Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2024. Vol. 4. No. 1; e-ISSN: 2808-4519

Optimizing Web-Based Survey Applications with Laravel and Cloud Computing

Nanda Saputri[1], Safriadi[2*], Zaharatul Dahlia[3], Rahmadani[4]

[1,2] Jurusan Teknologi Informasi Dan Komputer Politeknik Negeri Lhokseumawe
[3] Jurusan Teknik Elektro Politeknik Negeri Lhokseumawe
[4] Sistem Komputer Universitas Pembangunan Panca Budi
safriadi@pnl.ac.id

Abstract

This study aims to optimize the performance of a web-based survey application through the implementation of the Laravel framework and cloud computing technology. Survey applications often face challenges regarding response speed and scalability as the number of users increases. In this research, performance testing of the application was conducted both before and after optimization. The test results show that the application's response time significantly decreased after the implementation of Laravel and cloud technology. Before optimization, the average response time reached 4.5 seconds at 1,000 users, while after optimization, it dropped to 2.2 seconds. This performance improvement was achieved through efficient caching implementation, database query optimization using Eloquent ORM, and balanced load distribution via load balancers. Additionally, the application's availability increased to 99.9% thanks to cloud features such as auto-scaling and data replication. The conclusion of this study indicates that the combination of Laravel and cloud computing technology is effective in enhancing the performance and reliability of web-based survey applications, providing practical guidance for other web application developers facing similar challenges.

Keywords: *Cloud Computing, Laravel, Optimization*

1. Introduction

In today's digital era, web-based survey applications have become essential tools for many organizations to collect data and feedback from users efficiently and effectively. However, the main challenges faced by web-based survey applications are performance and scalability, particularly as the number of respondents grows significantly. Cloud computing technology emerges as a potential solution to address these issues, offering flexible scalability, high availability, and the ability to handle large workloads more efficiently.

Performance and scalability issues in web-based survey applications are not new. Previous research has shown that on-premise servers often struggle with sudden traffic surges, which can lead to data processing delays, downtime, and a poor user experience. Furthermore, the cost of maintaining physical infrastructure adds an additional burden for organizations. By leveraging cloud computing technology, survey applications can dynamically allocate resources as needed, which not only enhances performance but also optimizes operational costs. Previous studies have explored the benefits of cloud computing in various web applications. Research by Anton on the use of cloud computing technology in improving teaching and learning processes demonstrated cost savings in infrastructure, particularly in server provisioning, as multiple virtual servers can be created on a single physical server [1]. Meanwhile, another study by Pitriyani explained that cloud utilization for database management can simplify database administration for applications [2].

Another critical issue in implementing cloud computing is data security and user privacy. Since survey data often contains sensitive information, organizations must ensure that the selected cloud solution includes adequate security features to protect data from cyber threats. Moreover, compliance with data privacy regulations, such as GDPR, is an essential aspect that must be considered in the design and management of cloud-based survey applications.

This study aims to fill the existing gap by exploring how cloud computing technology can be used to optimize the performance and scalability of web-based survey applications. Through case studies and comparative analysis, this research will evaluate various technical and non-technical aspects of cloud computing implementation and identify best practices that survey application developers can adopt to achieve optimal and efficient performance. Thus, this study is expected to make a significant contribution to the development of more reliable and cost-effective web-based survey technology.

2. Literature Review

2.1. Cloud Computing

Cloud computing is a computing model that enables easy, on-demand access to a configurable pool of computing resources (e.g., networks, servers, storage, applications, and services) that can be quickly allocated and released with minimal management effort or provider interaction. This model includes various services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud computing offers several key characteristics, including on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [3], [4], [5].

One of the main advantages of cloud computing is cost efficiency. Users do not need to make significant capital investments to purchase and maintain hardware. Instead, cloud computing costs are typically operational, allowing companies to allocate their budgets more efficiently. Additionally, cloud service providers often offer various levels of service, from basic infrastructure to complete software solutions, which can be customized to meet user needs.

