

MongoDB and NoSQL Databases

Vidushi Jain
VIT University-Vellore

Aviral Upadhyay
VIT University-Vellore

ABSTRACT

With the information becoming drastically and the structure of information turning out to be progressively adaptable, MongoDB has supplanted the relational database in numerous applications. In many applications it has outperformed the traditional SQL databases, and hence in this paper there would be modeling about the changes from SQL no NoSQL database also bringing out it's advantages and disadvantages.

Keywords

NOSQL Database, MongoDB, Comparison between SQL and NOSQL databases

1. INTRODUCTION

MongoDB was released in 2009, and is written in C++. It is the most popular noSQL database system available. As a document-oriented database, MongoDB holds an arrangement of collections, which have comparable implications as the table of relation database. [3] An accumulation of MongoDB contains numerous records and does not have a predefined outline. Every record is an arrangement of fields, which is remarkably distinguished by the ID field, and it additionally has an adaptable structure. This paper starts by a simple survey on current NoSQL database systems and then it emphasizes on the major difference between current day SQL systems and the one with MongoDB. The way to bring about the change from SQL to MongoDB and then security issue faced with MongoDB that might have been absent in SQL system is mentioned in brief. Although, MongoDB is new but it is an open source and is efficient .It is also being used in many modern database projects and principles [3].

2. SURVEYS ON NOSQL DATABASE

2.1 Reasons

Many organizations came together, which had more demands and aspiration from the database technology in the given aspects [3]:

2.1.1 Efficient big data storage

There are very large applications such as the search engines which need an efficient database to meet the storage requirement and which is able to respond to some billions of traffic.

2.1.2 Incremented consistency, regularity, unity

With the ever incrementing number of consecutive and continuous requests and data the database needs to support data expansion and then at the zenith make sure of a not failing service.

2.1.3 Confined capacity

The present relational databases have a very confined data capacity and they are unable to support the massive data like in a search engine.

2.1.4 Lower management and operational price

As the data grows widely, the database prices that contain hardware prices, software prices and operating prices have

incremented. So, a reduction in these prices is much needed.

2.1.5 2.1.5 Slow reading and writing

A relational database has a well-defined logic complexity. As the data increments; it is more vulnerable to dead locks and other issues that leads to less efficiency.

2.1.6 Expansion difficult

Multi table correlation mechanism, which is present in relational database

2.2 Features of NoSQL database

2.2.1 Features of NoSQL

It reads and writes data fast. It is simpler to expand and it also supports massive storage with a low price.

2.2.2 Data model

It has a key value data model which means there is a value which leads to key, the structure is simple with an increase in query speed .It has a column oriented database that is a table but it does not support table association that is a join. Each and every column is the index of the database. There is a database in which there are collections and in collections columns and in columns documents are there.

Classification is done on the basis of CAP theorem.

2.2.3 CAP Theorem

It says that there could be three properties of a shared data system that is regularity, scalability and availability, and tolerance that is there in network partitions.

2.3 Mainstream NoSQL database

2.3.1 Key Value databases

One of the most efficient key value databases is Flare. It was made by the Green.jp in Japan. It has incremented scalability. Flare added a node server that is before the data servers so that data servers could be handled at the back .In return, users can add or delete data servers with the help of flare. Other efficient key value database is the TC namely Tokyo Cabinet. It is a very high performance engine. Tokyo Cabinet also ensures high performances of read and write at the same time, it relies on data firmness mechanism. Its main drawback is that when there is massive data, the concurrent write performance drops at a great extent.

2.3.2 Column oriented database

It manages high performance of analyzing data and business intelligence processing. It includes Cassandra and Hyper table. Cassandra is a database of Facebook that is an open source. Its highly symmetric in nature. It manages large amount of structured data and provides highly reliable services with least amount of failure. It does not have a fully-fledged relational database model but it has a model that is simpler and has control over data layout and format. It was Cassandra that implemented the Inbox search feature of Facebook. It has a key value store. Key here maps to values, which are widely distributed into column families. Here,

columns can only be appended to keys, which are already specified, and which implies each and every key has a different number of columns in a particular family. It also has super columns, a column whose values are columns that is ideally a sorted array of columns linked associatively. The values from column key for each family are stored together. This defines Cassandra as big hybrid data manager between a columns oriented DBMS and a row-oriented store.

