International Journal on Advanced Science Engineering Information Technology
Vol.7 (2017) No. 5
ISSN: 2088-5334

Performance Comparison of CRUD Operations in IoT based Big Data Computing

Jung Yeon Seo[#], Dae Won Lee[*], Hwa Min Lee[#]

[#] *Department of Computer Software & Engineering, Soonchunhyang University, SoonchunhyangRo 22, Asan, 31538, Korea*
E-mail: busyppp@gmail.com, leehm@sch.ac.kr

* *Department of Computer Engineering, Seokyeong University, SeokyeongRo 124, Seoul, 02713, Korea*
E-mail: daelee@skuniv.ac.kr

***Abstract*— Nowadays, due to the development of mobile devices, the kinds of data that are generated are becoming diverse, and the amount is becoming huge. The vast amount of data generated in this way is called big data. Big data must be processed in a different way than existing data processing methods. Representative methods of big data processing are RDBMS (Relational Database System) and NoSQL method. We compare NoSQL and RDBMS, which are representative database systems. In this paper, we use MySQL query and MongoDB query to compare RDBMS and NoSQL. We gradually compare the performance of CRUD operations in MySQL and MongoDB by increasing the amount of data. MongoDB sets index and compares it again. Through result of these operations is to choose a database system that fits the situation. This makes it possible to design and analyse big data more efficiently.**

***Keywords*— NoSQL; RDBMS, MongoDB; MySQL; big data; performance calculation.**

I. INTRODUCTION

With the advent of the IoT era, the need for real-time analysis of Big Data is increasing rapidly. In online analysis, there is no distinction between data generation time and analysis time, whereas 'real time analysis' is performed as close as possible to the point at which data is generated. The criteria for 'real time' are variously defined in minutes, seconds, less than 1 second depending on the nature of the work, but the requirements for the 'real time' are becoming more diverse and strengthened at the same time.

The real-time analysis of the IoT era deals with time series data generated from large number of sensors and social media, and especially log data generated from various machines is the main ingredient. Machine data is the fastest growing segment of the Big Data and is characterized by faster and more accurate real value in that it can identify various transactions and customer behaviour, sensor records, machine behaviour, security threats and fraud.

To store, process, and analyze the big data generated in this way, a big data method that is different from existing data storage, processing, and analysis methods is needed. Big data processing methods are typically RDBMS (Relational Database Management Systems) and NoSQL. A relational database (RDB) is a very simple database of principles that tabulates the simple relationships between keys and values. A relational database management system (RDBMS) is a system for managing an RDB [1][2]. The NoSQL database provides a mechanism for storing and retrieving data using a less restrictive consistency model than traditional relational databases. Motivations for this approach include design simplification, horizontal scalability, and fine control. The NoSQL database is a highly optimized key value storage space for simple retrieval and addition operations, with the goal of providing significant performance benefits regarding latency and throughput. NoSQL databases are widely used for commercial use of big data and real-time web applications. Also, the NoSQL system is sometimes referred to as "Not only SQL" in the sense that it emphasizes the fact that SQL-based query languages can be used [1].

However, due to the property of RDBMS, performance and cost issues have been raised in processing big data. NoSQL has been proposed as an alternative to this problem. NoSQL can process data less restrictively than RDBMS.

In this study, we compared the performance difference between MongoDB and MySQL. MongoDB is a representative example of NoSQL and MySQL is a representative example of RDBMS. We also compare the speed of operations with and without MongoDB index. We study how to design, analyse and process big data suitable for the system that user wants to design. Through the results of the study, we study a more convenient method for processing big data using NoSQL.

II. MATERIAL AND METHOD

With the development of smart devices and IoT (Internet of Things) devices, data is increasing. The amount of data generated in communication networks between objects due to the development of communication technologies such as M2M and IoT is increasing. Accordingly, the amount of data to be generated is numerous and varies. These data are called big data. With the development of IoT, big data continues to be generated over time. The generated data is important information for analysis and prediction. As the interest in big data increases, research on processing methods is increasing. Particularly, comparative studies on NoSQL and RDBMS which are widely used are actively made.

