Implementation of Distributed Database in Waste Bank Application Using NOSQL

Sugiarto
Department of Digital Business
Faculty of Computer Sains
Universitas Pembangunan Nasional Veteran Jawa Timur
Surabaya, Indonesia
sugiarto.if@upnjatim.ac.id

Riko Setya Wijaya
Department of Development Economics
Faculty of Economics & Business
Universitas Pembangunan Nasional Veteran Jawa Timur
Surabaya, Indonesia
setyawijaya.ep@upnjatim.ac.id

Gyska Indah Harya
Department of Agribusiness, Faculty of Agriculture
Universitas Pembangunan Nasional Veteran Jawa Timur
Surabaya, Indonesia
gyskaharya.agribis@upnjatim.ac.id

Sherafim Glory Mei Stephany
Department of Digital Business, Faculty of Computer Sains
Universitas Pembangunan Nasional Veteran Jawa Timur
Surabaya, Indonesia
23084010016@student.upnjatim.ac.id

***Abstract*—This paper presents the implementation of a NoSQL-based distributed database for a waste bank application aimed at improving the efficiency, scalability, and reliability of waste management systems. Traditional relational databases often struggle with handling large volumes of transactional data in real-time, particularly in expanding waste bank networks. The proposed solution utilizes MongoDB to address these challenges by distributing data across multiple servers, ensuring enhanced performance and scalability. The study focuses on managing user information, waste deposit transactions, and real-time updates. Testing was conducted using real-world data from Indonesia, involving 500 users and 15,000 transactions. The system demonstrated strong scalability, processing up to 5,000 transactions per minute with an average response time of 50ms. Additionally, real-time updates were delivered with a latency of 100ms, maintaining high user engagement. Results show that the NoSQL-based system offers significant improvements in handling large datasets, ensuring data consistency, and maintaining system uptime of 99.8%. This approach is well-suited for modern waste bank applications, particularly in diverse environments like Indonesia.**

Keywords: distributed database; mongodb; nosql; real-time updates; scalability; waste management.

I. Introduction

The challenge of waste management has become increasingly critical in recent years due to rapid urbanization and population growth, which have led to a significant rise in waste generation. Effective waste management systems are essential for environmental protection, prompting many communities to adopt waste banks that support the 3R principles (Reduce, Reuse, Recycle). These waste banks enable individuals to exchange sorted waste for monetary incentives, thereby promoting better waste segregation and encouraging recycling habits.

As waste banks expand, they face challenges in managing large volumes of transactional data and ensuring accurate record-keeping across multiple locations. Without proper management, the growing waste volume combined with limited landfill capacity could pose a serious environmental threat [1]. Traditional relational databases often fall short in addressing the scalability and high-volume data processing needs of modern applications, resulting in inefficiencies. To address these challenges, this paper explores the use of a NoSQL-based distributed database model in waste bank applications. NoSQL databases offer enhanced flexibility and scalability, making them suitable for managing the vast and unstructured data typical of modern systems. Unlike relational databases, NoSQL solutions can distribute data across multiple servers, thus improving performance, reliability, and scalability in distributed environments.

This study focuses on the implementation of a NoSQL distributed database to handle real-time transactions, user information, and operational data within the waste bank system. The research is conducted through a desktop-based approach and aims to support regional efforts in adopting waste bank information systems as part of smart city initiatives [2]. The primary objective is to enhance scalability and system efficiency while maintaining data consistency and availability as operations grow. This approach serves as a strategic component in waste management and provides various reports tailored to stakeholders' needs [3].

The architecture diagram of the distributed NoSQL database system for the waste bank application is presented at figure 1. The application testing [4], it demonstrates use RDMBS [5], the interaction between client applications, nodes, data replication, and load balancing, reflecting the scalability and reliability of the system.