2.3.3 Document Database

It does not concern read or writing at the same time and there performance. They help in managing massive data and effective query performance. It includes MongoDB [1], CouchDB. CouchDB is a flexible database that can tolerate faults. It has some drawbacks that its reading and writing concurrent is less. MongoDB on the other hand is a non-relational database and has a very high access on massive data. It has a very efficient query language and supports complicated data types. It is a cluster of nodes where not every node is the same that is it lack symmetry.

NoSQL is not that much efficient in cloud computing and research had to be strengthened in these areas. [5]

3. DIFFERENCE BETWEEN ORACLE (SQL) AND MONGODB (NOSQL)

The comparison is done, because Oracle is the most popular system used and has been around for ages whereas, MongoDB is relatively new, but still has been used by many projects and products.

3.1 Main Difference

The principle contrast is that NoSQL is a class of database motors that don't bolster SQL keeping in mind the end goal to accomplish execution or unwavering quality elements that are contrary with their adaptability of SQL.[7] Utilizing NoSQL forces some engineering imperatives that are depending of what one is doing with the information. It is uncommon for a whole existing SQL-driven application to be moved in its integrality over to NoSQL still some algorithms like the mentioned above can be used to translate a database in SQL to MongoDB. As mentioned the above paper, MongoDB doesn't store data in fixed tables but have a rather flexible outline as compared to SQL. Also, MongoDB is document-oriented schema-less where as Oracle as we all know is a relational database. The input data size for MongoDB is 16MB contrasted to 4KB of Oracle, which is indeed too small to be modern. The major difference is that MongoDB is open source and free whereas Oracle is a service one has to pay for, but is the mainstream and have been for long.

3.2 Syntax Difference

Oracle Database utilizes regular SQL dialect. It utilizes information control proclamations as SELECT, UPDATE, DELETE.

MongoDB utilizes functions for the operations of including new records, upgrading and erasing existing records.

For PL/SQL like statements, MongoDB uses call back functions. [1]

As there are no relations between two tables in MongoDB one can drop table without worrying about the fact that it may create exceptions and other errors. In MongoDB, the formation of the table is made on the first insertion operation. As a result, of the adaptability of the MongoDB accumulation, the structure doesn't need to be altered.

As to inclusion, Oracle Databases have different sorts of imperatives on tables. When one is attempting to add another record they must be mindful so as not to damage any of the limitations. Then again, MongoDB is more adaptable with the information. One doesn't have any imperatives with respect to the information from the collections. This is the significant contrast between a SQL Database and a NoSQL one.

In NoSQL databases information is put away utilizing collections. MongoDB collections have no imperative with respect to the information put away in them. The collections don't have a settled structure. The fields can have diverse information sorts. NoSQL differs from SQL as in SQL data type for a column is fixed and in it only the specified value can be entered, otherwise an error arises. In MongoDB, to find a particular data find() function is used and here in MongoDB, there aren't any tables. The information is put away utilizing collections of information. This collection doesn't have an altered structure. The things put away in these collections can have distinctive fields. MongoDB uses the concept of class, i.e. all the functions are member functions of a class and are called using '.' operator of C++. Example, db.tt.find().sort('field_name'). To completely understand MongoDB, one has to go through all the documentations provided with it, and is available on docs.mongodb.com.

3.3 Time Comparison

While switching to MongoDB, one not only has to consider time of execution but it plays a major part in the decision of switching and can be one of the deciding factor.

MongoDB is more productive while embedding a lot of information. It takes too little to long to embed less number of records. [7]

Oracle Database bargains extremely pleasant with little measure of information, yet when there is a discussion on extensive sums, more noteworthy than 10000 records, the time spent on this operation is greater than MongoDB by a significant amount. The story is the same for updating a record, while Oracle takes a lot of time, updating few or many records, MongoDB takes no time compared to Oracle updating records irrespective of how many are updated the average time to update a record is around 1ms. The time taken in deleting a record is huge, MongoDB beating Oracle. While time taken rises exponentially in Oracle it remains same for MongoDB, no matter what is the amount of records that need to be deleted.

