Performance Analysis of Distributed Database System in Cloud Computing Environment

1st Nicholas Prasetio
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
nicholas.prasetio@binus.ac.id

2nd Daniel Ferdinand Ardianto
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
daniel.ardianto@binus.ac.id

3rd Brandon Christopher Wijaya
Computer Science Department
Doctor of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
brandon.wijaya@binus.ac.id

4th Maria Susan Anggreainy
BINUS Graduate Program
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11530
marias.susan001@binus.ac.id

5th Afdhal Kurniawan
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
afdhal.kurniawan@binus.ac.id

2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED) | 979-8-3503-7012-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCED60214.2023.10425452

Abstract- **This research discusses the performance evaluation of distributed database systems in a cloud computing environment. Cloud computing environments allow data and applications to be stored and deployed on infrastructure located in different parts of the world. However, the use of distributed database systems in cloud computing environments can cause performance issues, such as complex data access and factors such as network latency, security, and scalability that affect system performance. Therefore, performance evaluation of distributed database systems is necessary to ensure effective data management across the infrastructure. The purpose of this research is to measure and understand the performance of distributed database systems in a cloud computing environment. This is important because proper performance evaluation is needed to ensure distributed database systems can operate effectively and efficiently in such environments. This research will analyze the features of distributed database systems, factors that affect performance, how to measure and compare system performance, and how to improve system performance. Analyzing the performance of distributed database systems in a cloud computing environment can help users choose the most appropriate and efficient cloud computing platform for their business needs and improve operational efficiency.**

Keywords-distributed database system, cloud computing, performance measurement, performance analysis, factors affecting performance.

I. INTRODUCTION

Cloud computing solutions are a result of thegrowing demand for massive data processing and administration. Distributed database systems can be built ona particularly adaptable and scalable foundation provided by cloud computing. Numerous databases that are dispersed geographically and connected by a network make up a distributed database system. Numerous elements, such as load balancing, data replication, and fault tolerance, have an impact on the performance of distributed database systems in cloud computing environments. Therefore, assessing how well-distributed database systems work in cloud computing environments is necessary.

This study aims to examine how well-distributed database systems function in cloud computing environments. The fundamental objective of the study is to examine the efficacy of several distributed database processing methods for processing industrial IoT data utilizing Docker as a service. The study will also go over load-balancing strategies that can be used to improve distributed database system performance in cloud computing environments.

The initial segment of the article [1] will address the challenges associated with utilizing cloud computing platforms for the purpose of managing extensive quantities of data. Subsequently, a comprehensive examination of the benefits and challenges presented by advanced cloud technologies of the next generation, including distributed cloud, fog, and edge computing, will be conducted [2]. The forthcoming publication [3] will present an experimental performance analysis of a scalable distributed Hyperledger Fabric for a large-scale IoT testbed. The study [4] will evaluate the efficacy of distributed database systems utilizing Docker for Industrial Internet of Things (IoT) data. The subsequent segment of the investigation [5] will delve into load-balancing tactics within cloud computing environments.

The investigation will additionally examine plausible hazards, vulnerabilities, and remedial measures associated with cloud computing environments. As partof the project, a demonstration of a sophisticated indoor positioning system for cloud-based environments spanning multiple floors will be presented. The conclusion of thestudy will propose an enhanced quality of service (QoS) cantered methodology for evaluating trust in cloud computing settings, as referenced in citation. The projectproposes a

hybrid database integration technique utilizing SQL for cloud computing environments, as documented in the reference. The study will address the implementationof deep learning and cloud computing techniques for the purpose of database generation, which will be discussed in the penultimate phase.

This research study will meticulously scrutinize the performance of distributed database systems in cloud computing environments. The discussion will encompass the load balancing, data replication, fault tolerance, and security features of distributed database systems. The paper will prove to be a valuable resource for scholars and experts who are interested in enhancing the efficiency of distributeddatabase systems in cloud computing settings.

II. Literature Review

Experience the revolutionary impact of cloud computing on data storage and processing. Enjoy scalable and cost-effective solutions for a wide range of applications. In cloud computing environments, distributed database systems play a crucial role in managing and accessing large volumes of data across multiple nodes. The performance of distributed database systems is of paramount importance to ensure efficient data processing and retrieval. Discover the ultimate analysis of research studies on distributed database systems' performance evaluation in cloud computing environments with our comprehensive literature review.

