxmox	Web Server (E-learning)	OS: Ubuntu Server 22.04 CPU: 4 Core RAM: 8 GB Disk: 25 GB
	CDN	OS: Ubuntu Server 22.04 CPU: 4 Core RAM: 8 GB Disk: 25 GB

The web server in this case is filled with the Moodle Learning Management System (LMS). Moodle was chosen as the LMS for this research because it is the official platform used by Politeknik Negeri Ujung Pandang. Moreover, Moodle is an open-source LMS which is widely used due to its flexibility and extensive customization options. Additionally, Moodle's modular architecture supports integrations with third-party tools and plugins, which allow it to accommodate various educational needs.

The other server is filled with CDN and Varnish Cache. The server configuration involves several key steps that are carried out sequentially to ensure the system operates according to the needs of the research. These steps include server configuration and CDN configuration:

2.2.1. Server Configuration

Server configuration includes:

1) Installation of Ubuntu Server 22.04: Ubuntu is installed on the prepared server due to its stability and compatibility with Varnish and Apache Web Server.
2) Installation of Apache Web Server: Apache is installed on the main server (VM Web Server), which will act as the Origin Server. Apache will handle HTTP requests that are not cached by the CDN.
3) Installation of Learning Management System Moodle: Moodle is installed as one of the web applications on the Apache server [22], [23].

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

4) Installation of Database Server: MySQL is used to store the database for Moodle.

2.2.2. CDN Configuration

CDN configuration includes:

1) Varnish Cache Installation: Varnish is installed on the VM CDN Server and configured to cache content from the main web server.
2) Backend Configuration: Varnish is configured to set the Apache web server as the backend server. This backend is the source of content that will be cached by Varnish [24].
3) DNS Server: DNS redirection is set up to ensure that all user requests are first directed to the CDN server (Varnish). This configuration is done on the DNS server, which maps the domain to the IP address of the CDN server. If the CDN does not have the requested content in cache, the request will be forwarded to Apache (origin server).

2.3. Evaluation

After the CDN system is set up, testing is carried out to evaluate the web performance. The tests compare the performance of the web without CDN and with CDN. These tests are conducted with scenarios involving 10, 50, 100, 200, 300, 500, and 1000 concurrent users. The measurement tool used in this research is Apache JMeter. The parameters measured include:

1) Throughput: The number of requests processed per second [25].
2) Response Time: The time taken by the server to respond to a request [26], [27].
3) Latency: Measuring the delay that occurs in data transmission [28].
4) Error Rate: The percentage of requests that failed [29], [30].

After the testing is conducted, the CDN implementation is expected to improve throughput, reduce response time, decrease latency, and lower the error rate, especially under high load conditions.

3. RESULTS AND DISCUSSION

3.1. Evaluation of Throughput

The CDN was evaluated in terms of Throughput to see how many requests the web can handle in a second. The results of the evaluation are shown in Table 2.

p-ISSN: 2656-5935 http://journal-isi.org/index.php/isi e-ISSN: 2656-4882

Table 2. Evaluation of Throughput

Users	Throughput (req/sec)	
	Without CDN	With CDN
10	10.7	47.3
50	38.1	57.6
100	52.6	83.5
200	45.2	87.6
300	34.8	110.5
500	26.3	121.1
700	34.6	123.3
1000	51.9	142.9

The throughput analysis shows a significant improvement when using a CDN compared to operating without one. At low user loads (10-50 users), throughput increases by up to 400% with the CDN, demonstrating its effectiveness in reducing latency and improving system efficiency. As user loads increase to 100-300 users, the CDN continues to provide consistent performance, improving throughput by up to 217%, while the non-CDN setup experiences noticeable performance degradation as show in Figure 3.