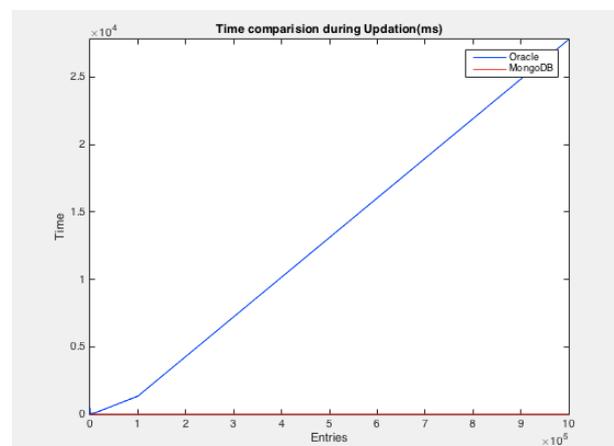


Figure 1. Time Taken by an SQL (Oracle) and NOSQL (MONGODB) to Update Records in a Database, Y Axis is Time in Seconds and X Axis Denotes Number of Records Updated

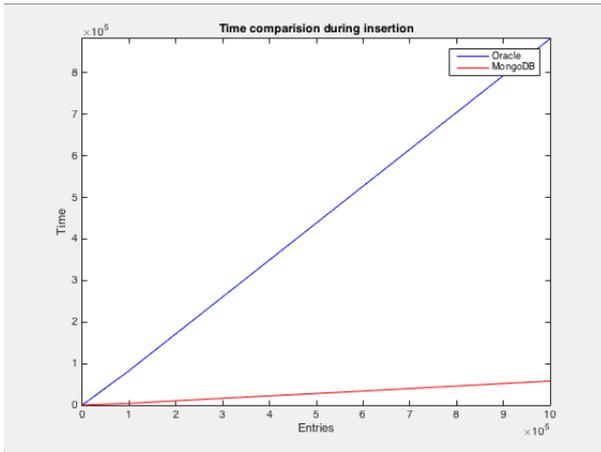


Figure 2. Time Taken by an SQL (Oracle) and NOSQL (MONGODB) to Update Records in a Database, Y Axis is Time in Seconds and X Axis Denotes Number of Records Updated

4. TRANSFERRING SQL DATABASE TO MONGODB

The use of a MongoDB instead of SQL database may bring about change in many aspects. The steps to transfer are very easy but cautiousness is needed while doing so. Thereby, there are changes initiated in the model.

- i. Considers the inquiry attributes and information qualities of the relational database.
- ii. Plans a model change calculation in light of depiction labels and activity labels.
- iii. Consequently relocates the information into MongoDB in view of the consequence of model change, and
- iv. Builds up a helpful device. There are some trials, which have been outlined. These trials demonstrate that utilizing the above mentioned approach can accomplish a superior read execution.

4.1 Challenges

4.1.1 Model transformation Challenge

Because of the absence of specialized devices, a considerable measure of model change systems depends on database's head's experience. Individuals regularly outline the physical model of MongoDB physically in view of the current relational database. It is exceptionally badly designed when confronting with a mind-boggling structure of connections. Also, as one increases the number of related table, performance may be increased but data redundancy is induced. Hence, we need to make a trade off between numbers of relations included.

4.1.2 Data Migration challenge

Presently, information relocation apparatuses are utilizing extremely basic movement methodologies, such as moving every table of relational database to a particular collection of MongoDB.

5. LIMITATIONS OF SQL DATABASE

The motivation behind moving information from social database to NoSQL database is to acquire a superior execution. Consequently, there is a need to think about the impediments of the current relational database. After that, the upsides of NoSQL database can be taken to conquer the deficiencies of social database. MongoDB doesn't support

joins, but it does have hierarchy and nested tables.

5.1 Conversion

All SQL systems maintain a log, so as to store about the commands executed and implemented by the user. In this paper, all these tags are extracted from the log and divided into categories such as.

- Frequent Joins: It implies two tables as often as possible execute join operation and the query speed is not sufficient.
- Big Size: Data is large.
- Frequent Modify: Data is modified frequently
- Frequent insert: New data is added frequently.

Depiction labels and the relationship of social database creates "Action labels". The fundamental motivation behind creating action labels is to utilize installing or references amid the model change and route the nesting heading.

The ER model of the database in relational form is used to form an outline of database in MongoDB. In spite, of the fact that MongoDB does not bolster join operation, it can store organized information by references. Consequently, Reference in MongoDB could be utilized to depict Relationship in social database. As mentioned earlier, some time recently, a considerable measure of movement devices do that way. The main distinction is that every report will have an id-field to extraordinarily distinguish it. Three algorithms were specified to convert from relational database to MongoDB, which are: Conversion of ER model of relational database to physical model of MongoDB. But as there is a need of action tags, an algorithm to generate action tags from the ER model, and a third algorithm to transfer the data is used.

