Evaluation of NoSQL Databases Features and Capabilities for Smart City Data Lake Management

Nurhadi, Rabiah Binti Abdul Kadir, and Ely Salwana Binti Mat Surin

Abstract A data lake means there's an immense data resource or repository. Data lake stores enormous data and uses advanced analytics to pair data from various sources with different types of structured, semi-structured and un-structured information. The lifeblood of a smart city is data. Effective data management is not limited to data collection and storage, but must also involve shared and combined data so that it can be accessed, analyzed and used across agencies, within organizations, and even across the society. NoSQL is a form of database that is becoming increasingly common among web firms. NoSQL databases are non-tabular and store data rather than relational tables in a different way. NoSQL databases come in a variety of forms, mainly document, key-value, wide-column, and graph based on their data model. NoSQL offers easier scalability, better performance compared to conventional relational databases, and consists of many data types, such as document, key-value, wide-column, and graph. This work studies NoSQL database features and capabilities by considering four indicators, namely performance, scalability, accuracy and complexity, in order to measure the compatibility of NoSQL databases with multiple data types. The result of the experiment reveals that when accommodating massive data volumes, MongoDB is the most stable NoSQL database.

Keywords Data lake · NoSQL · Smart city database

Nurhadi · R. B. A. Kadir (✉) · E. S. B. M. Surin
Institute of IR4.0, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia
e-mail: rabiahivi@ukm.edu.my

Nurhadi
e-mail: p91334@siswa.ukm.edu.my

E. S. B. M. Surin
e-mail: elysalwana@ukm.edu.my

H. Kim et al. (eds.), *Information Science and Applications*, Lecture Notes in Electrical Engineering 739, https://doi.org/10.1007/978-981-33-6385-4_35

1 Introduction

Smart City uses data lake technology to store data in enormous capacity and there are many types of databases. One of the main and key features of big data technologies is NoSQL database. NoSQL database is able to store structured, semi-structured and unstructured data [1–3] regardless of type, format with 4Vs features: "volume, velocity, variety, veracity" [8–11] which is consisting of several model of data such as document, key-value, wide-column, and graph [4–9].

Data lake provides a scalable framework for storing large amounts of data and generating analytics that can assist multiple stakeholders in making effective decisions and developing new markets [10–13]. However, data lake still has many problems and drawbacks, one of the main issue is the use of trigger functions within ACID (Atomicity, Consistency, Isolation and Durability) to process complex online transactions and text statements [14–16]. This paper focuses on the evaluation of NoSQL database execution of data lake storage to support the use of trigger functions in managing various transactions. Four features and capabilities were evaluated in this study, namely, performance, scalability, accuracy and complexity to measure the execution of the selected NoSQL databases product i.e. MongoDB, Cassandra, Redis and Neo4j.

- **Performance**—The performance measurements in this study include several operations consisting of; select, enter, update, delete [9, 17]. Database performance can be defined as optimizing the use of resources in carrying out operations to in-crease throughput and minimizing errors, allowing as much workload as possible to be processed.
- **Scalability**—The scalability measurement in this study consists of several operations which include; data storage (write), retrieval (read), data sharding, data chacing, cluster management [17, 18].
- **Accuracy**—The accuracy measurements in this study consisted of several operations which included; import data, export data, load data. Accuracy access data is a component of data quality and refers to whether the value of data stored for an object is the correct value [19].
- **Complexity**—The complexity consisting of operations; query, function, variety. for query operations and functions used in this study include; group by, order by, select distinct, aliases, create primary [20].

The rest of the paper is organized as follows: Sect. 2 briefly discusses related work on comparison of NoSQL database features and capabilities followed by the detail description of methodology of comparison in Sect. 3. Section 4 discuss the result of the experiment. Finally, the conclusion and propose relevant expansion suggestions will be described in Sect. 5.

2 Related Work

In order to collect data, a smart city uses distinct types of electronic Internet of Things (IoT) sensors. These information and communication systems are integrated in digital technology throughout all city functions. It is a term that incorporates several ICT solutions in a safe way to manage the assets of a community, including transportation systems, waste management, water management, protection systems, information systems of municipal departments and other community services, as well as data management. With that, data lakes are a perfect place to store large amounts of data redundantly scale and store it. The concept is to connect, store and analyze the various very heterogeneous data sources, and by using NoSQL databases, data must be systematized, organized and modified for further use.