Cloud computing also supports enhanced collaboration, as data and applications can be accessed from anywhere, anytime. This capability enables geographically dispersed teams to work together more effectively, share information in real time, and increase productivity. Furthermore, cloud services often come with integrated collaboration and communication tools, such as shared storage, simultaneous document editing, and video conferencing [5], [6].

Security and infrastructure maintenance are also the responsibility of cloud service providers, who generally uphold high security standards and employ experienced technical teams. Cloud providers implement various measures to protect user data, including data encryption, continuous security monitoring, and compliance with industry standards and regulations [7].

2.2. Laravel

Laravel is one of the most popular PHP frameworks widely used in web application development. Initially released by Taylor Otwell in 2011, Laravel offers an elegant and clean syntax, along with a range of features that simplify the development of robust and scalable web applications. The framework adopts an MVC (Model-View-Controller) architecture, which separates application logic, user interface, and data, enabling more structured and manageable development [8], [9].

Laravel includes a variety of built-in features, such as a flexible routing system, Eloquent ORM (Object-Relational Mapping) for database interactions, a strong database migration system, and various development tools like the Artisan CLI. Additionally, Laravel supports security features such as CSRF (Cross-Site Request Forgery) protection, input validation, and data encryption, which are crucial for maintaining the integrity and security of web applications [8].

In the context of cloud computing, Laravel is highly compatible with cloud services like AWS, Google Cloud, and Microsoft Azure. Features such as queueing, caching, and job scheduling in Laravel can be easily integrated with cloud services to enhance the performance and scalability of web applications. Thus, Laravel is an ideal choice for developers looking to leverage cloud computing technology to build efficient and reliable web-based survey applications. With these advantages, Laravel not only accelerates the development process but also ensures that the resulting applications are high-quality and ready for large-scale deployment [10].

Your paper must be in one column format with a space of 0.5cm between columns.

3. Research Methodology

3.1. System Development Stages

The research methodology for the study on *Optimization of Web-Based Survey Applications Using Laravel and Cloud Computing* is illustrated in Figure 1.

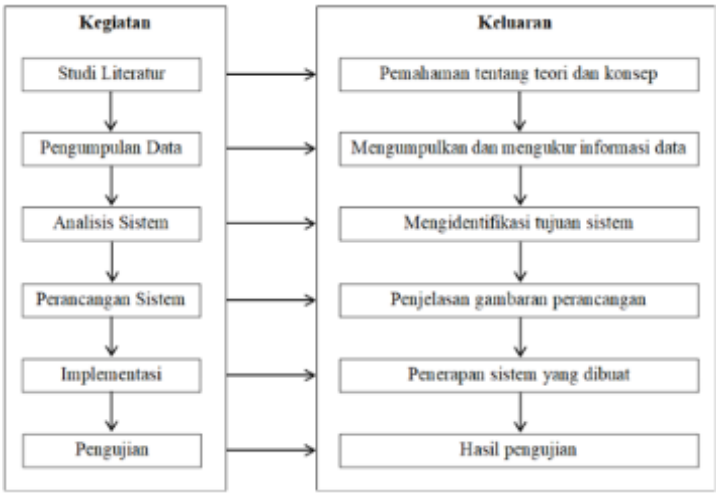

Fig. 1: Research Stages

The research methodology, as illustrated in Figure 1, is discussed as follows: The first stage of this study is a literature review aimed at gathering comprehensive information and insights on the research topic. Researchers review various academic sources, such as journals, books, and scientific publications, that are relevant to cloud computing, the Laravel framework, and web-based survey applications.