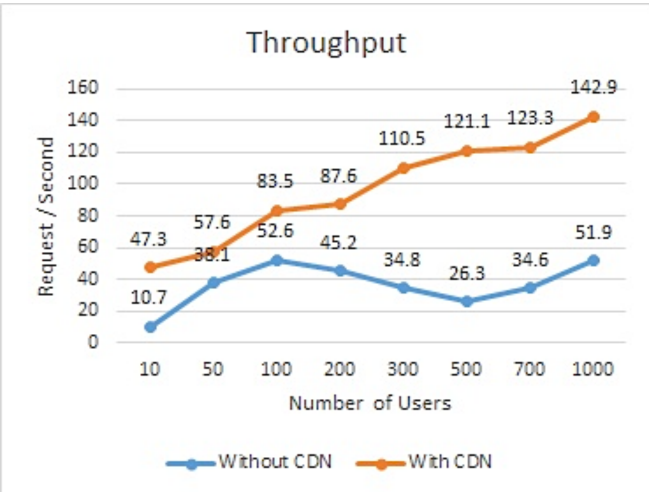

Figure 3. Throughput Comparison Chart

For high user loads (500-1000 users), the CDN demonstrates its scalability by maintaining throughput up to 142.9 req/sec, compared to a maximum of 51.9 req/sec without the CDN. This highlights the CDN's ability to handle traffic spikes by offloading requests to distributed cache servers, preventing bottlenecks at the origin web server.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.2. Evaluation of Response Time

The CDN was evaluated in terms of Response Time to see how fast the web can response. The results of the evaluation are shown in Table 3.

Table 3. Evaluation of Response Time

Users	Response Time (ms)	
	Without CDN	With CDN
10	1173	132
50	2082	706
100	2899	985
200	6140	2072
300	8154	3842
500	11881	4884
700	14839	5830
1000	16476	7526

The response time result shows a clear performance improvement when using a CDN compared to operating without one. At low user loads (10-50 users), response times are reduced by up to 90% with the CDN, highlighting its ability to significantly lower latency and improve user experience. As the user load increases to 100-300 users, the CDN continues to provide noticeable improvements, reducing response times by up to 66%, while the non-CDN setup sees a sharp rise in latency, indicating strain on the server as shown in Figure 4.

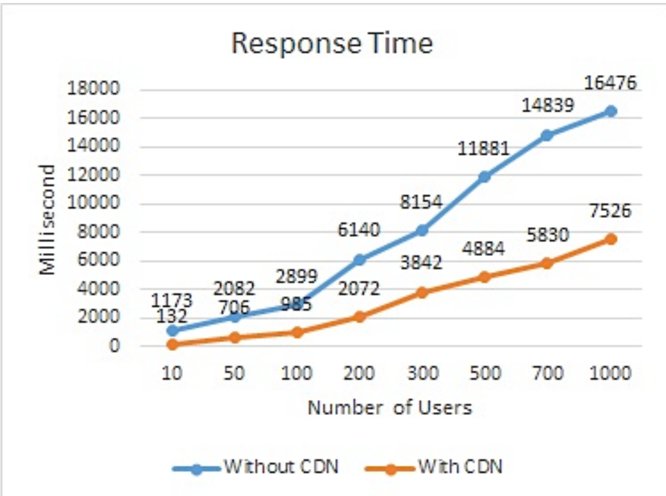

Figure 4. Response Time Comparison Chart

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

For higher user loads (500-1000 users), the CDN remains effective, maintaining lower response times even as the load increases. The response time with the CDN only rises moderately, from 4884 ms at 500 users to 7526 ms at 1000 users. In contrast, response times increased drastically without a CDN, reaching 16476 ms at 1000 users. This demonstrates the CDN's ability to scale efficiently and mitigate the impact of high traffic, reducing the risk of server overload and latency spikes.

3.3. Evaluation of Latency

The CDN was evaluated in terms of Latency to see the delay between a user's request and the server's response. The results of the evaluation are shown in Table 4. The latency result demonstrates a clear advantage in using a CDN compared to operating without one, particularly as the number of users increases.

Table 4. Evaluation of Latency

Users	Latency (ms)	
	Without CDN	With CDN
10	202.8	40.7
50	527.0	263.6
100	958.8	170.7
200	1139.8	462.2
300	1575.3	797.9
500	2253.4	1199.4
700	3543.5	1652.9
1000	3555.8	624.8

At low user loads (10 users), the latency drops from 202.8 ms without the CDN to 40.7 ms with the CDN, a significant improvement of about 80%. This highlights the CDN's effectiveness in reducing the time it takes for the web to respond, providing much faster experience for users.