A. Method

1) MongoDB

There is a growing interest in database management systems other than the existing relational models. There is a database management system called NoSQL, which is a system that does not use SQL. MongoDB is an open source document-oriented database that stores data as a collection of JSON-like documents among NoSQL projects [1][3]. MongoDB has the following five features [4].

- Various data models

MongoDB is a document-oriented database. A document-oriented model allows a complex hierarchical relationship to be represented as a single record. This model is suited for developers using the latest object-oriented language.

MongoDB does not have a predefined key in the document or no fixed schema. Because of there is no schema change, a large amount of data conversion is not necessary. A new key or a missing key can be processed at the application level without having to force the same form for all data. This provides greater flexibility when dealing with a constantly changing data model.

- Easy expansion

MongoDB was originally designed with distributed expansion in mind. The document-oriented data model automatically distributes data across multiple servers. Automatically redistribute documents to control the amount and load of data in the cluster. Because developers can concentrate more on programming, not database expansion. When needing more capacity, just add the new device to the cluster and let the database clean up everything else [5].

- Various functions

MongoDB provides distributed extensions with useful functionality of relational databases such as range queries, auxiliary indexes, and sorting functions. MongoDB also provides some features such as built-in MapReduce-based aggregation operations and geospatial indexing.

- High performance

MongoDB's main goal is an outstanding performance. This greatly influenced the design. MongoDB basically uses the binary wire protocol rather than the overhead protocol like HTTP / REST of the main protocol that interacts with the server. It also pre-allocates additional space dynamically in the document to maintain consistent performance even when storage is used more. The basic storage engine used a memory-mapped file and passed the responsibility of memory management to the operating system. It provides a dynamic query optimizer that remembers the fastest way to execute a query.

- Easy management

MongoDB is designed to manage the servers themselves, so a user can easily manage the database. Because there is little to do except for starting the database. If the master server does not work, MongoDB automatically switches the backup slave to master. In a distributed environment, just inform the cluster that a new node has been created, it automatically adds and configures the node.

MongoDB can dynamically change the schema of the data, is fast and easy to expand. Therefore, it is being used effectively in web-based services and mobile services, in which the needs of customers are changing from time to time.

2) MySQL

Relational databases have long been studied and developed to provide reliable and reliable services such as large data processing, transactions, locking, and security mechanisms, and efficiency in data processing is widely recognized [6]. Among these relational databases, MySQL is used worldwide and is advantageous for large capacity processing.

MySQL has the following features [7-8].

- Openness

MySQL is open software. MySQL's SQL syntax is based on ANSI SQL2. The database engine can run on a variety of platforms, including Windows 2000, Mac OS X, Linux, FreeBSD, and Solaris.

- Application Support

MySQL provides APIs for almost all languages. In particular, it is possible to create database applications that access MySQL with C, C ++, Java, Python, PHP, and others.

- Multiple database connections

It is possible to build a MySQL query that can connect tables in different databases.

- External connection support

MySQL supports both external connections using both ANSI and ODBC syntax.

- Internationalization

MySQL supports several character sets. It also supports sorting for different character sets and is easily configurable. Error messages can also be displayed in several languages.

But most importantly, MySQL is fast and inexpensive. The most important feature is the speed and the low price compared to other relational databases.

A Study on Data Input and Output Performance Improvement of MongoDB and PostgreSQL in the Big Data Environment compares the performance of PostgreSQL and MongoDB by inputting, querying, updating, and deleting data [9]. As a result, we confirmed that MongoDB's operation speed is faster overall. From a performance perspective, MongoDB can see that designing with an unstructured data model yields better performance than designing with a relational data model.

Research on Utilizing NoSQL by Comparison of Processing Large Scale Data in MongoDB and MySQL [10] compared MongoDB and MySQL's large-scale data processing performance measurements. As a result, MongoDB results are compared with about 80% performance of a large capacity data. In large scale distributed environments, it is expected that results will be further increased. However, if you are running a system that focuses on data consistency and data integrity instead of MongoDB, which has not yet had enough data and experience to accumulate, it has been suggested that you should use MySQL.