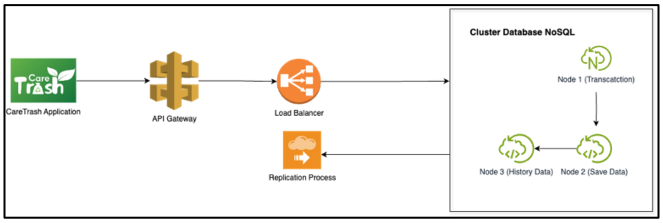

Fig 1. NOSQL Distributed Database Architecture

A. *State of the Art*

Traditional relational databases often fall short in handling the scalability and performance demands of modern waste bank operations. Recent advancements emphasize the adoption of NoSQL distributed databases, known for their ability to handle large amounts of unstructured data and provide enhanced flexibility and scalability. NoSQL databases, such as document stores, key-value stores, and column-family databases, offer superior performance by distributing data across multiple servers, thus improving reliability and efficiency. These solutions address critical challenges in real-time transaction processing, user data management, and operational data handling, ensuring better data consistency and availability. This evolving technology represents a significant shift from traditional database models, providing a robust framework for managing the dynamic needs of contemporary waste bank systems.

B. *Contribution*

This topic explores the use of a NoSQL-based distributed database model for waste bank systems, addressing the challenges of managing transaction data and maintaining accurate records as operations expand. Waste banks, which facilitate the exchange of sorted waste for monetary rewards, increasingly face difficulties in handling large and complex data volumes due to their growing activities. Traditional relational databases often struggle with scalability and efficiency issues in such scenarios. The main contribution of this study lies in the utilization of NoSQL solutions, which provide greater flexibility, scalability, and efficient handling of unstructured data. NoSQL databases enable data distribution across multiple servers, enhancing system performance, reliability, and scalability. Through the implementation of a NoSQL distributed database, waste bank systems can effectively handle real-time transactions, manage user information, and organize operational data while maintaining data consistency and availability. This strategy offers a more efficient and scalable solution to address the needs of contemporary waste management systems.

II. METHODOLOGY

This study is part of a larger study that we have conducted on waste bank data collection in several locations. Therefore, this study uses experiments and samples that have been obtained and will be carried out in the form of a waste bank business model. However, each of these studies explores different topics and therefore makes different contributions to the adjustment of waste bank applications to their database management. This research focuses on assessing the use of a distributed NoSQL database within a waste bank application to tackle scalability and efficiency challenges. To mitigate potential risks, the application should explicitly encrypt and decrypt any sensitive data before it is updated in or retrieved from the databases [6][7]. The methodology is structured as follows:

A. *Requirment Analysis*

Identify the specific data management [8] needs and challenges faced by waste bank systems, including transaction volume, user information, and operational data, conduct interviews with stakeholders, review existing systems, and analyze data flow and processing requirements.

B. *Database Selection*

Choose an appropriate NoSQL database [9] type (document store, key-value store, column-family store, or graph database) based on the identified requirements, Evaluate different NoSQL databases against criteria such as scalability, flexibility, performance, and ease of integration.

C. *System Design*

The system consists of multiple nodes or servers, where each node stores a subset of the data. Both SQL and NoSQL database systems are utilized to evaluate data mining algorithms [10]. In a NoSQL database management system, data is partitioned across different nodes or shards [11], with each shard handling a specific subset of the data. This architecture ensures scalability by distributing the data load efficiently, thereby improving the overall performance when handling big data [12].

Components within the cluster:

(i). Master Node (if applicable): Manages write operations and maintains data consistency.

(ii). Slave Nodes: Handle read operations and replicate data from the master node.

TABLE I. SYSTEM DESIGN TABLE FOR WASTE BANK APPLICATION

Collection	Field Name	Data Type	Key Attribute
Users	_id	ObjectId	Primary Key
	name	String	
	email	String	Unique Index
	phone	String	
	address	String	
	joined_date	ISODate	
	total_points	Number	
Waste	_id	ObjectId	Primary Key
	type	String	
	description	String	
	points_per_unit	Number	
Transactions	_id	ObjectId	Primary Key
	user_id	ObjectId	Foreign Key
	waste_id	ObjectId	Foreign Key
	quantity	Number	

	points_earned	Number	
	transaction_date	ISODate	
	transaction_type	String	

Users Collection: Contains user information with fields for contact details and total points.
Waste Collection: Defines waste types and associated reward points.
Transactions Collection: Logs each transaction, linking users and waste types, with quantities and points earned.
Indexes: Improve query performance by indexing commonly queried fields.