In the case of big data and real-time web applications, NoSQL databases are progressively used. NoSQL databases are particularly useful for working with vast sets of distributed data and are compliant with smart city data collection functionality. For relational data bases (RDBMS) to tackle on their own, this data explosion is proving to be too big and too complex. NoSQL databases are not constrained by the confines of a fixed schema model, unlike relational databases. NoSQL databases implement Schema on Read instead of applying Schema on Write. This makes NoSQL databases especially appropriate for the high-volume, high-variety online applications of today.

Currently, data model is the most important feature in selecting the appropriate NoSQL databases. Though, there are studies conducted by several researchers regarding the comparison of NoSQL databases based on features and capabilities such as performance, integration, and security [9, 21–23]. Meanwhile, other studies support the comparison of performance [17], integrity, cloud service criteria [8, 24], and frameworks [25]. However, those studies do not discuss the accuracy of data access and scalability, which are important features in evaluating of NoSQL databases capabilities for the purpose to select the appropriate NoSQL databases in supporting the trigger function.

Therefore, in this study we conducted a comparison of the NoSQL databases based on four features and capabilities namely; performance, scalability, accurate and complexity. In this study, we compared four NoSQL databases product such as MongoDB, Cassandra, Redis and Neo4j that used for data model document, wide-column, key-value, and graph respectively.

3 Research Evaluation Method

This section presents the research method in evaluating of NoSQL databases product for Smart City Data Lake Management based on features and capabilities. In this study, the research method was implemented based on the framework as shown in Fig. 1.

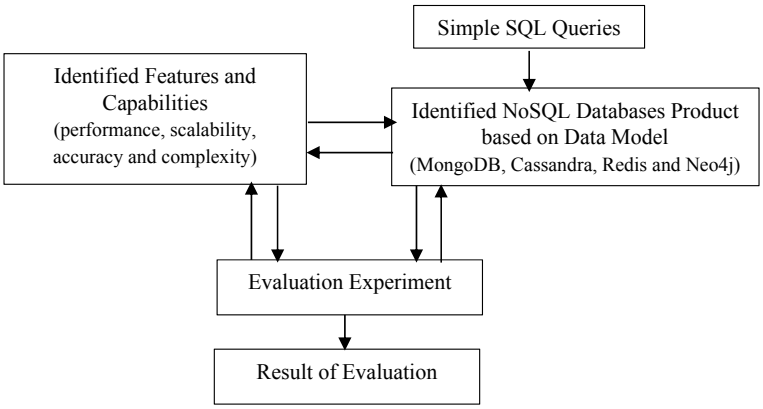

Fig. 1 Framework of research evaluation

In this experiment, 10 simple SQL queries have been tested with the combination of functions and operations such as *SELECT, INSERT, UPDATE, DELETE, IMPORT DATA, EXPORT DATA, LOAD DATA ORDER BY, GROUP BY* etc. as shown in Table 1. Each SQL query was implemented by the selected of NoSQL databases product, which is MongoDB, Cassandra, Redis and Neo4j in executing different data model: document, key-value, wide-column, and graph. The identified features and capabilities of NoSQL databases were measured and compared. The scope of the study is covering four features and capabilities as below:

1. Performance
   Capability to find, analyze and then resolve various database congestion that can impact application response times or hinder application performance.
2. Scalability
   Capability of a system to handle a growing amount of work or potential to perform more total work in the same elapsed time when processing power is expanded to accommodate growth.
3. Accuracy:
   Capability to represent the right data in a form that is consistent and unambiguous and most relevant to historical records stored on computer-accessible digital media.
4. Complexity:
   Capability of the query in evaluating the function and size of the expression.

4 Results and Analysis

The following outcomes of average scores using functions and operations available in NoSQL were obtained from the experiment, as shown in Table 1.

Table 1 Average NoSQL response score

No.	Subject and operation	Document base (MongoDB)	Key-value store (Redis)	Graph store (Neo4j)	Wide column store (Cassandra)
1	Select	95	96	90	95
2	Insert	96	96	88	96
3	Update	96	94	90	94
4	Delete	94	98	92	95
5	Import data	85	79	64	71
6	Export data	84	65	65	70
7	Load data	80	80	66	69
8	Data storage (write)	90	97	86	93
9	Retrieval (read)	90	94	85	96
10	Data sharding	89	93	84	95
11	Data chacing	90	98	85	95
12	Cluster management	91	93	85	94
13	Query (order by)	71	48	72	63
14	Query (group by)	73	50	74	65
15	Function (select distinct)	69	71	72	76
16	Function (aliases/as)	70	73	69	75
17	Function (create primary key)	68	72	69	77
18	Variety	79	60	62	69

Based on performance, scalability, accuracy, and complexity, the results were grouped, as shown in Figs. 2, 3, 4 and 5 respectively.