Also, according to Performance evaluation of MongoDB application for small scale marina operation system [11], Performance Analysis about Structurally Improved MariaDB and MongoDB [12], and so on, MongoDB has proved to be superior to RDBMS in insertion, retrieval, modification, and deletion operations. Depending on the structure of the system, the required database system is different. RDBMS is required in the stable system, and MongoDB is required in the system where speed is important.

The following Table I compares the features of MySQL and MongoDB [13].

TABLE I
FEATURES COMPARISON BETWEEN MYSQL AND MONGODB

Feature	MySQL	MongoDB
Rich Data Model	X	O
Dynamic Schema	X	O
Typed Data	O	O
Data Locality	X	O
Field Update	O	O
Easy Programming	X	O
Complex Transactions	O	X
Audit Function	O	O
Automatic Shading	X	O

The following Table II compares MySQL and MongoDB terms [14].

TABLE II
TERM COMPARISON BETWEEN MYSQL AND MONGODB

MySQL	MongoDB
Database	Database
Table	Collection
Index	Index
Row	BSON Document
Column	BSON Field
Join	Embedded documents and linking
Primary key	Primary key
Group by	Aggregation

B. Related Works

[15] poses the problem that the reliability of information is low even though the amount of information is rapidly increasing. Therefore, they develop a search engine that can provide satisfactory search results reflecting user's intention. It produces unsatisfactory results when the early search results are not appropriate. Helps users select appropriate terms from vast amounts of data.

Between the massive amount of data, the user must provide the desired product more efficiently. However, [16] poses the problem that the cooperative filtering algorithm provides low accuracy. Therefore, in this paper, to improve the predictive ability of the Collaborative Filtering technique, they propose a recommendation system that utilises opinion mining based not only on quantitative data but also on reviews after a purchase.

Programming mistakes can be time consuming for software developers and can pose a serious risk to customers. [17] uses sensors to predict programmers' work difficulties and programmer-level expertise. It attaches bio-sensors to programmers and collects and analyses data. The amount of data collected through the sensor is vast.

It is efficient to analyse these studies using the Big Data System used in this paper. By using the method that is appropriate to the situation, the efficiency of the system can be increased.

III. RESULT AND DISCUSSION

To compare the performance of MongoDB and MySQL, the following system environment was constructed. A computing device for constructing a system environment was implemented using a virtual server. Performance comparison between MySQL version 5.7 and MongoDB version 3.2. The system performance of the virtual server is shown in Table III.

Compare the performance of MySQL, the relational database system, and MongoDB, the NoSQL. Add indexes to make MongoDB's queries more efficient and compare them one more time. Performance comparison compares the computation time after executing the CRUD (Create, Read, Update, Delete) operation. The data were tested with US aviation data for 2008. There are about 70,000 aviation data. Thus, we performed operations with 100, 200, 300, 800, and 1000 data, respectively.

TABLE III
EXPERIMENT SERVER SPECIFICATION

System	Specification
CPU	Intel Core I5-35300 CPU @ 3.3Hz
OS	Linux 3.10.0 (64bit)
RAM	4G Byte

A. Create Operation

The create operation measures the time taken to add each data. The execution time of the create operation is the same as the result of Table IV.

TABLE IV
CREATE OPERATION EXECUTION TIME (UNIT: SECOND)

Number of Data	**MySQL**	**MongoDB**	**MongoDB (Index)**
100	0.04	0.0226	0.0100
200	0.05	0.0224	0.0104
300	0.05	0.0223	0.0118
500	0.08	0.0223	0.0151
800	0.07	0.0226	0.0175
1000	0.08	0.0225	0.0202

Table IV is shown in the chart as Fig. 1.

