Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

# Comparative analysis of NoSQL (MongoDB) with MySQL Database

Lokesh Kumar[1] , Dr. Shalini Rajawat[2],Krati Joshi[3]

[1]*Computer Science & Engineering, Vivekananda Global University, Jaipur, Rajasthan, India,*
[2]*Computer Science & Engineering, Vivekananda Global University, Jaipur, Rajasthan, India,*
[3]*Computer Science & Engineering, Vivekananda Global University, Jaipur, Rajasthan, India,*

**Abstract-** New requirements are arising in database area where we have higher volumes of data with high operation rates, good development and big database. In recent years, a increase number of companies have adopted different-different types of non-relational database (MongoDB, Cassendra ,Hypertable, Hbase/Hadoop ,CouchDB etc) , commonly referred to as NoSQL database. This reflects the increasing interactivity of applications which are becoming more networked and search engine, handling more requests to the database where high-performance NoSQL database such as MongoDB, Cassendra becomes favorable. This paper attempts to use NoSQL database to replace the relational database. It mainly focuses on one of the new technology of NoSQL database i.e. MongoDB, and makes a comparison study with MySQL and thus justifies why MongoDB is liked over MySQL. Lastly, a method is suggested to integrate with different-2 technology of these two types of database by adding a middleware (Metadata) between application layer and database layer. But in this paper we are propose with open source language like PHP.

**Index Terms**- ACID, BASE, MySQL, MongoDB, Metadata, NoSQL, RDBMs, PHP, Xampp

## I. INTRODUCTION

Databases are defined as a collections of data . Although when using the term database we refer to the complete database system, the term actually refers only to the collection and data. The system which handles Big data, transactions, problems or any other aspect of the database is the Database Management System (DBMS). What follows is a description of the two database types which will be compared in this dissertation. Early designs and implementations were based on the use of linked lists to create relations between data and to find specific data. These models were not standardized and required extensive training in order to make efficient use of them. These models and other important types are explained briefly as well.

Databases were created in order to satisfy this need of storing and finding data in an good manner. after their inception in the 1960's different types have invented, each using its own representation of data and different-2 technology for handling transactions. They began with navigational databases which were based on linked-lists, moved on to relational databases with joins, afterwards object-oriented and without joins in the late 2000s NoSQL(MongoDB,Cassendra, Hypertable, Hbase/Hadoop , CouchDB etc) emerged and has become a popular trend [1].

 