Performance—consists of operations; select, insert, update, delete. The average results can be seen in Fig. 2 as below.

Scalability—In Fig. 3 shows the operations such as data storage (write), retrieval (read), data sharding, data chacing, and cluster management, where virtually all types of NoSQL databases have high values.

Accuracy—includes of operations such as import data, export data, and load data. The average results were illustrated in Fig. 4, where the average score for MongoDB and Redis is the highest.

Complexity—consists of tasks such as questions, functions, combinations, and it is possible to see the results in Fig. 5.

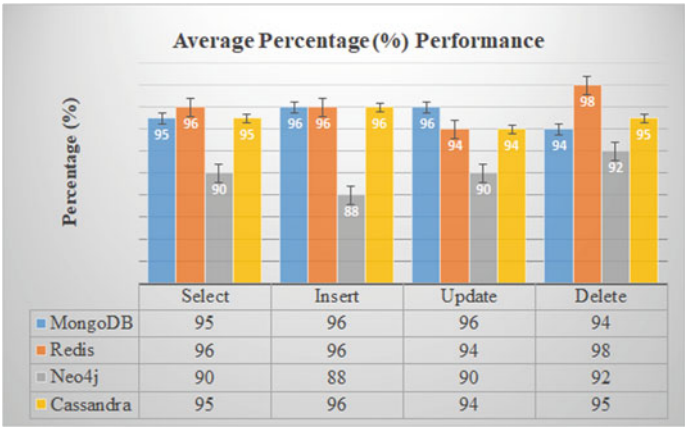

Fig. 2 Comparison of averages based on performance criteria

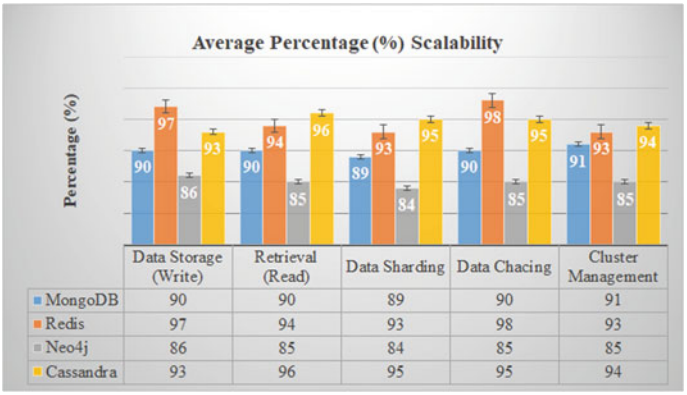

Fig. 3 Comparison of averages based on scalability criteria

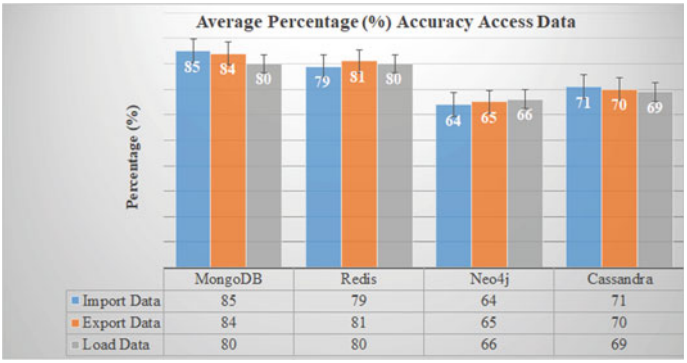

Fig. 4 Comparison of averages based on accuracy criteria

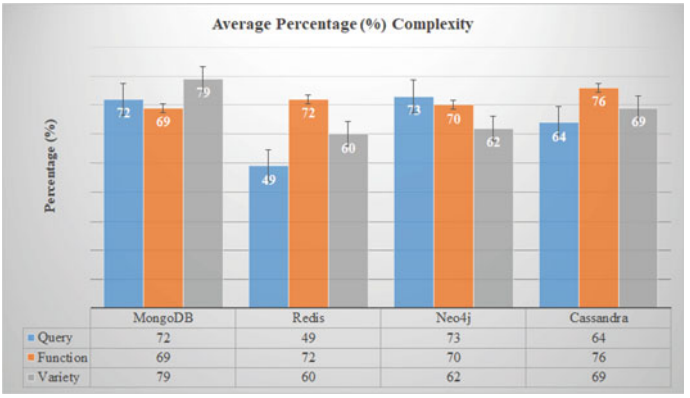

Fig. 5 Comparison of averages based on complexity criteria

The findings showed that MongoDB and Cassandra had the highest results, while Redis and Cassandra owned data scalability, while Neo4j and Cassandra owned middle-class data access accuracy. The highest percentage for complexity issues are MongoDB and Cassandra, while Redis and Neo4j are relatively poor. This demonstrates the difference between NoSQL databases by referring to performance, scability, the accuracy of database access, and complexity as shown in Fig. 6.

The complexity and accuracy given a moderate value is shown by several studies that have been carried out because it is affected by the semi-structured data format of the input. It is also possible to view the results of this study in a Table 2, where all NoSQL databases have high average output criteria. As for the complexity criterion, as shown in Table 2, all NoSQL databases have moderate values.

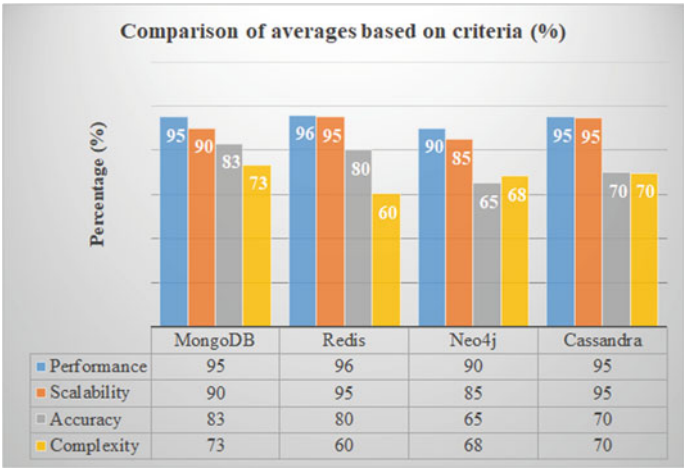

Fig. 6 Comparison of averages based on indicator criteria

Table 2 Implementation of NoSQL comparisons based on items

No.	Items	Document base	Key-value store	Graph store	Wide column store
1	Database Name	MongoDB	Redis	Neo4j	Cassandra