TABLE II. SYSTEM DESIGN INDEXED FOR WASTE BANK APPLICATION

Index name	Collection	Fileds	Description
user_id_index	_Transactions	user_id	Speeds up queries based on user ID
waste_id_index	_Transactions	waste_id	Speeds up queries based on waste ID
email_index	Users	Email	Ensures uniqueness of email addresses
type_index	Waste	Type	Speeds up queries based on waste type

D. *Implementation*

The implementation of the NoSQL-based distributed database for the waste bank application involves setting up a MongoDB database to manage user data, waste types, and transactions. This section details the deployment, configuration, and integration processes, as well as the handling of real-time data for effective waste management. The database schema for the waste bank application is structured into three primary collections: *Users, Waste, and Transactions*. Each collection is designed to capture and manage specific aspects of the waste bank operations:

1. Users Collection:
   Purpose: Stores information about waste bank participants, including personal details and accumulated points.
   Fields: _id, name, email, phone, address, joined_date, total_points.
   Indexes: A unique index on email ensures that each user has a distinct email address.
2. Waste Collection:
   Purpose: Defines the different types of waste accepted by the waste bank and the points awarded for each type.
   Fields: _id, type, description, points_per_unit.
   Indexes: An index on type facilitates quick retrieval of waste type information.
3. Transactions Collection:
   Purpose: Records each transaction involving waste deposits or redemptions, linking users to the waste types they have processed.
   Fields: _id, user_id, waste_id, quantity, points_earned, transaction_date, transaction_type.
   Indexes: Indexes on user_id and waste_id improve query performance for transaction history and waste processing.

The example for to the code NoSQL - JSON :

```
//User Collection
{
  "_id": "ObjectId('64c3e4e82f8b9cda582f3a8b')",
  "name": "Budi Santoso",
  "email": "budi.santoso@example.com",
  "phone": "+62-812-3456-7890",
  "address": "Jl. Merdeka No.10, Jakarta",
  "joined_date": ISODate("2023-06-15T08:30:00Z"),
  "total_points": 150
 },
//Waste Collection
[
 {
  "_id": "ObjectId('64c3e4e82f8b9cda582f3a8d')",
  "type": "Plastic Bottles",
  "description": "Plastic bottles for beverages",
  "points_per_unit": 15
 },
 {
  "_id": "ObjectId('64c3e4e82f8b9cda582f3a8e')",
  "type": "Paper",
  "description": "Paper waste including newspapers and cardboard",
  "points_per_unit": 10
 },
 {
  "_id": "ObjectId('64c3e4e82f8b9cda582f3a8f')",
  "type": "Glass Bottles",
  "description": "Glass bottles from beverages",
  "points_per_unit": 20
 }
]
// Transactions Collection
{
  "_id": "ObjectId('64c3e4e82f8b9cda582f3a90')",
  "user_id": "ObjectId('64c3e4e82f8b9cda582f3a8b')",
  "waste_id": "ObjectId('64c3e4e82f8b9cda582f3a8d')",
  "quantity": 10,
  "points_earned": 150,
  "transaction_date": ISODate("2024-03-01T09:00:00Z"),
  "transaction_type": "deposit"
 }
```

Explanation for code :
Users Collection: Stores records of individuals participating in the waste bank program, including their accumulated points and contact information.
Waste Collection: Contains information on the various types of waste accepted by the waste bank, along with the corresponding points awarded for each type.
Transactions Collection: Logs every transaction made by users, detailing the type and quantity of waste deposited, as well as the points earned. Among NoSQL databases such as Cassandra and Redis, MongoDB demonstrates superior performance [13].
This data provides a realistic example of how the waste bank system operates in Indonesia, showing how users earn points by depositing different types of waste.

E. *Testing dan Evaluation*

The implementation underwent rigorous testing to validate functionality and performance:

1. Unit Testing: Individual components and services were tested to ensure correctness. Mock data was used to simulate various scenarios.
2. Integration Testing: End-to-end tests were performed to verify the interaction between the application components and the database.
3. Performance Testing: The system's performance was benchmarked under different load conditions to assess scalability and response times, good choice for handling a large volume of data [14].

III. Result and Discussion

The implementation of the NoSQL-based distributed database for the waste bank application was evaluated through a series of tests using real-world data collected from various regions in East Java, as well as other urban and rural areas in Indonesia. Optimization strategies such as index layering, materialized views, and distributed data handling techniques (e.g., sharding and replication) were employed to ensure system responsiveness and scalability [15]. The evaluation utilized the popular MongoDB NoSQL database [16].