As the user load increases to 50-100 users, the CDN continues to deliver lower latency, reducing response times by up to 82%. For instance, at 100 users, the latency drops from 958.8 ms without the CDN to just 170.7 ms with the CDN. However, as traffic increases to 200-300 users, the performance improvement, though still significant, becomes less dramatic. At 300 users, the latency is 797.9 ms with the CDN compared to 1575.3 ms without the CDN, a reduction of about 50% as shown in Figure 5.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

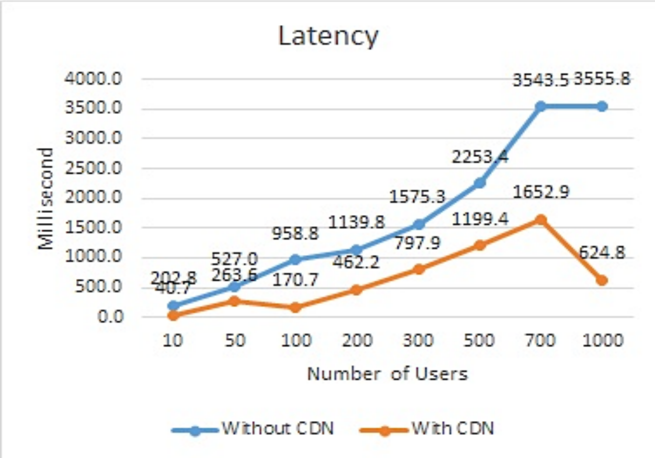

Figure 5. Latency Comparison Chart

For very high user loads (500-1000 users), the CDN continues to provide a notable reduction in latency, but the performance gap begins to shrink. At 1000 users, the latency is reduced from 3555.8 MS without the CDN to 624.8 MS with the CDN, which is still a 82.4% improvement. This data shows that the CDN maintains its effectiveness in reducing latency even under heavy traffic.

3.4. Evaluation of Error Rate

The CDN was evaluated in terms of Error Rate to assess its impact on the reliability and stability of the web service. Error rate refers to the proportion of requests that result in errors, such as 404 (Not Found) or 500 (Internal Server Error), which can significantly affect user experience and website performance. A lower error rate indicates better system stability, while a higher error rate suggests potential issues in handling user requests. The results of this evaluation are shown in Table 5.

Table 5. Evaluation of Error Rate

Users	Error Rate (%)	
	Without CDN	With CDN
10	6.7	0.0
50	4.7	0.7
100	13.3	1.3
200	11.4	1.5
300	10.3	1.6
500	18.5	1.3
700	23.4	4.7
1000	31.4	17.2

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

The error rate result reveals a significant improvement in reliability when using a CDN compared to operating without one. At low user loads (10 users), the error rate drops from 6.7% without the CDN to 0.0% with the CDN, indicating that the CDN helps eliminate errors and ensures a more stable experience for users. This trend continues as the user load increases, with the error rate staying considerably lower with the CDN across all user loads as shown in Figure 6.

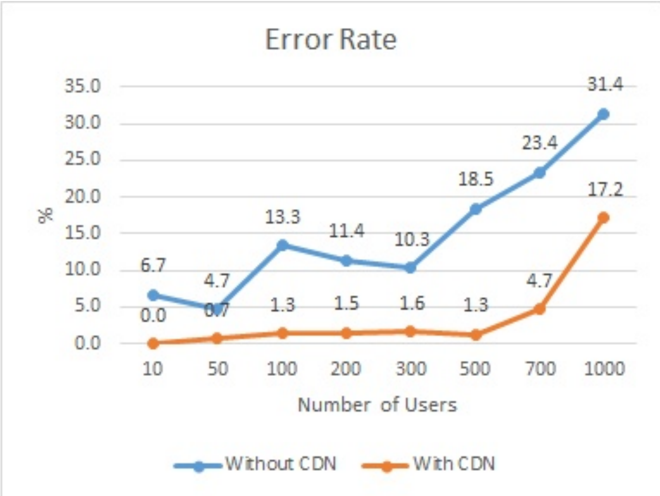

Figure 6. Error Rate Comparison Chart

As the user count increases to 50-100 users, the error rate remains substantially lower with the CDN. For example, at 100 users, the error rate without the CDN is 13.3%, whereas with the CDN, it reduces to just 1.3%. This suggests that the CDN effectively mitigates errors, even under moderate traffic levels. However, as the user load continues to rise to 200-500 users, the difference in error rates between the two setups starts to decrease slightly, although the CDN still performs better. At 500 users, the error rate without the CDN reaches 18.5%, while the CDN maintains a much lower rate of 1.3%.