An automated implementation of hybrid cloud for performance evaluation of distributed databases presents an automated implementation of a hybrid cloud infrastructure to evaluate the performance of distributed databases. Introducing a cutting-edge hybrid cloud model that seamlessly integrates the advantages of public and private clouds for optimal performance. By utilizing automated deployment and orchestration techniques, the model enables efficient performance evaluation of distributed databases. The authors conduct experiments to demonstrate the effectiveness of their approach and highlight its potential for improving the performance of distributed database systems in cloud environments.

Discover the cutting-edge world of Database Development with Deep Learning and Cloud Computing. This innovative study delves into the seamless integration of deep learning techniques and cloud computing to revolutionize database development. Discover how deep learning algorithms can revolutionize the way you handle database operations. Our study showcases the remarkable efficiency and accuracy that can be achieved through this cutting-edge technology. By leveraging cloud computing resources, the proposed approach enables scalable and parallel processing of data. The authors showcase the advantages of their approach through experiments, highlighting the potential for improving the performance of distributed database systems in cloud computing environments.

Performance Evaluation of Distributed Database Strategies Using Docker as a Service for Industrial IoT Data: Application to Industry 4.0 focuses on the performance evaluation of distributed database strategies in the context of Industrial Internet of Things (IIoT) data. Introducing the revolutionary solution for deploying and managing distributed database instances with unparalleled efficiency - Docker as a service. The study recommends leveraging this cutting-edge technology to streamline your database operations. Gain valuable insights into the performance characteristics of distributed database systems through the authors' comprehensive evaluation of various strategies, such as sharding and replication. Discover the significance of carefully choosing strategies to enhance performance in cloud based IoT environments through our latest findings [6].

Discover an innovative integration approach for hybrid databases in cloud computing environments with the help of "An Integration Approach of Hybrid Databases Based on SQL in Cloud Computing Environment." Introducing the revolutionary concept of SQL-based hybrid databases, this study seamlessly blends the unparalleled benefits of both relational and NoSQL databases [7]. Discover the remarkable advantages of the authors' approach, including unparalleled scalability, remarkable flexibility, and exceptional performance. Experience the power of integration! Our experimental evaluations prove that our approach is highly effective in managing large-scale datasets and enhancing the performance of distributed database systems.

Discover the latest findings on SQL and NoSQL database software architectures with Khan et al.'s (2023) comprehensive systematic literature review on performance analysis and assessments. Discover the comprehensive analysis of multiple studies that delve into performance metrics, workload types, benchmarking methodologies, and architectural considerations [8]. Discover the trade-offs between consistency and performance in distributed database systems and learn why benchmarking tools and frameworks are crucial for success. Explore the impact of architectural choices on database performance with insightful findings. Gain a deeper understanding of database performance evaluation in a broader context with the insights provided in this review. While not exclusively focused on distributed databases in a cloud computing environment, these insights can still be applied to inform the evaluation and optimization of distributed database systems in the cloud.

Discover an innovative approach to trust assessment in cloud computing environments with the Enhanced QoS-

Based Model [9]. This cutting-edge model utilizes a quality of service (QoS) framework to provide unparalleled accuracy and reliability. Discover how our study tackles the obstacles of trust assessment in distributed database systems. We consider various QoS factors, including reliability, availability, and performance. Discover a comprehensive framework that incorporates various factors to evaluate the trustworthiness of cloud service providers, as introduced by the authors. With rigorous experimentation, they have confirmed the efficacy of their model in enhancing the efficiency and dependability of cloud-based distributed database systems.

Discover the latest insights on load balancing techniques in cloud computing environments with "Load Balancing Techniques in Cloud Computing Environment: A Review" [10]. Load balancing plays a crucial role in distributing data and workload across multiple nodes to optimize performance and resource utilization. The study examines various load balancing algorithms, such as round-robin, weighted round-robin, and dynamic algorithms. Discover the pros and cons of each technique as the authors delve into their practicality within the realm of cloud-based distributed database systems. The review emphasizes the importance of load balancing for achieving efficient resource utilization and improving the overall performance of distributed database systems.