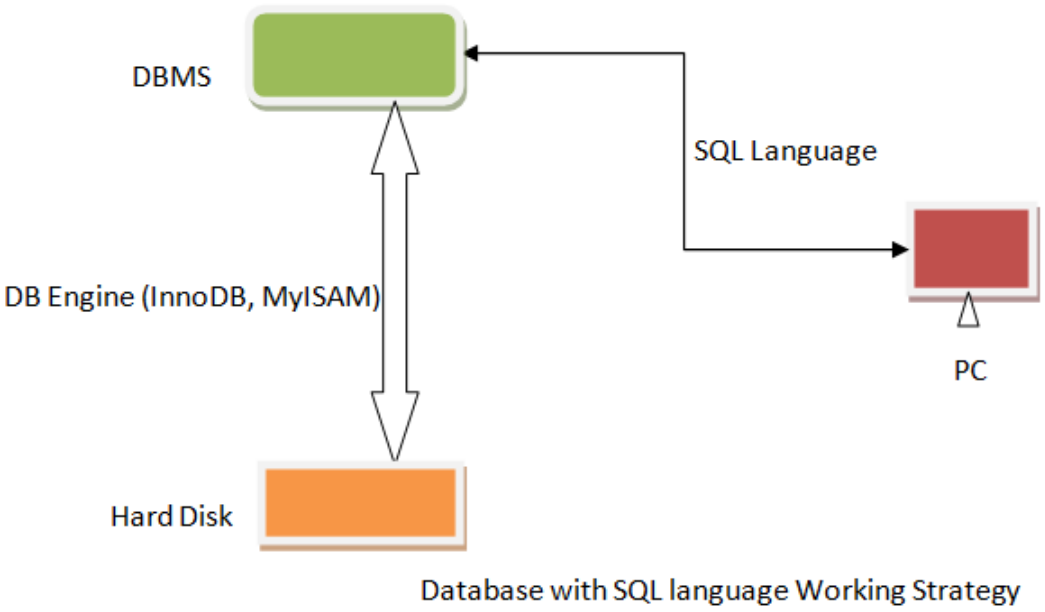


*Figure: 1 Database with SQL language Working Strategy*

## II. OVERVIEW OF MONGODB DATABASE

MongoDB is a document-store database designed for best scalability, high availability and good performance. It allows data persistence in a nested state and have the ability to query the nested data in an undefined fashion with embedded queries. In addition, it does not inflict schema, allowing it to adapt quickly as applications. Moreover, a MongoDB document can contain field types that other documents of the same collection do not have. Anyhow of this flexibility, MongoDB still ensures expected functionalities such as full query language and consistency.

MongoDB is in the forefront of NoSQL databases, providing agility and scalability to businesses. More than thousand companies and new start-up companies have acquire and are using MongoDB to develop new applications, refine client experience, fast track marketing time and minimize costs. It's use of mostly bigger web applications like facebook , Amazon, Google etc.

### 2.1 Architecture

A. Document Data Representation

MongoDB stores data as document in a binary-encoded form called BSON or simply Binary JSON. Like in relational databases, MongoDB organizes documents that tend to have similar structure as collections. A Collection in MongoDB corresponds to a table in relational databases, a document is a row, and a field is a column.

 

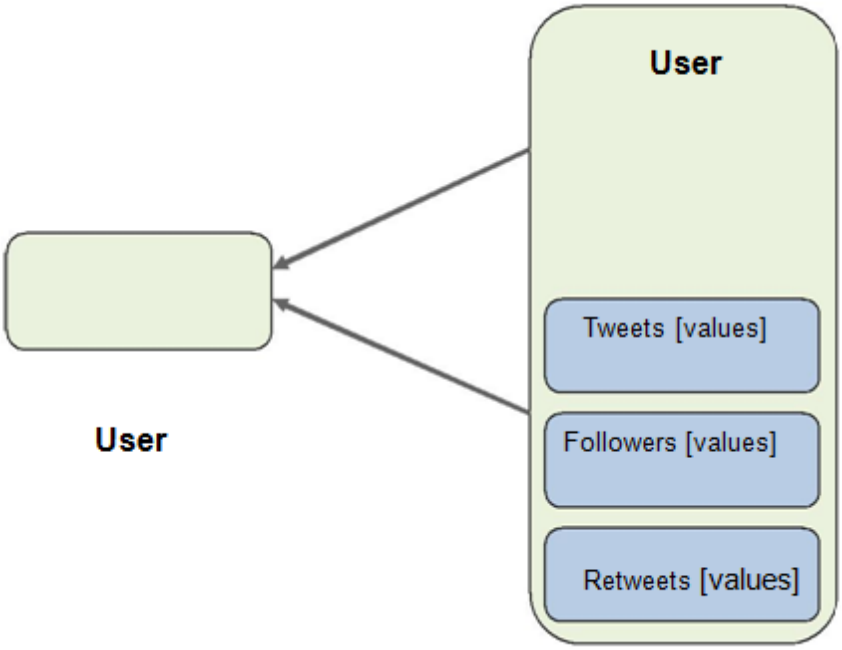


*Figure:2.1 Data model for a Twitter related application*

Let us consider the data model for a twitter clone application is a example for understanding . A relational database will model the data as multiple tables of User, Followers, Tweets and Retweets. However, in MongoDB Database content the data could be represented as a single collection of Users. Each User's document may be contain followers, Tweets, Retweets, represented as an embedded array because in MongoDB Database joins like inner join and outer joins are not working it's use for joins type working done with embedded queries. In order words, while information of a each different-2 record in relational databases is usually spread across multiple tables with multiple row and column, MongoDB may have all data of a particular record in a single document. so it's a worth doing database.

*B.* *Dynamic* Schema

The structure of a MongoDB document can vary from document to document in the form of JSON, unlike in a relational database where the structure for a row must be defined. For instance, all documents describing Twitter Users might contain user ID, tweets and followers. However, some documents do not necessarily have to require user ID for one or more third-party applications. Hence, fields can be added to a document if need be, without disrupting other documents or updating the central system catalog or having system downtime.