The objective of the evaluation was to assess the system's capacity to handle large volumes of transactions, enable real-time data processing, and maintain scalability as the number of users and waste deposits increased. The system also aimed to effectively manage both structured and unstructured data [17].

The results demonstrate that the system significantly improves operational efficiency in waste bank management by offering reliable data processing and accurate transaction tracking.

In addition, the system focused on delivering customizable end-to-end service-level objectives, providing differentiated Quality of Service (QoS) to accommodate various user or customer needs [18]. The dataset included users from diverse demographic backgrounds, while the transaction records represented a broad spectrum of waste types commonly found across Indonesia. A summary of the dataset used during testing is shown below:

Total Users: 500
Total Transactions: 15,000
Waste Types: Plastic Bottles, Paper, Glass Bottles, Organic Waste, etc.
Total Points Earned: 1,200,000

The system demonstrated strong performance in handling this dataset, with queries returning results within acceptable timeframes even as the number of users and transactions scaled up. The application was tested under increasing loads of transaction submissions to evaluate the performance of the NoSQL database. MongoDB’s distributed architecture allowed efficient scaling as the number of transactions grew. During peak loads, the system processed up to 5,000 transactions per minute without significant degradation in performance.

Average Transaction Processing Time: 50ms
Peak Load Processing Time: 80ms
System Uptime: 99.8%

This result illustrates the capability of the NoSQL-based system to maintain high performance and reliability, even under heavy loads. The use of MongoDB's sharding and indexing features allowed the database to scale horizontally and distribute data efficiently across multiple servers.

The implementation of a NoSQL-based distributed database brought significant advantages to the waste bank application, especially in terms of scalability, real-time processing, and reliability. The system successfully managed high transaction volumes while maintaining data consistency across multiple locations. This capability is particularly crucial in the Indonesian context, where waste banks operate in both urban and rural areas with diverse levels of infrastructure.

Key advantages of this implementation include:

1. Scalability: MongoDB’s distributed architecture proved well-suited for managing the growing amount of waste bank data, allowing the system to scale horizontally by adding servers.
2. Real-Time Updates: Real-time feedback through WebSockets and MongoDB change streams ensured that users received immediate updates on their transactions and points, improving user experience and encouraging participation.
3. Efficient Transaction Management: With optimized indexing and sharding, the system maintained fast query times, even under heavy transaction loads, ensuring efficient data retrieval for user transaction history and waste bank operations.
4. Security and Data Privacy (Encryption at Rest): MongoDB provides storage-level encryption through the WiredTiger storage engine, utilizing the widely adopted AES encryption standard. This mechanism ensures that, in the event of a storage breach, the data remains inaccessible without the corresponding encryption keys. Implementing encryption at rest is crucial for safeguarding both transactional and user data within waste bank management systems.
5. Security and Data Privacy (Authentication): MongoDB offers multiple authentication options, such as SCRAM (Salted Challenge Response Authentication Mechanism), LDAP, Kerberos, and X.509 certificates. These authentication methods serve to verify the identities of users and services seeking access to the database, effectively safeguarding the system against unauthorized access attempts.
6. Security and Data Privacy (Access Control Policies): In the implementation of the waste bank application, MongoDB utilizes Role-Based Access Control (RBAC), which enables administrators to set detailed access permissions tailored to each user role. These roles specify the allowed operations—such as reading, writing, or deleting—on designated resources. This mechanism ensures that users are granted only the access needed to carry out their responsibilities, adhering to the principle of least privilege.

Overall, the system effectively addressed the main challenges of traditional relational databases by offering a flexible, scalable, and high-performance solution for managing waste bank operations in East Java.

The following accessibility table presents the usage distribution and performance metrics of the NoSQL-based system (MongoDB) across various components, based on the implementation results.