Detecting impersonation attacks in cloud computing environments using a centric user profiling approach focuses on the detection of impersonation attacks in cloud computing environments [11]. Introducing a cutting-edge approach to user profiling, this study aims to uncover suspicious activities and thwart potential impersonation attacks in distributed database systems. Discover how our profiling model can optimise the security and performance of your distributed databases in the cloud. Our expert authors have analysed user behaviour patterns and access logs to develop a cutting-edge solution that will take your database management to the next level. Experience the power of their approach in reducing the risks of impersonation attacks, as proven by experimental evaluations.

Discover how Althwab's investigative study delves into the performance of distributed relational databases in cloud computing environments [13]. Discover valuable insights into the performance characteristics and challenges associated with distributed relational databases with this groundbreaking study published in 2015. Gain valuable insights into the performance analysis of distributed database systems in today's cutting-edge cloud computing landscape by delving into the findings of this study.

Explore the dynamic world of big data and cloud computing as we delve into the exciting discussions and challenges that lie ahead. Discover the intriguing world of big data and cloud computing integration challenges and discussions with. Discover the fascinating effects of big data on performance in cloud environments, as our study delves into the intricate workings of distributed database systems that handle vast amounts of data. Discover the fascinating insights shared by the authors on the critical components of big data management. From data storage to processing and analytics, they delve into the importance of implementing scalable and efficient distributed database systems to handle the demands of big data workloads. Our solution tackles the obstacles surrounding data security, privacy, and regulatory compliance in cloud-based big data environments.

Discover the groundbreaking findings of "Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed [15]." This study delves into the experimental performance analysis of a scalable distributed Hyperledger Fabric for an IoT testbed. Discover how Hyperledger Fabric, a leading distributed ledger technology, excels in managing vast amounts of IoT data in a cloud-based setting through our comprehensive study. Experience a comprehensive evaluation of performance metrics such as transaction throughput, latency, and resource utilization by the esteemed authors. Discover the remarkable scalability and efficiency of Hyperledger Fabric for distributed database applications in cloud based IoT environments with these groundbreaking findings.

Discover a wealth of insights on the performance analysis of distributed database systems in cloud computing environments through our comprehensive literature review. Our analysis covers a wide range of studies, providing you with a complete understanding of this complex topic. Explore a wide range of topics in the reviewed research works, including hybrid cloud implementation, deep learning integration, performance evaluation strategies, security threats, trust assessment models, load balancing techniques, big data challenges, and detection of impersonation attacks. Unlock valuable insights into the performance factors, challenges, and opportunities of distributed database systems in the cloud computing context with these studies. Gain a deeper comprehension of distributed database systems in cloud environments with the valuable insights provided by these studies. Discover optimal performance characteristics and optimization approaches that can enhance your system's efficiency.

III. Material and Method

This research analyses the performance of a distributed database in a cloud computing environment based on the duration, which means the amount of time it takes for a request to complete is calculated as the sum of processing time and communication costs. This is an indicator of whether the experiment is problematic and, as a general rule,

shorter periods are more optimal. However, longer durations indicate a result that is not really optimal. Please note that the level of complexity of a query and much more affects the duration for executing the query

To find out which one is superior between cloud and local based on the duration of executing a query, an experiment was conducted by using simple databases that we created.

This test is carried out using AWS cloud computing (remoted location at N.Virginia), as well as a local computer, both of which will use MySql in the same application, namely XAMPP to execute the same query, where later the time to execute the query will be used as a benchmark in this study.

In the experiment, 5 SQL queries were executed. three queries retrieve data, one query deletes data, and one query updates. Where all three queries implement an inner join. The complexity of the query is related to the number of join conditions and use clauses, such as sorting or aggregation of data, in the query.

The details of the experiment are as follows:

EXPERIMENT1 (can be seen in Fig 1 and Fig 2) Implements an inner join with two tables and using where condition; transaction_header and customers, two column from transaction_header that is transaction_id and transaction_date, and three column from customers that is customer_name, customer_gender, and customer_address.

Query: select th.transaction_id,c.customer_name,c.customer_gender,c.customer_address,th.transaction_date from transaction_header th join customers c on th.customer_id = c.customer_id where year(th.transaction_date) >= 2001;

EXPERIMENT2 (can be seen in Fig 3 and Fig 4) Implements an inner join with four tables; transaction_header, customers, transaction_detail, products, two column from transaction_header that is transaction_id, and transaction_date, one column from customers that is customers_name, and two column from products that is product_name and product_price times transaction_detail.quantity named as Total Price, and Order by transaction_id ASC.