Through the literature review, the researchers identify gaps in previous research, find relevant theories, and gain insights into methodologies applicable to this study.
The next stage is data collection, where the researchers gather necessary data to support the research. Data can be obtained through methods such as surveys, interviews, observations, or document analysis. In this context, data may include technical information on the performance of web-based survey applications, user feedback, and data related to the usage and performance of cloud computing services and the Laravel framework. Following data collection is the system analysis stage. Here, researchers analyze the collected data to understand existing issues, assess system needs, and determine the functional and non-functional specifications of the survey application to be developed. This analysis includes identifying performance and scalability issues that need to be addressed through the use of cloud computing and the Laravel framework.
The system design stage involves creating a detailed design of the web-based survey application. This includes system architecture, database design, user interface, and application workflow. Researchers also determine how cloud computing and Laravel will be integrated to optimize application performance and scalability. At this stage, prototyping or mockups of the application may be developed to gather initial feedback before implementation. Implementation is the phase where the system design is translated into functioning code. The researchers use the Laravel framework to develop the web-based survey application and integrate cloud computing services as needed. This phase involves coding, server configuration, and setting up the required cloud infrastructure. Researchers ensure that each system component functions according to the specifications established in the design phase.
The final stage in this research is testing, which aims to ensure that the web-based survey application operates correctly and meets user needs. Testing includes various types, such as functional testing, performance testing, security testing, and usability testing. Researchers evaluate the application's performance under different conditions, including high-load scenarios, to ensure that cloud computing integration and the use of Laravel have successfully enhanced the application's scalability and efficiency. Testing results are then used to make improvements and refinements to the system before full deployment of the application.

3.2. System Workflow

The workflow of this application development consists of several stages as shown in Figure 2.

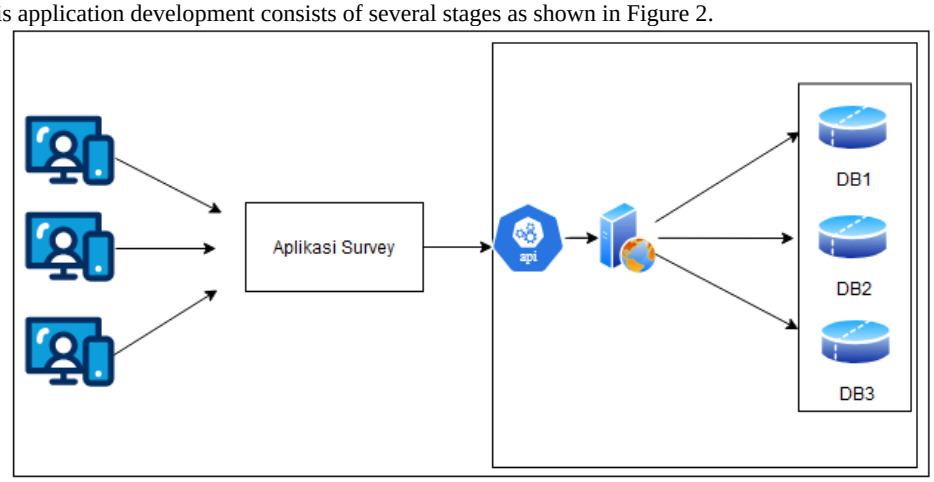

Fig.2: System Workflow

Based on Figure 2, the workflow of this system can be described as follows: The users of this system consist of individuals who access the survey application through various devices such as computers, tablets, and smartphones. Users can be respondents filling out the survey or administrators managing and analyzing survey results. When a user wants to interact with the survey application, they submit a request through the interface provided by the application.
The survey application is the core of the system and is developed using the Laravel framework. Laravel offers various features such as routing, middleware, and Object-Relational Mapping (ORM) that facilitate the development and maintenance of the application. When users submit a request, the survey application receives and processes that request, subsequently sending data to the backend services for further processing. As a PHP framework, Laravel employs the Model-View-Controller (MVC) architecture, which helps separate application logic, user interface, and data, making the application more structured and manageable.
The backend services encompass servers and other components that handle the business logic of the application. In this context, the backend services process survey data received from the survey application, authenticate users, and communicate with the data storage system. The backend server may utilize Laravel APIs to handle various requests from the survey application. Additionally, the backend server manages connections to the database, executes data processing scripts, and ensures that survey data is stored securely and accurately.
To ensure that the survey application can handle high traffic loads and fluctuating user requests, the system employs a load balancer. The load balancer distributes user requests evenly across multiple backend servers, preventing overload on a single server. With the load balancer in place, the performance of the survey application can remain stable and responsive, even when accessed by many users simultaneously.
The data storage system consists of several databases that store all information related to the surveys. These databases receive processed data from the backend services and store it in an organized manner. Each database is designed to support replication and horizontal scaling. The cloud computing technology utilized ensures that the databases are always available and secure, allowing for quick and efficient access to data when needed, whether for result retrieval by administrators or for further analysis. With the integration of cloud technology and the Laravel framework, this survey application can operate optimally and effectively meet user needs.