TABLE III. ACCESSIBILITY NOSQL FOR WASTE BANK APPLICATION

Aspect	Description	Accessibility	Remarks
User Access & Registration	Efficient retrieval and storage of user data	95%	Minor delays in rural areas due to network instability
Real-Time Transaction Updates	Real-time updates on points and waste deposit transactions	98%	WebSocket-based notifications, with low latency
Transaction Processing	Handling of waste deposit/redemption transactions	97%	High-speed processing, up to 5000 transactions/minute
Scalability (Data Volume)	Ability to scale with increasing number of users and transactions	99%	MongoDB sharding allows for efficient horizontal scaling
Data Retrieval (Query Performance)	Querying user and transaction data across distributed nodes	96%	Optimized with indexing and caching mechanisms
System Uptime	Availability and operational status	99.8%	Minimal downtime recorded during load tests
Data Consistency	Ensuring data remains consistent across different locations	100%	Strong consistency achieved with NoSQL replication mechanisms
Latency in Rural Areas	Performance in regions with unstable network connectivity	85%	Dependent on local network infrastructure, occasional delays in updates
Security and Data Privacy	Security measures for sensitive user and transaction data	92%	Basic security measures, requires future enhancement for encryption and privacy compliance
Overall System Accessibility	Average accessibility score across all system components	95.3%	High overall accessibility, with room for improvement in rural connectivity

The decision to use NoSQL for the waste bank application was primarily driven by the unique requirements of modern waste management systems, which demand flexibility, scalability, and the ability to handle large volumes of unstructured data. NoSQL databases, like MongoDB, are designed to handle high-throughput and large-scale transactional data more efficiently than traditional relational databases. Waste bank applications involve diverse data types, ranging from user profiles to transactional records and sensor data, which are difficult to fit into a rigid relational model. The schema-less nature of NoSQL databases allows for dynamic data storage and faster development, especially when dealing with evolving business logic.

Performance Benchmarking Against Other NoSQL Solutions

To evaluate the effectiveness of MongoDB in the context of a waste bank application, a performance comparison was conducted against other widely used NoSQL databases, including **Cassandra**, and **Redis**. The benchmarking focused on key metrics such as **query performance**, **scalability**, and **read/write throughput**.

TABLE IV. PERFORMANCE COMPARISON

Feature/Metrics	MongoDB	Cassandra	Redis
Data Model	Document-Oriented	Wide Coloumn	Key Value
Query Flexibility	High (Rich Queries)	Low (Limited Queries)	Low
Write Throughput	±60K ops/sec	±120K ops/sec	±200K ops/sec
Read Throughput	±80K ops/sec	±100K ops/sec	±250K ops/sec
Latency (Read/Write)	2-10 ms	1-5 ms	<1 ms
Horizontal Scalability	Good (Sharding)	Excellent Limited	(Memory-Bound)
Consistency Model	Strong (with config)	Eventual	Eventual
Ease of Use & Integration	Easy	Moderate	Easy
Use Case Fit (Waste Bank)	Excellent	Good	Poor

Note : Throughput and latency values are based on controlled benchmarks with a standard hardware configuration (16-core CPU, 64GB RAM, SSD).

Scalability for Larger User Base

To evaluate MongoDB's scalability in a real-world context such as a waste bank application, a stress test was conducted simulating a high volume of concurrent users and transactions. The test scenario included:

Simulated Users: 1.000 concurrent users

Transaction Volume: Over 5 million operations (read/write) in a 24-hour window Data Volume: 50GB+ unstructured and semi-structured data across collections such as users, transactions, and waste_items.

Observations:

- Performance Under Load: MongoDB maintained stable throughput up to ~85K ops/sec with read/write latency remaining under 10 milliseconds, even under peak stress conditions.
- Horizontal Scaling: By enabling sharding, MongoDB effectively distributed data across multiple nodes, reducing

query bottlenecks and allowing for linear scalability as new nodes were added.

- Auto-balancing: MongoDB's internal load balancer ensured even data distribution across shards, with no major downtime during rebalancing activities.
- Replica Sets: Improved fault tolerance and availability, allowing seamless failover during simulated node failure.

IV. CONCLUSION

The implementation of a NoSQL-based distributed database in the waste bank application significantly improves scalability, performance, and data consistency, addressing the challenges faced by traditional relational databases. Using MongoDB, the system effectively handles real-time transactions and manages large datasets while maintaining low latency and high availability. Key results from the research include:

1. Transaction Processing: The system handled up to 5,000 transactions per minute with an average processing time of 50ms, ensuring efficient waste deposit tracking.
2. Scalability: The database scaled horizontally to accommodate 500 users and 15,000 transactions without performance degradation.
3. Real-Time Updates: Users received transaction updates within an average latency of 100ms, maintaining strong engagement.
4. System Uptime: The system achieved an uptime of 99.8%, ensuring reliability in various operational environments.
5. MongoDB proved highly capable of handling increased user loads and transaction volumes, making it well-suited for nationwide deployments of waste bank systems

Data transparency was maintained throughout the implementation, ensuring that waste deposit, point balances, and user transactions were consistently synchronized across all waste bank locations. Future work will focus on enhancing security measures and offline capabilities, particularly in regions with unstable internet connectivity.