Query: select th.transaction_id, DATE_FORMAT(th.transaction_date ,'%M, %d %Y') `Date of Transaction` , c.customer_name , p.product_name , p.product_price*td.quantity `Total Price` from transaction_header th join customers c on th.customer_id = c.customer_id join transaction_detail td on th.transaction_id = td.transaction_id join products p on td.product_id = p.product_id order by th.transaction_id

EXPERIMENT3 (can be seen in Fig 5 and Fig 6) Implements an UNION with two tables and using CONCAT;UNION data in from table employee and customer, four coulumn from employee that is employee_id,employee_name,employee_gender,employee_dob and for column from customer that is customer_id,customer_name,customer_gender,customer_dob.

Query: SELECT employee_id,employee_name,employee_gender,employee_dob from employee UNION SELECT customer_id,customer_name,customer_gender,customer_dob from customers

EXPERIMENT4 (can be seen in Fig 7 and Fig 8) Implements delete data with where condition;Delete customers table data with a AND condition.

Query: DELETE from customers WHERE year(customer_dob) < 2000 AND customer_gender = 'Female'

EXPERIMENT5 (can be seen in Fig 9 and Fig 10) Implements Create view with a aggregate function;Create TopSellerProduct view that have product_id, product_name, product_sold.

Query: create view TopSellerProduct AS select p.product_id ,p.product_name ,p.product_price ,COUNT(td.product_id) as product_sold from transaction_detail td join products p on td.product_id = p.product_id GROUP by td.product_id having COUNT(td.product_id) >= 1

IV. RESULTS

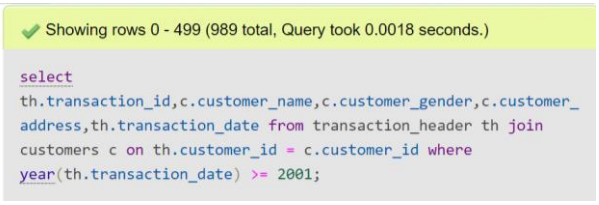

Fig 1. Experiment 1 on Desktop

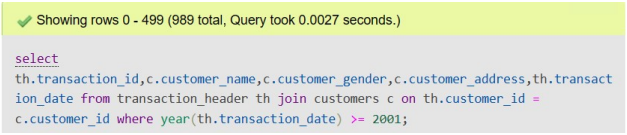

Fig 2. Experiment 1 on AWS

Experiment 1 yields a query that executes an inner join operation between the "transaction_header" and "customers" tables. The query retrieves the transaction ID and date from the transaction_header table, along with the customer's name, gender, and address from the customer's table. The query incorporates a criterion aimed at restricting transactions that possess a transaction date falling on or after the year 2001. It was executed for 0.0018

seconds on desktop and 0.0027 seconds on AWS to show the result.

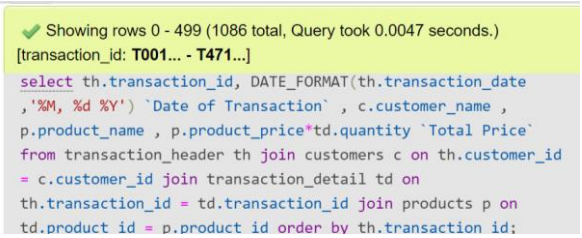

Fig 3. Experiment 2 on Desktop

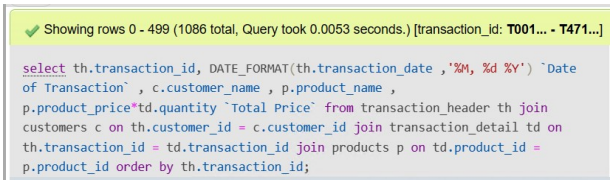

Fig 4. Experiment 2 on AWS

The outcome of the conducted experiment is a query that executes an inner join operation on four distinct tables, namely "transaction_header," "customers," "transaction_detail," and "products." The query retrieves the transaction ID, transaction date, customer name, product name, and total price, which is calculated by multiplying the product price by quantity, from their corresponding tables. The outcome is arranged in ascending order based on the transaction ID. It was executed for 0.0047 seconds on desktop and 0.0053 seconds on AWS to show the result.