For very high user loads (700-1000 users), while the CDN still performs better, the error rate increases noticeably, with a peak of 17.2% at 1000 users. Without the CDN, the error rate reaches 31.4% at the same user count. This shows that while CDN can greatly reduce errors, its ability to handle very high traffic is not limitless, and there may be some increase in error rates under extreme loads. Nevertheless, the CDN significantly improves system stability and error handling, especially when compared to a non-CDN setup.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3.5. Discussion

The significant enhancements in web performance achieved through the implementation of CDN and Varnish Cache have critical implications for student learning outcomes and engagement within digital learning environments. One of the key factors determining the success of e-learning platforms is their ability to deliver a smooth and responsive user experience. The recorded improvements including a 175.3% increase in throughput, a 54.3% reduction in response time, and an 82.4% decrease in latency signal a more robust, scalable infrastructure that can reliably support high user volumes without service degradation. This is particularly crucial during peak usage periods when learning activities, live sessions, and assessments occur concurrently.

Throughput analysis reveals dramatic improvements in request handling capabilities. Without a CDN, the system struggled to maintain consistent performance beyond 100 users, while the CDN-enhanced setup scaled effectively, reaching up to 142.9 requests per second at 1000 users. This scalability ensures that e-learning platforms can accommodate sudden traffic surges, preventing bottlenecks at the origin server and maintaining uninterrupted access for all users [31].

Response time data further demonstrates the CDN's impact. At low user loads (10–50 users), response times decreased by up to 90%, and even under heavier loads (500–1000 users), performance remained stable, with the CDN consistently reducing wait times by over 50%. For students, faster response times translate into smoother navigation, instant page loading, and more reliable access to multimedia content. These factors are vital in sustaining learner motivation, reducing frustration, and enhancing retention rates [32].

Latency evaluations underscore the CDN's ability to minimize communication delays. With reductions reaching 82.4% at peak loads, students benefit from real-time responsiveness, which is especially advantageous during live lectures, interactive discussions, or assessments requiring immediate feedback. By ensuring low latency, the CDN fosters an environment that supports seamless interaction, promoting active engagement and knowledge retention [31].

In terms of system stability, the drop-in error rates from 31.4% to 17.2% at 1000 users is significant. The CDN effectively mitigates errors caused by traffic overloads, keeping the platform operational even under extreme load conditions. This stability is essential for maintaining learner trust and ensuring that critical learning tasks like accessing resources or submitting assignments—are completed without disruption. Error-free operations are closely tied to student satisfaction and academic performance [33].

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Altogether, these performance improvements directly support digital learning success. By minimizing delays, reducing errors, and ensuring responsive service delivery, CDN and Varnish Cache technologies enhance the accessibility, reliability, and usability of e-learning platforms. This optimization plays a direct role in reducing dropout rates, increasing course completion, and promoting learner persistence. Moreover, institutions deploying these technologies demonstrate a commitment to high-quality digital education, ensuring that technical limitations do not impede student success.

In summary, the CDN's contribution to throughput, response time, latency, and error reduction highlights its central role in building a responsive and reliable e-learning infrastructure. These improvements not only align with technical performance metrics but also significantly influence educational engagement and learning outcomes, making CDN integration an essential strategy in modern digital learning ecosystems [31], [32], [33].

4. CONCLUSION

The analysis of the CDN and Varnish Cache's impact on web performance demonstrates significant benefits in terms of throughput, reducing response times, latency, and error rates, particularly under varying user loads. For low to moderate traffic, the CDN and Varnish Cache consistently provide substantial performance improvements, with a throughput increase of 175.5%, response times decreasing by 54.3%, latency reduced by 82.4%, and error rates dropping from 31.4% to 17.2%. Even under high traffic conditions, the CDN and Varnish Cache maintain scalability and reliability, ensuring improved performance compared to a non-CDN setup. However, at extreme user loads, the performance benefits begin to taper, with increasing latency and error rates, highlighting areas where the efficiency of this combination may be further optimized. Future research could explore a more comprehensive comparison between CDN and Varnish Cache with alternative optimization techniques, such as edge computing or AI-driven adaptive caching mechanisms. Additionally, further investigation into the impact of these technologies on server costs and environmental sustainability due to reduced server load would provide valuable insights. Another potential research direction is the inclusion of CPU and RAM usage during testing to better understand the resource efficiency of CDN implementations under different traffic conditions.