2	License	Open source	Open source	Open source	Open source
3	Database	Database	Database	Database	Keyspace Column Family
4	Table	Collection	Hash, List, Set, Sorter Set, and String	Label	
5	Value	Document	High	Node and edges	Rows
6	Data Source	Web Page data is open	Web Page data is open		Web Page data is open
7	Performance	High	High	High	High
8	Scalability	High	High	High	High
9	Accuracy of Data Access	High	High	Moderate	Moderate
10	Complexity	Moderate	Moderate	Moderate	Moderate

5 Conclusion

This study is to measure and compare the use of the NoSQL databases product for smart city data lake which is includes several data model. The evaluation has been conducted based on quantitative analysis through the experiment. The result of evaluation able to find the appropriate NoSQL databases product based on performance, scalability, accuracy, and complexity. The most important technical characteristics from each NoSQL databases have been studied in selecting the appropriate level of each database within the given features and capabilities. Although the NoSQL database product has the same performance, scalability and complexity scores, the accuracy shows the effect of the difference. MongoDB and Redis have high scores, although there are modest values for Neo4j and Cassandra. NoSQL databases compromise consistency to provide high performance and scalability in order to indicate the requirements that NoSQL is suitable for analyzing and accessing data across agencies based on investigation through experiments. It is in line with the success of the web-scale information system, that availability and speed are of high importance, and accuracy is compromised to some degree by sufficient NoSQL databases to meet these needs.

The future work from this study will involves optimizing algorithms and supporting complex transaction features for NoSQL databases with security and data integrity. In addition, we intend to support more categories of NoSQL databases in future testing and implementation.

Acknowledgements The authors would like to express gratitude to the University Kebangsaan Malaysia (UKM) for providing the opportunity and financial support under the project code ZG-2019-003.