6. SECURITY ISSUES IN NOSQL DATABASE

A very disastrous con is NoSQL databases have compromised with the security and consistency for performance improvement. Here, is a glance of the techniques that are implemented for security in NoSQL databases that are namely Cassandra and MongoDB. The main benefit of using the NoSQL is that unlike relational database, it can manage all the unstructured data such as documents, multimedia, e-mail etc. It has a very simple data model with no complexities, easy query language. The con or the most unreliable feature of the NoSQL database is its lack of security and regularity. It is not well equipped with management of data with unity and integrity constants like foreign key handling.

6.1 Cassandra

Cassandra is highly symmetric in nature. It manages large amount of structured data and provides highly reliable services with least amount of failure. It does not have a fully-fledged relational database model but it has a model that is simpler and has control over data layout and format. It was Cassandra, which was used to implement the Inbox search feature of Facebook. It has a key value store. Key here maps to values that are widely distributed into column families. Here, columns can only be appended to keys, which are already specified, and which implies each and every key has a different number of columns in a particular family. It also has super columns, a column whose values are columns, which is ideally, a sorted array of columns linked associatively. The values from column key for each family are stored together. This defines Cassandra as big hybrid data manager between a columns oriented DBMS and a row-oriented store.

6.1.1 Security in Cassandra

- Cassandra data files are unencrypted in nature. It has no mechanism to convert unencrypted data to encrypted data.
- The communications, which takes place between the database and the users, is unencrypted in nature. Thus, the attacker who is able to see the database traffic would be able to see the unencrypted data as if it were the client.
- CQL namely Cassandra language used in Cassandra is a parsed language and thus is highly vulnerable in nature to injections.
- It provides Simple Authenticator function, which permits making users names and passwords. This particular file is formatted as a set of password properties that are in plain text only.
- It is not in favor of inline auditing.

6.2 MongoDB

It is mostly written in C++. It is a document kind database that manages JSON-like documents. Here data can be nested in complex hierarchies and then also are indexable. Documents are stored in the form of collections in collections and are in turn stored in database. It provides high amount of unity, regularity and the ability to change size.

MongoDB offers a set of collection in collection. A collection can be considered as a table but the table does not have a schema here, set of fields could be defined as a document. In this every document consists of an id. MongoDB is a cluster of nodes where not every node is the same, that is it lacks symmetry.

6.2.1 Security in MongoDB

- In MongoDB the data files are unencrypted in nature and it does not provide an implementation to encrypt the files automatically.
- It uses a binary wire level protocol which uses TCP port 27017 by default. It is at high verge of injection attacks because it uses mostly java script language. Thus, any attacker can easily acquire the passwords of the data files of user in a particular database.
- No auditing is allowed here.
- In data communication no encryption is there.

The main drawback in both of the NoSQL database was the lack of encryption, weak authentication between the client and the server, denial-of-service and the vulnerability to injection in the database. To decrement the injection in the database it should be verified that the application does reasonable input validation.

The data at rest that is unencrypted can be protected with the OS level mechanism. [4]

7. IMPLEMENTATION OF MONGODB IN TEXTBOOK MANAGEMENT SYSTEM, AN EXAMPLE

With the development of technology, the relational database is more and more encouraged in information management system. But it does not work efficiently when a query of a large amount of data is done with multi-table join queries. To combat this noSQL is introduced. It is a non-relational database that has no join operation in it.

Relational database uses a two dimensional table to represent

data and has an austere regularity, scalability, unity of the database transaction with the inbuilt ability to handle multi table related query which becomes less efficient when a large amount of data is there. To implement the relational database MySQL is used.

The non-relational database namely NoSQL has no join operation but its query is efficient. NoSQL unlike relational database manages all the unstructured data such as documents, multimedia, e-mail etc. It has a very simple data model with no complexities, easy query language. MongoDB does the NoSQL here. MongoDB is a document-oriented database that has a database in which there are collections and in collections there are documents. It has a de-normalized model that basically contributes in incrementing the query speed.

7.1 Implementation Schema

7.1.1 Traditional way

The system that uses a traditional data model using MySQL basically comprises of students, faculty, notebooks, allotted textbook, usage of notebook, storing notebook, entries notebook, delivery notebook, and some other entities. Students related to a particular section and department are given by foreign key. The department code would have a one to many relations with the faculty information. The faculty information has many to many relations with the notebook. If a query has to be done for a certain teacher's department and using his/her notebooks of information, there is a need to relate the department code, faculty information,

Faculty use-book and notebook, these four columns. A massive amount of time is spent in solving this multi table query, which is not even solved efficiently.