REFERENCES

[1] H. Yang, H. Pan, and L. Ma, "A Review on Software Defined Content Delivery Network: A Novel Combination of CDN and SDN," *IEEE Access*, vol. 11, pp. 43822–43843, 2023, doi: 10.1109/ACCESS.2023.3267737.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

[2] M. Ghaznavi et al., “Content Delivery Network Security: A Survey,” *IEEE Commun. Surveys Tuts.*, vol. 23, no. 4, pp. 2166–2190, 2021, doi: 10.1109/COMST.2021.3093492.
[3] S. R. Chanthati, “Website Visitor Analysis & Branding Quality Measurement Using Artificial Intelligence,” *Am. J. Educ. Technol.*, vol. 3, no. 3, pp. 88–101, Aug. 2024, doi: 10.54536/ajet.v3i3.3212.
[4] M. N. Y. Utomo, A. Bastian, and A. Winursito, “Improving Speed Performance of Select Random Query in SQL Database,” *INTEK: J. Penelit.*, vol. 7, no. 1, pp. 26–31, 2020, doi: 10.31963/intek.v7i1.1536.
[5] A. R. Ahmad et al., “A Systematic Review of Web-Based Learning in Enhancing Visualization Skill,” *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 50, no. 2, pp. 119–142, Aug. 2025, doi: 10.37934/araset.50.2.119142.
[6] R. Hanafi, A. Haq, and N. Agustin, “Comparison of Web Page Rendering Methods Based on Next.js Framework Using Page Loading Time Test,” *Teknika*, vol. 13, no. 1, pp. 102–108, Mar. 2024, doi: 10.34148/teknika.v13i1.769.
[7] A. Gopalakrishna, “SEO-Oriented Web Applications: A Comprehensive Guide,” *Int. J. Comput. Eng. Technol.*, vol. 15, no. 4, pp. 814–828, 2024, doi: 10.5281/zenodo.13472461.
[8] A. Bogdan and I. Malavolta, “An Empirical Study on the Impact of CSS Prefixes on the Energy Consumption and Performance of Mobile Web Apps,” in *Proc. IEEE/ACM 11th Int. Conf. Mobile Softw. Eng. Syst. (MOBILESoft)*, Apr. 2024, pp. 12–21, doi: 10.1145/3647632.3647989.
[9] T. Khan Mohd et al., “Comparative Analysis on Various CSS and JavaScript Frameworks,” *J. Softw.*, vol. 17, no. 6, pp. 282–291, 2022, doi: 10.17706/jsw.17.282-291.
[10] B. Wollmer et al., “The Case for Cross-entity Delta Encoding in Web Compression (Extended),” *J. Web Eng.*, vol. 22, no. 1, pp. 131–146, 2023, doi: 10.13052/jwe1540-9589.2217.
[11] M. Trevisan et al., “Measuring HTTP/3: Adoption and performance,” in *Proc. 19th Mediterranean Commun. Comput. Netw. Conf. (MedComNet)*, 2021, doi: 10.1109/MEDCOMNET52149.2021.9501274.
[12] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, “Load balancing techniques in cloud computing environment: A review,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1727–1751, Jul. 2022, doi: 10.1016/j.jksuci.2021.02.007.
[13] N. Singh et al., “Load balancing and service discovery using Docker Swarm for microservice based big data applications,” *J. Cloud Comput.*, vol. 12, no. 1, Dec. 2023, doi: 10.1186/s13677-022-00358-7.
[14] B. Gupta, P. Mittal, and T. Mufti, “A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services,” in *Proc. 2nd Int. Conf. ICT Digit., Smart, Sustain. Dev. (ICIDSSD)*, Mar. 2021, doi: 10.4108/eai.27-2-2020.2303255.