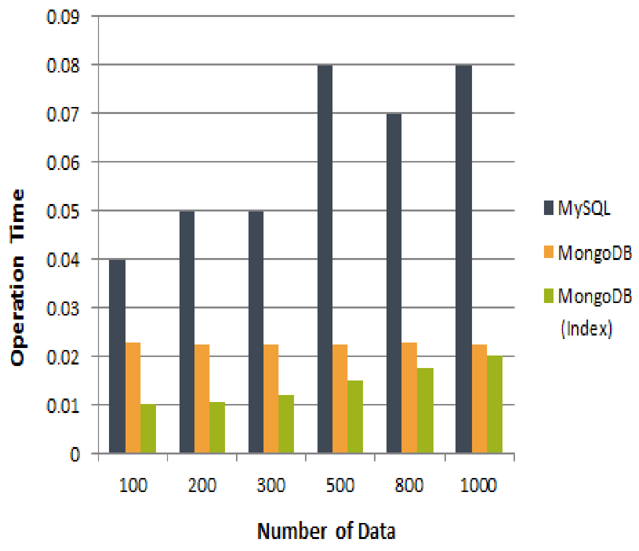

Fig. 1 Create Operation Execution Time Chart

Refer to Table IV and Fig. 1, they can be seen that MongoDB create operation speed is about four times faster than MySQL create operation speed. In addition, it is the same MongoDB, but the performance difference is about two times or more depending on the existence of Index.

B. Read Operation

The read operation measures the time taken to query the total air delay data, about 70,000 data, including the data added by the create operation in each database system. In the case of read operation, since the time is not measured when a small amount of data is retrieved, the criterion is based on the number of times the total data is retrieved. The execution time of the read operation is the same as the result of Table V.

TABLE V
READ OPERATION EXECUTION TIME (UNIT: SECOND)

Views	**MySQL**	**MongoDB**	MongoDB (Index)
1	0.05	0.0224	0.0167
3	0.04	0.0223	0.0144
5	0.04	0.0224	0.0136
7	0.04	0.0225	0.0134
9	0.05	0.0224	0.0146
11	0.04	0.0226	0.0143
13	0.04	0.0226	0.0147
15	0.04	0.0226	0.0144

Table V is shown in the chart as Fig. 2.

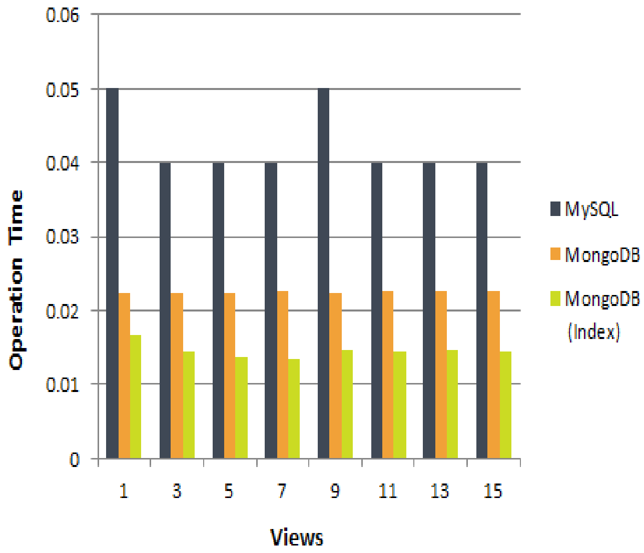

Fig.2 Read Operation Execution Time Chart

Refer to Table V and Fig. 2, they can be seen that MongoDB read operation speed is about twice faster than MySQL read operation speed. Also, it is the same MongoDB, but it varies by about 1.5 times depending on the existence of Index.

C. Update Operation

The update operation measures the time taken to correct each data. The execution time of the update operation is the same as the result of Table VI.

TABLE VI
UPDATE OPERATION EXECUTION TIME (UNIT: SECOND)

Number of Data	**MySQL**	**MongoDB**	**MongoDB (Index)**
100	0.04	0.0226	0.0100
200	0.05	0.0224	0.0118
300	0.05	0.0223	0.0145
500	0.08	0.0223	0.0157
800	0.07	0.0226	0.0185
1000	0.08	0.0225	0.0195

Table VI is shown in the chart as Fig. 3.