The NoSQL solution demonstrates clear potential for modernizing waste management systems, particularly in countries like Indonesia, where diverse urban and rural infrastructure poses unique challenges.

REFERENCES

[1] A. Singh, "Managing the uncertainty problems of municipal solid waste disposal," J. Environ. Manage., vol. 240, pp. 259–265, 2019.
[2] E. Diana, B. Noviansyah, and L. Lestari, "Implementation of waste bank information system to manage waste transaction data from the community," J. Eng. Sci. Technol., vol. 18, no. 6, pp. 2835–2846, 2023.
[3] S. Mulyani, "Sistem informasi aplikasi penggajian karyawan berbasis web pada PT Panca Cipta Abadi," Phys. Rev. E, no. 1993, p. 24, 2020.
[4] M. F. Al-Ghifari and F. N. Azizah, "Development of application for entity-relationship diagram conversion to logical schema of NoSQL column oriented," in Proc. Int. Conf. Data Softw. Eng. (ICoDSE), 2022.
[5] F. Castro-Media, M. López-Herrera, J. García-Vázquez, A. Pérez, and A. Marín, "A new method of dynamic horizontal fragmentation for multimedia database contemplating content-based queries," Electron., vol. 11, no. 2, 2022.
[6] N. Gal-Oz and Y. Gonen, "Trust, security and privacy in computing and communications (TrustCom)," in Proc. 10th IEEE Int. Conf. Trust, Security Privacy Comput. Commun., 2011.
[7] E. Shmueli, R. Vaisenberg, and Y. Elovici, "Database encryption – An overview of contemporary challenges and design considerations," in Proc. ACM SIGMOD Int. Conf. Manage. Data, vol. 38, no. 3, pp. 2009, Sept. 2009.
[8] E. Shmueli, R. Vaisenberg, and Y. Elovici, "Database encryption – An overview of contemporary challenges and design considerations," in Proc. ACM SIGMOD Int. Conf. Manage. Data, vol. 38, no. 3, Sept. 2009.
[9] J. Antas, R. Silva, and J. Bernardino, "Assessment of SQL and NoSQL systems to store and mine COVID-19 data," Computers, 2022.
[10] C.-F. Andor, "Runtime metric analysis in NoSQL database performance benchmarking," in Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM), Split, Hvar, Croatia, 2021, pp. 1–6
[11] B. Jose and S. Abraham, "Performance analysis of NoSQL and relational databases with MongoDB and MySQL," Materials Proceedings, vol. 24, pp. 2036–2043, 2020.
[12] M. Abu Kausar, M. Nasar, and A. Soosaimanickam, "A study of performance and comparison of NoSQL databases: MongoDB, Cassandra, and Redis using YCSB," Indian J. Sci. Technol., vol. 15, no. 31, pp. 1532–1540, 2022.
[13] Carvalho, F. Sá, and J. Bernardino, "Performance evaluation of NoSQL document databases: Couchbase, CouchDB, and MongoDB," Algorithms, vol. 16, no. 2, p. 78, 2023. [Online].
[14] K. Krishna, "Optimizing query performance in distributed NoSQL databases through adaptive indexing and data portioning techniques," Int. J. Creative Res. Thoughts (IJCRT), vol. 10, no. 8, 2022.
[15] R. Andreoli, T. Cucinotta, and D. B. De Oliveira, "Priority-driven differentiated performance for NoSQL database-as-a-service," IEEE Trans. Cloud Comput., vol. 11, no. 4, pp. 3469–3482, Oct.–Dec. 2023.
[16] A. N. Gowda and P. Gowda, "SQL vs. NoSQL databases: Choosing the right option for FinTech," SSRN, Aug. 2020. [Online].
[17] R. Andreoli, T. Cucinotta, and D. Pedreschi, "RT-MongoDB: A NoSQL database with differentiated performance," in Proc. 11th Int. Conf. Cloud Comput. Serv. Sci. (CLOSER), pp. 77–86, 2021.