4. Result and Discussion

The results of the implementation testing of the survey application using the Laravel framework and cloud computing are presented in Table 1.

Table 1: Results of Application Response Time Testing Before Optimization

No	Number of Users	Response Time	Standard Deviation
1	10	1,5	0,2
2	50	1,7	0,3
3	100	2,0	0,4
4	500	3,2	0,6
5	1000	4,5	0,9

From Table 1, it can be observed that the response time of the survey application increases as the number of users grows. When there are only 10 users accessing the application simultaneously, the average response time is 1.5 seconds, with a standard deviation of 0.2 seconds. However, when the number of users rises to 50, the response time increases to 1.7 seconds, accompanied by a standard deviation of 0.3 seconds, indicating an increased load on the system. As the number of users reaches 100, the average response time rises to 2.0 seconds, with a standard deviation of 0.4 seconds. A more significant increase is observed at higher user counts, where with 500 users, the average response time spikes to 3.2 seconds, with a standard deviation of 0.6 seconds. At its peak, when the application is accessed by 1,000 users simultaneously, the response time reaches 4.5 seconds, with a standard deviation of 0.9 seconds. These results demonstrate that the survey application experiences a significant decline in performance as the number of users accessing the application concurrently increases. The increase in response time reflects the additional load placed on the server and supporting infrastructure, which can lead to a suboptimal user experience.

Table 2: Results of Application Response Time Testing After Optimization

No	Number of Users	Response Time	Standard Deviation
1	10	0,8	0,1
2	50	0,9	0,1
3	100	1,0	0,2
4	500	1,5	0,3
5	1000	2,2	0,5

From the table above, it is evident that after optimization, the response time of the survey application has shown a significant performance improvement compared to before optimization. When only 10 users access the application simultaneously, the average response time is 0.8 seconds, with a standard deviation of 0.1 seconds. This indicates that the application can respond to requests quickly and consistently under low load. As the number of users increases to 50, the response time only slightly increases to 0.9 seconds, while the standard deviation remains low at 0.1 seconds. This suggests that the application remains highly responsive and stable despite the increase in user count.

When the user count reaches 100, the average response time increases slightly to 1.0 seconds, with a standard deviation of 0.2 seconds. This relatively small increase in response time indicates that the application can handle increased load effectively without significant performance degradation. At 500 users, the response time rises to 1.5 seconds, with a standard deviation of 0.3 seconds. While there is a more noticeable increase, the application remains responsive and stable, showcasing the system's scalability after optimization.

At its peak, when the application is accessed by 1,000 users simultaneously, the average response time reaches 2.2 seconds, with a standard deviation of 0.5 seconds. Although this response time is higher compared to smaller user counts, the application is still able to deliver relatively fast and stable responses. Furthermore, this increase in response time is still lower than before optimization, indicating that the use of the Laravel framework and cloud computing technology has been effective in enhancing the performance and scalability of the survey application.

5. Conclusion

Based on the research findings and analyses conducted, it can be concluded that the optimization of the web-based survey application using the Laravel framework and cloud computing technology has significantly improved the application's performance. Prior to optimization, the application experienced a considerable increase in response time as the number of users rose, indicating that the initial infrastructure was unable to efficiently handle a large user load. However, following the implementation of Laravel and cloud technology, the application's response time showed a marked decrease, even with an increased user count.