References

1. Lakhe B (2016) Practical Hadoop Migration. Academic Press
2. Memoriam I, Gray J (2018) Database-Centric Scienti fi c Computing
3. Al-mandhari IS, Guan L, Edirisinghe EA (2019) Advances in Information and communication networks, vol 886. Springer
4. Patil MM, Hanni A, Tejeshwar CH, Patil P (2017) A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages. In: Proceedings of international conference IoT in social, mobile, analytics and cloud on I-SMAC, pp 325–330
5. González-Aparicio MT, Ogunyadeka A, Younas M, Tuya J, Casado R (2017) Transaction processing in consistency-aware user's applications deployed on NoSQL databases. Hum Cent Comput Inf Sci 7(1)
6. Patil MM, Hanni A, Tejeshwar CH, Patil P (2017) A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages. In: Proceedings of international conference on I-SMAC (IoT in social, mobile, analytics and cloud) (I-SMAC) 2017, pp 325–330
7. Chen JK, Lee WZ (2019) An introduction of NoSQL databases based on their categories and application industries. Algorithms 12(5)
8. Tool B, Chakraborttii C (2019) Performance evaluation of NoSQL systems using yahoo cloud serving performance evaluation of NoSQL systems using yahoo cloud serving benchmarking tool. February 2015
9. Patil MM, Hanni A, Tejeshwar CH, Patil P (2017) A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing-Sharding in MongoDB and its advantages. In: *Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017*, pp. 325–330
10. Challenges HD, Gupta S, Giri V (2018) Practical enterprise data lake insights. Academic Press
11. Phyu KP, Shun WZ (2018) Data lake : a new ideology in big data era. In: ITM Web Conference 17, vol 03025, pp 1–11
12. Mathis C (2017) Data lakes. Datenbank-Spektrum
13. I. Nosql, R. Database, B. Data, and CC ((2018)) Transactions "Romulo Alceu Rodrigues, Lineu Alves Lima Filho, Gildarcio Sousa Gonc ¸ alves, Lineu F.S. Mialaret, Adilson Marques da Cunha, and Luiz Alberto Vieira Dias, pp 443–451
14. AE Lotfy, AI Saleh, HA El-Ghareeb, HA Ali A middle layer solution to support ACID properties for NoSQL databases. J King Saud Univ Comput Inf Sci 28(1):133–145
15. Davoudian A, Chen L, Liu M (2018) A Survey on NoSQL Stores. ACM Comput Surv 51(2):1–43
16. Schreiner GA, Duarte D, dos S. Mello R (2019) When relational-based applications go to NoSQL databases: A survey. INF 10(7):1–22
17. Flores A, Ramirez S, Toasa R, Vargas J, Barrionuevo RU, Lavin JM (2018) Performance evaluation of NoSQL and SQL queries in response time for the e-government. In: 2018 5th international conference eDemocracy eGovernment, ICEDEG 2018, pp 257–262
18. Mohan A, Ebrahimi M, Lu S, Kotov A (2016) A NoSQL data model for scalable big data workflow execution. In: Proceedings of 2016 IEEE international congress on big data, big data congress 2016, pp 52–59

19. Li C, Gu J (2019) An integration approach of hybrid databases based on SQL in cloud computing environment. Softw Pract Exp 49(3):401–422
20. Sánchez-de-Madariaga R, Muñoz A, Castro AL, Moreno O, Pascual M (2018) Executing complexity-increasing queries in relational (MySQL) and NoSQL (MongoDB and EXist) size-growing ISO/EN 13606 Standardized EHR databases. J Vis Exp 133:1–11
21. Hajoui O, Dehbi R, Talea M, Batouta ZI (2015) An advanced comparative study of the most promising NoSQL and NewSQL databases with a multi-criteria analysis method. J Theor Appl Inf Technol 81(3):579–588
22. Sánchez-de-Madariaga R, Muñoz A, Castro AL, Moreno O, Pascual M (2018) Executing complexity-increasing queries in relational (MySQL) and NoSQL (MongoDB and EXist) size-growing ISO/EN 13606 standardized EHR databases. J Vis Exp 133
23. Kolonko K (2018) Master of Science in Software Engineering Performance comparison of the most popular relational and non-relational database management systems
24. Kumar MS, Prabhu J (2018) Comparison of NoSQL database and traditional database-an emphatic analysis. JOIV Int J Inf Vis 2(2)51
25. Reniers V, Van Landuyt D, Rafique A, Joosen W (2019) Object to NoSQL database mappers (ONDM): a systematic survey and comparison of frameworks. Inf Syst 85:1–20