[15] M. Gupta and A. Garg, "Optimizing and load balancing for flash crowd to improve quality of service in content delivery network," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 11, pp. 2568–2574, Sep. 2019, doi: 10.35940/ijitee.K1868.0981119.
[16] L. Gao and X. Zhu, "ICN-Based Enhanced Content Delivery for CDN," *Future Internet*, vol. 15, no. 12, Dec. 2023, doi: 10.3390/fi15120390.
[17] V. Sathiyamoorthi et al., "An intelligent web caching system for improving the performance of a web-based information retrieval system," *Int. J. Semant. Web Inf. Syst.*, vol. 16, no. 4, pp. 26–44, Oct. 2020, doi: 10.4018/IJSWIS.2020100102.
[18] M. T. Sultan and H. El Sayed, "QoE-Aware Analysis and Management of Multimedia Services in 5G and Beyond Heterogeneous Networks," *IEEE Access*, vol. 11, pp. 77679–77688, 2023, doi: 10.1109/ACCESS.2023.3298556.
[19] M. Taha and A. Ali, "Redirection and Protocol Mechanisms in Content Delivery Network-Edge Servers for Adaptive Video Streaming," *Appl. Sci.*, vol. 13, no. 9, May 2023, doi: 10.3390/app13095386.
[20] J. Murillo et al., "CDN-Shifter: Leveraging Spatial Workload Shifting to Decarbonize Content Delivery Networks," in *Proc. ACM Symp. Cloud Comput.*, Nov. 2024, pp. 505–521, doi: 10.1145/3698038.3698516.
[21] A. S. George and A. S. H. George, "The Evolution of Content Delivery Network: How it Enhances Video Services, Streaming, Games, e-commerce, and Advertising," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 10, no. 7, pp. 2395–2402, 2021, doi: 10.5281/zenodo.6788660.
[22] A. David et al., "Scalability through Distributed Deployment for Moodle Learning Management System," *Procedia Comput. Sci.*, vol. 213, pp. 34–41, 2022, doi: 10.1016/j.procs.2022.11.145.
[23] C. B. Mpungose and S. B. Khoza, "Postgraduate Students' Experiences on the Use of Moodle and Canvas Learning Management System," *Technol. Knowl. Learn.*, vol. 27, no. 1, Mar. 2022, doi: 10.1007/s10758-020-09475-1.
[24] N. Savani, "The Future of Web Development: An In-depth Analysis of Micro-Frontend Approaches," *Int. J. Comput. Trends Technol.*, vol. 71, pp. 65–69, 2023, doi: 10.14445/22312803/IJCTT-V71I11P109.
[25] M. N. Y. Utomo, "Sistem Manajemen Konferensi Ilmiah Berbasis Web Menggunakan Metode Pengembangan Waterfall," *J. Tek. Elekterika*, vol. 19, no. 1, p. 1, 2022, doi: 10.31963/elekterika.v6i1.2751.
[26] M. N. Y. Utomo, M. Sudaryanto, and K. Saddhono, "Tools and Strategy for Distance Learning to Respond COVID-19 Pandemic in Indonesia," *Ingénierie des Systèmes d'Information*, vol. 25, no. 3, pp. 383–390, 2020, doi: 10.18280/isi.250314.
[27] A. Anwyl-Irvine et al., "Realistic precision and accuracy of online experiment platforms, web browsers, and devices," *Behav. Res. Methods*, vol. 53, no. 4, pp. 1407–1425, 2021, doi: 10.3758/s13428-020-01501-5.

[28] A. Mubariz et al., "Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang)," in *Seminar Nas. Tek. Elektro dan Informatika (SNTEI)*, 2020, pp. 72–77.

[29] M. N. Y. Utomo, I. Irmawati, and R. Nur, "Pengembangan Sistem Survei Tracer Study Berbasis Web Menggunakan Arsitektur Model View Controller (MVC)," *Inspir.: J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, p. 166, 2021, doi: 10.35585/inspir.v11i2.2633.

[30] A. C. Barus et al., "Performance Testing and Optimization of DiTenun Website," *J. Appl. Sci. Eng. Technol. Educ.*, vol. 4, no. 1, pp. 45–54, Jun. 2022, doi: 10.35877/454ri.asci841.

[31] M. Liu and D. Yu, "Towards intelligent E-learning systems," *Educ. Inf. Technol.*, vol. 28, no. 7, pp. 7845–7876, Jul. 2023, doi: 10.1007/s10639-022-11479-6.

[32] A. E. Khedr, A. M. Idrees, and R. Salem, "Enhancing the e-learning system based on a novel tasks' classification load-balancing algorithm," *PeerJ Comput. Sci.*, 2021, doi: 10.7717/peerj-cs.669.

[33] C. N. Akpen et al., "Impact of online learning on student's performance and engagement: a systematic review," *Discov. Educ.*, vol. 3, no. 1, p. 205, Nov. 2024, doi: 10.1007/s44217-024-00253-0.