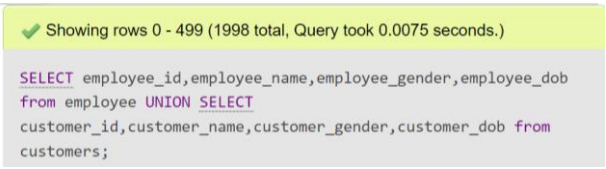

Fig 5. Experiment 3 on Desktop

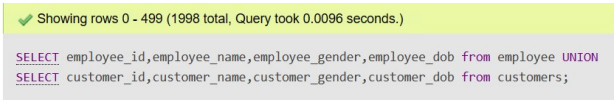

Fig 6. Experiment 3 on AWS

The experimental outcome entails a query that employs the UNION operator to merge information from the "employee" and "customers" tables. The query retrieves employee data, including employee ID, name, gender, and date of birth, from the employee table, and customer data, including customer ID, name, gender, and date of birth, from the customers table. The outcome is a consolidated dataset derived from the amalgamation of the two tables. It was executed for 0.0075 seconds on desktop and 0.0096 seconds on AWS to show the result.

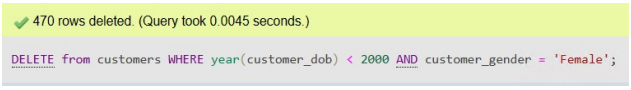

Fig 7. Experiment 4 on Desktop

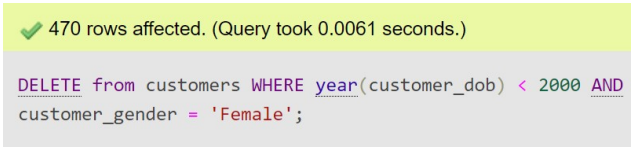

Fig 8. Experiment 4 on AWS

The outcome of the experiment involves the execution of a DELETE query, which effectively eliminates data from the "customers" table. The query incorporates a criterion that eradicates entries in which the birthdate of the customer predates the year 2000 and the gender is female. The aforementioned query proficiently eliminates customer data that corresponds to the designated criteria. It was executed for 0.0061 seconds on desktop and 0.0045 seconds on AWS to show the result.

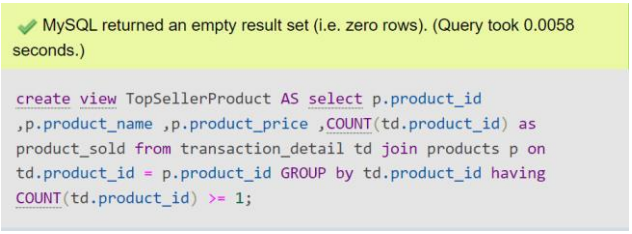

Fig 9. Experiment 5 on Desktop

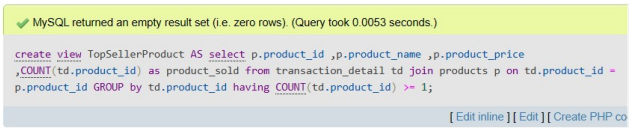

Fig 10. Experiment 5 on AWS

The outcome of the conducted experiment is the generation of a query that produces a view named "TopSellerProduct." The view incorporates data sourced from the "products" table, encompassing the product's unique identifier, name, price, and the tally of its sales frequency. The perspective is established through the execution of an inner join operation between the "transaction_detail" and "products" tables, followed by the grouping of outcomes based on product ID, and the application of a filter to retain only those products that have been sold at least once. It was executed for 0.0058 seconds on desktop and 0.0053 seconds on AWS to show the result.

Table 1. Results

Number of Experiment	Desktop	AWS
1	**0.0018 s**	0.0027 s
2	**0.0047 s**	0.0053 s
3	**0.0075 s**	0.0096 s
4	0.0061 s	**0.0045 s**
5	0.0058 s	**0.0053 s**

IV. Conclusion

This manuscript discusses distributed database systems in cloud computing, the experiments were conducted using AWS cloud computing with a remote location in North Virginia (N. Virginia), which could have implications for network latency and communication costs. Conducted a comparative analysis of the performance of a distributed database in a cloud computing environment (AWS) and a local environment using XAMPP with MySQL. The primary metric for evaluation was the duration it took for SQL queries to complete.

Based on the experiment result, it can be concluded that the execution times for the given experiments were generally faster on the desktop compared to AWS (Amazon Web Services). This shows that for these specific experiments, the desktop environment outperformed AWS in terms of processing speed. However, it's important to note that the actual performance may vary depending on the specific tasks and configurations involved in the experiments.