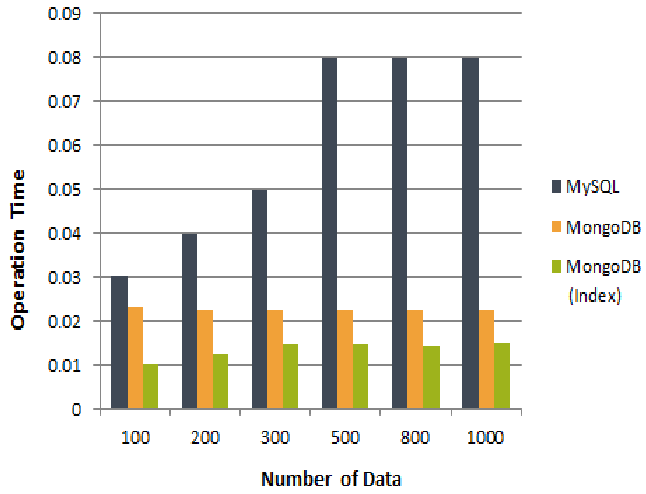

Fig.3 Update Operation Execution Time Chart

Refer to Table VI and Fig. 3, they can be seen that MongoDB update operation speed is about twice faster than MySQL update operation speed. In addition, it is the same MongoDB, but the performance difference is about two times or more depending on the existence of Index.

D. Delete Operation

The delete operation measures the time taken to delete each data. The execution time of the delete operation is the same as the result of Table VII.

TABLE VII
DELETE OPERATION EXECUTION TIME (UNIT: SECOND)

Number of Data	MySQL	MongoDB	MongoDB (Index)
100	0.03	0.0231	0.0100
200	0.04	0.0225	0.0122
300	0.05	0.0223	0.0146
500	0.08	0.0223	0.0146
800	0.08	0.0223	0.0142
1000	0.08	0.0224	0.0151

Table VII is shown in the chart as Fig. 4.

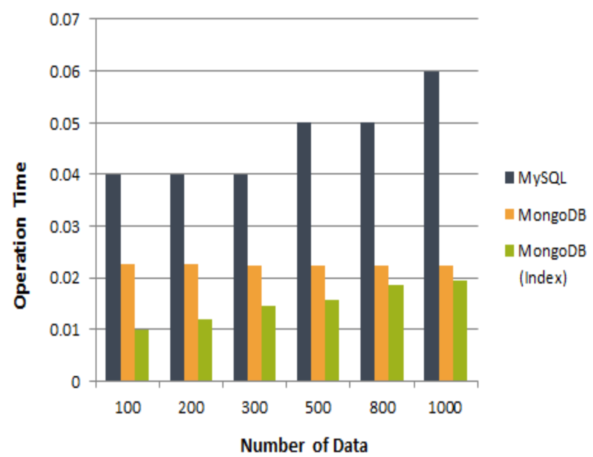

Fig.4 Delete Operation Execution Time Chart

Refer to Table VII and Fig. 4, they can be seen that MongoDB delete operation speed is about 4 times faster than MySQL delete operation speed. In addition, it is the same MongoDB, but the performance difference is about two times or more depending on the existence of Index.

In this paper, we compare the computation speeds for create, read, update, and delete operations of MySQL and MongoDB. Experimental results show that MongoDB's operation speed is at least twice as fast as all operations. This is because MySQL consumes more time than MongoDB to guarantee ACID (Atomic, Consistent, Isolated, Durable), the nature of the database.

In addition, it is the same MongoDB, but operation speed is about two times or more depending on the existence of Index. Because of MongoDB uses a collection scan method. This method takes a long time if there is a lot of data. To improve these disadvantages, Index is generated. Create a B-Tree with the set key value of the data and scan it. Because the speed of the operation is faster.

Thus, the results of the experiment indicate that MongoDB is preferred over MySQL for handling large amounts of data.