7.1.2 MongoDB

Primary information like student and faculty information will be made in MongoDB. Allotted textbook, usage of notebook, storing notebook, entries notebook, delivery notebook can be placed into the data of faculty with embedded documents in the collection. When the number of embedded documents led to an incrementing size of one single document, there is a necessity to create a separate notebook that is contained in a collection.

7.2 System Implementation

Using JAVA/C++ to implement the MongoDB as it can work with any platform.

7.2.1 Insert

MongoDB database utilizes question situated thinking to actualize framework, to begin with, agreeing "textbook" to make a DBCollection protest collection, and after that make a BasicDBObject protest doc, add course readings data to the protest doc, lastly add the question doc to the question collection, finally embed information into MongoDB database.

7.2.2 Query

Query information in MongoDB database takes after object oriented thinking as well. To make DBCollection question collection for deciding the extent of the inquiry by getting Collection "textbook" in the database. Later make BasicDBObject protest condition, and compose the question condition into condition, at long last put the question condition into the technique findOne of the question. Collection, the query comes about come back to DBObject question return.

7.3 Performance

In performance testing, if 10000 notebooks information data is inserted into database, the cost time of MongoDB and MySQL were noted. In the performance of inserting data, MongoDB spends less time than MySQL, which implies that query efficiency is improved. [6]

8. CONCLUSIONS

NoSQL database are faster and better than SQL database in many ways, ranging from speed to flexibility. Also, an existing database in SQL can be transferred into a MongoDB (NoSQL) with quite an ease and these are the reasons many companies are shifting their projects so as to use MongoDB instead of traditional SQL database. The Erwin HAWK tool can be used to choose a suitable NoSQL database and replace the traditional relational database. NoSQL is also called the future of data economy.

MongoDB is not only faster in most of the queries a Oracle database faces, it is also more flexible and can store large data with ease. It's flexibility and speed is the main up-votes to switch to MongoDB. Also, as mentioned earlier one can migrate over to MongoDB from SQL based Oracle, with little difficulties.

Taking everything into account, on the off chance that if one needs to utilize a quick, adaptable database, he or she can depend on MongoDB. In the event that the speed of the database isn't a principle concern, and on the off chance that relations are required between the tables and the accumulations, one should depend on the great, Oracle Database.

There are certain drawbacks of NoSQL database too, major concern being the addressing the security issue which, in case of NoSQL is still in its nascent state and needs time to develop. For the same reason and some other it is very uncommon to use MongoDB in cloud and other such applications. One can easily move to NoSQL database as the next best thing for the future as it is an open-source and free, also its more flexible. But some majors concerns like security still need a fixation, without which NoSQL will always be a step behind SQL, as security is one of the major reasons to implement any database using SQL and now a days security is one of the biggest issues keeping in mind all the scams and ability hackers have acquired. To decrement the injection in the database, it should be verified that the application does

reasonable input validation. NoSQL is not much efficient in cloud computing and research has to be strengthened in these areas.

9. LIST OF FIGURES

Serial Number	Name
Figure 1	TIME TAKEN BY AN SQL (ORACLE) AND NOSQL (MONGODB) TO UPDATE RECORDS IN A DATABASE, Y AXIS IS TIME IN SECONDS AND X AXIS DENOTES NUMBER OF RECORDS UPDATED
Figure 2	TIME TAKEN BY AN SQL (ORACLE) AND NOSQL (MONGODB) TO INSERT RECORDS IN A DATABASE, Y AXIS IS TIME IN SECONDS AND X AXIS DENOTES NUMBER OF RECORDS INSERTED

10. ACKNOWLEDGMENTS

Special thanks to faculty, Prof. Naveen Kumar, SCOPE and the university for helping through out.

11. REFERENCES

- [1] MongoDB Documentation <https://docs.mongodb.com/>
- [2] MongoDB <https://www.mongodb.com/>
- [3] Tianyu Jia, Xiaomeng Zhao, Zheng Wang, Dahan Gong and Guiguang Ding - Model Transformation and Data Migration from Relational Database to MongoDB IEEE 2016.
- [4] Lior Okman, Nurit Gal-Oz, Yaron Gonen, Ehud Gudes, Jenny Abramov - Security Issues in NoSQL Databases IEEE2011
- [5] Jing Han, Haihong E, Guan Le, Jian Du - Survey on NoSQL Database IEEE2011
- [6] Zhu Wei-ping, Li Ming-xin- Using MongoDB to Implement Textbook Management System instead of MySQL IEEE2011
- [7] Alexandru Boicea, Florin Radulescu, Laura Ioana Agapin - MongoDB vs Oracle - database comparison. IEEE2012