The choice between cloud and local hosting may depend on factors like query complexity, network latency, and the specific requirements of the application. Further analysis and consideration of these factors would be necessary to make a definitive determination regarding the superiority of one environment over the other for the given use case.

References

[1] Mansouri, Y., Prokhorenko, V., & Babar, M. A. (2020). An automated implementation of hybrid cloud for performance evaluation of distributed databases. Journal of Network and Computer Applications, 167, 102740. https://doi.org/10.1016/j.jnca.2020.102740.

[2] Liu Rukun, T. Wang, Y. Yang, and B. Yu, "Database Development Based on Deep Learning and Cloud Computing," *Mobile Information Systems*, vol. 2022, pp. 1–10, Apr. 2022, doi: https://doi.org/10.1155/2022/6208678.

[3] T. Gkamas, V. Karaiskos, and S. Kontogiannis, "Performance Evaluation of Distributed Database Strategies Using Docker as a Service for Industrial IoT Data: Application to Industry 4.0," *Information (Switzerland)*, vol. 13, no. 4, Apr. 2022, doi: 10.3390/info13040190.

[4] C. Li and J. Gu, "An integration approach of hybrid databases based on SQL in cloud computing environment," *Software - Practice and Experience*, vol. 49, no. 3, pp. 401–422, Mar. 2019, doi: https://doi.org/10.1002/spe.2666.

[5] Maniah, Edi Abdurachman, Ford Lumban Gaol, and Benfano Soewito, "Survey on Threats and Risks in the Cloud Computing Environment," *Procedia Computer Science*, vol. 161, pp. 1325–1332, Jan. 2019, doi: https://doi.org/10.1016/j.procs.2019.11.248.

[6] H. A. Hassan, A. I. El-Desouky, A. Ibrahim, E.-S. M. El-kenawy, and Reham Arnous, "Enhanced QoS-Based Model for Trust Assessment in Cloud Computing Environment," *IEEE Access*, vol.8, pp. 43752–43763, Mar. 2020, doi: https://doi.org/10.1109/access.2020.2978452.

[7] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7. King Saud bin Abdulaziz University, pp. 3910–3933, Jul.01, 2022. doi: 10.1016/j.jksuci.2021.02.007.

[8] W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo, "SQL and NoSQL Database Software Architecture Performance Analysis and Assessments—A Systematic Literature Review," vol. 7, no. 2, pp. 97–97, May 2023, doi: https://doi.org/10.3390/bdcc7020097.

[9] Atieh, A. T. (2021). The Next Generation Cloud technologies: A Review On Distributed Cloud, Fog And Edge Computing and Their Opportunities and Challenges. ResearchBerg Review of Science and Technology, 1(1), 1–15. https://researchberg.com/index.php/rrst/article/view/18.

[10] Dong, Z., Zheng, E., Choon, Y., & Zomaya, A. Y. (2019). DAGBENCH: A Performance Evaluation Framework for DAG Distributed Ledgers. 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). https://doi.org/10.1109/cloud.2019.00053.

[11] Kholidy, H. A. (2021). Detecting impersonation attacks in cloud computing environments using a centric user profiling approach. Future Generation Computer Systems, 117, 299–320. https://doi.org/10.1016/j.future.2020.12.009.

[12] S.-H. Kim and T. Kim, "Local Scheduling in KubeEdge-Based Edge Computing Environment," vol. 23, no. 3, pp. 1522–1522, Jan. 2023, doi: https://doi.org/10.3390/s23031522.

[13] A. Althwab, "Distributed relational database performance in Cloud Computing: an investigative study," *Aut.ac.nz*, 2015. https://openrepository.aut.ac.nz/items/844643ae-7206-4f74-b825-596d0a84e976 (accessed May 03, 2023).

[14] Sandhu, A. K. (2022). Big data with cloud computing: Discussions and challenges. Big Data Mining and Analytics, 5(1), 32–40. https://doi.org/10.26599/bdma.2021.9020016.

[15] H. H. Pajooh, M. A. Rashid, F. Alam, and S. Demidenko, "Experimental Performance Analysis of a Scalable Distributed Hyperledger Fabric for a Large-Scale IoT Testbed," *Sensors*, vol. 22, no. 13, Jul. 2022, doi: 10.3390/s22134868.