IV. CONCLUSIONS

MongoDB is more readable and writable than MySQL, has flexible scalability, and has no schema structure, so freely change the format of data. This is a trend that many companies are now transforming from relational database systems to NoSQL. MongoDB is designed to handle the hard parts of traditional relational database systems. However, there are disadvantages compared with relational database systems. The transaction function is weak compared with the relational database system, and if a failure occurs during the update, the data may be lost, and the memory space is large. Relational databases and NoSQL are suitable solutions in some situations. We recommend the introduction of NoSQL in database systems where computational performance is good and speed is important, and recommend the introduction of relational database systems in a structured and accurate database system. When designing the database according to the requirements, a more suitable database system is selected and introduced.

Future research will need a way to quickly and safely store large amounts of data by compensating for backward performance (Atomic, Consistent, Isolated, Durable) when switching from relational database systems to NoSQL. We will also be able to contribute to the development of analytical techniques for large data environments by performing operations on unstructured data models, adding research on the possibility of distributing NoSQL, and complementing safety.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2017R1A2B4010570) and the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) (IITP-2016-R0992-16-1006) supervised by the IITP (Institute for

Information & communications Technology Promotion).
*Corresponding Author: HwaMin Lee(leehm@sch.ac.kr)

REFERENCES

[1] Heripracoyo S, Kurniawan R. Big Data Analysis with MongoDB for Decision Support System. TELKOMNIKA (Telecommunication Computing Electronics and Control). 2016;14(3);1083.

[2] Alenazi SR, Ahmad K. Record Duplication Detection in Database: A Review. International Journal on Advanced Science, Engineering and Information Technology. 2016;6(6).

[3] Boicea A, Radulescu F, Agapin LI, editors. MongoDB vs Oracle-Database Comparison. EIDWT;2012.

[4] Chodorow K. MongoDB: the definitive guide: " O'Reilly Media, Inc."; 2013.

[5] Liu Y, Wang Y, Jin Y, editors. Research on the improvement of MongoDB Auto-Sharding in cloud environment. Computer Science & Education (ICCSE), 2012 7th International Conference on; 2012: IEEE.

[6] Xun Li, SangBong Yoo. Design of Relational Database for Efficient Storing and Processing of Ontology. The Journal of Korean Institute of Information Technology. 2010;8(9):143-51.

[7] King T, Reese G, Yarger RJ. Managing and using mysql: O'Reilly; 2002.

[8] Franke C, Morin S, Chebotko A, Abraham J, Brazier P, editors. Distributed semantic web data management in HBase and MySQL cluster. Cloud Computing (CLOUD), 2011 IEEE International Conference on; 2011: IEEE.

[9] Jung M-G, Youn S-A, Bae J, Choi Y-L, editors. A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment. Database Theory and Application (DTA), 2015 8th International Conference on; 2015: IEEE.

[10] Kim Eun-Ki. Research on utilizing Nosql by comparison of processing large scale data in MongoDB and MySQL. M. Eng. Thesis:Soongsil Univ; 2016.

[11] Sung Hong-Ki. Performance evaluation of mongoDB application for small scale marina operation system. M. Eng. Thesis:Dongmyung Univ; 2015.

[12] Cha Seung-Hoon. Performance analysis about structurally improved MariaDB and MongoDB. M. Eng. Thesis:Soongsil Univ; 2014.

[13] Győrödi C, Győrödi R, Pecherle G, Olah A, editors. A comparative study: MongoDB vs. MySQL. Engineering of Modern Electric Systems (EMES), 2015 13th International Conference on; 2015: IEEE.

[14] Padhy RP, Patra MR, Satapathy SC. RDBMS to NoSQL: reviewing some next-generation non-relational database's. International Journal of Advanced Engineering Science and Technologies. 2011;11(1):15-30.

[15] Automatic extraction of user's search intention from web search logs. Multimedia tools and applications;61(1);145-162;Springer Science+Business Media, LLC 2011

[16] Developing a hybrid collaborative filtering recommendation system with opinion mining on purchase review. Journal of Information Science; 2017

[17] Mining biometric data to predict programmer expertise and task difficulty. Cluster Computing;1-11;Springer Science+Business Media New York 2017