The Laravel framework offers various outstanding features such as caching, efficient routing, and the Eloquent ORM, which optimize database management. The application of these features not only improved response times but also enhanced the overall stability of the application. Testing revealed that the response time for the application with 1,000 users decreased from 4.5 seconds to 2.2 seconds after optimization, with a lower standard deviation as well, indicating improved consistency in application performance.

Cloud computing technology provides the flexibility and scalability necessary to accommodate increased user loads. By utilizing cloud services, the survey application can evenly distribute the load through a load balancer, thereby preventing overload on a single server and ensuring high service availability. The application's reliability also increased, featuring a higher availability rate that guarantees the application can be accessed at any time without significant interruptions.

Overall, the results of this study demonstrate that the integration of the Laravel framework and cloud computing technology is an effective strategy for optimizing the performance of web-based survey applications. These findings are not only relevant to survey applications but can also be applied to the development of other web applications that require high performance and the capacity to handle substantial user loads. Consequently, this research makes a significant contribution to the field of web application development and cloud computing technology, providing practical guidance for developers seeking to enhance their application performance.

References

[1] A. Anton, B. Nuryadi, and H. Herlawati, "Pemanfaatan Teknologi Cloud Computing Untuk Peningkatan Proses Belajar Mengajar," *Jurnal PROSISKO*, vol. 1, 2014, [Online].

[2] S. Pitriyani and R. Firdaus, "Pengembangan Data Base Terdistribusi untuk Aplikasi Cloud Computing," *Innovative: Journal Of Social Science Research*, vol. 4, no. 3, pp. 15905–15917, 2024.

[3] A. Nanda, H. Toha Hidayat, and M. Mahlil, "Implementasi Cloud Computing Untuk Media Pembelajaran Interaktif Bahasa Inggris Berbasis Android," *JAISE : Journal of Artificial Intelligence and Software Engineering*, vol. 3, no. 2, pp. 44–49, 2023, doi: https://dx.doi.org/10.30811/jaise.v3i2.4579.

[4] S. Safriadi and R. Rahmadani, "Analisis Kinerja Load Balancing Round Robin Pada Website Skalabel," *Journal of Information System Management (JOISM)*, vol. 5, no. 2, pp. 227–232, 2024.

[5] M. Kushwaha, B. L. Raina, and S. N. Singh, "Advanced weighted round robin procedure for load balancing in cloud computing environment," in *Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering*, Institute of Electrical and Electronics Engineers Inc., Jan. 2021, pp. 215–219. doi: 10.1109/Confluence51648.2021.9377049.

[6] K. A. Jadhav, M. Moin Mulla, and N. D. G, "An Efficient Load Balancing Mechanism in Software Defined Networks," in *International Conference on Computational Intelligence and Communication Networks* , 2020, pp. 116–122. doi: 10.1109/CICN.2020.23.

[7] J. Wei, X. Chen, J. Wang, X. Hu, and J. Ma, "Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy," *IEEE Trans Dependable Secure Comput*, vol. 19, no. 4, pp. 2318–2332, 2022, doi: 10.1109/TDSC.2021.3055495.

[8] S. M. Al Zikri, "Perancangan Sistem Pengelolaan Data Penerima Dana Zakat, Infaq Dan Sedekah Menggunakan Framework Laravel," *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 2, no. 3, pp. 344–352, 2021.

[9] Z. Li, C. Shang, J. Wu, and Y. Li, "Microservice extraction based on knowledge graph from monolithic applications," *Inf Softw Technol*, vol. 150, p. 106992, Oct. 2022, doi: 10.1016/j.infsof.2022.106992.

[10] S. Ponnusamy and P. Gupta, "Scalable Data Partitioning Techniques for Distributed Data Processing in Cloud Environments: A Review," *IEEE Access*, vol. 12, pp. 26735–26746, 2024, doi: 10.1109/ACCESS.2024.3365810.