## 2.2 Query Model

A. Unique Drivers

MongoDB supports mostly well-known programming languages and frameworks by providing native drivers to enhance natural development. Each MongoDB driver is idiomatic for the respective language. Listed below are supported popular drivers:

- Ruby
- PHP

- JAVA
- .NET
- Java Script
- Node.JS
- Python
- Perl

B. Query Type

MongoDB queries come in different-different form. A query issued to extract information from a collection may return a document in the form of JSON or a particular set of fields within the document. Listed below are MongoDB query types.

- **Map Reduce Queries** perform complex data processing expressed in JavaScript and performed all data in the database collections.
- **Text Search Queries** return results in the order of relevance using text arguments containing Boolean operators like NOT,AND,OR.
- **Aggregation Framework Queries** return groups of values returned by the query, analogous to GROUP BY in SQL statements.
- **Key-value Queries** return results using a specific field in the document, usually the primary key.
- **Range Queries** return results using values defined as inequalities such as equal to, greater than, less than, less than , greater than or equal to or equal to.
- **Geospatial Queries** return results using a proximity area, inclusion and intersection as specified by a circle and point and plygon.

C. Indexing

Indexes in MongoDB are a significant mechanism for optimize query execution system performance. In spite of the enhanced performance of operations in order of importance, indexing poses the consequence of slower write operation, disk and memory usage. Below are listed the types of indexes MongoDB supports on any field in a document of collections.

- Unique Indexes
- Compound Indexes
- Array Indexes
- Time To Live (TTL) Indexes
- Geospatial Indexes
- Sparse Indexes
- Text Search Indexes

### 2.3 Auto sharing

Sharing is a technique MongoDB uses for providing horizontal scale-out for NoSQL databases. It involves the spreading of all data multiple physical servers known as shards, thereby solving the hardware fixed area of a single server without causing complexity in the system. MongoDB has a mechanism of balancing the data growth across the cluster and also when the cluster either multiplies or decreases. The three types of sharing supported by MongoDB are Range-based, Hash-based and Tag-aware sharing's.

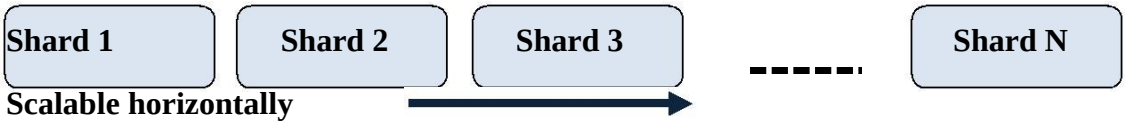


*Figure:2.2 Providing horizontal scalability through sharing.*

## III. Comparison on MySQL with MongoDB Database

As per the detailed review of several research papers and SQL and NoSQL Databases Books, a comparative study is made between MySQL and MongoDB based on their concept and commands used for different operations.

### A. Based on Terms/Concept[5]

Table I: TERMS/CONCEPT

| SQL terms/concept | MongoDB terms/concept |
|---|---|
| Table | Collections |
| Row/Records, Columns/Types | Key-Value Pairs,Documents |
| Index | index |
| table joins | Embedded Documents and Linking |
| fixed schema | schema less |
| primary key (explicitly) | primary key (implicitly) |

### B. Based on behalf of Queries

Table II: Queries Statement [6]

| SQL Queries | MongoDB Queries |
|---|---|
| Select Query:<br>Select * From Students where id=”200A20” | Select Query:<br>db.Students.find( { },{t_id:1}) |
| Insertion Query:<br>Insert into students (s_id,name, course, branch ,email)<br>values(“200A20”,“Lokesh”,”M.Tech”,”CSE”,” lkbansal1993@gmail.com”) | Insertion Query:<br>db.students.insert({s_id:"200A20",name:”Lokesh”,course:“M.Tech”,branch:”CSE”,email:lkbansal1993@gmail.com}) |
| Create Query:<br>Create table students (s_id char,name varchar(50),course varchar(100),branch varchar(100),email varchar(50)); | Create Query:<br>db.students.insert({s_id:"200A20",name:”Lokesh”,course:“M.Tech”,branch:”CSE”,email:lkbansal1993@gmail.com}) |
| Drop Query:<br>Drop table students; | Drop Query:<br>db.students.drop(); |
| Delete Query:<br><br>Delete From students where s_id=’200A20’ | Delete Query:<br>Db.students.delete({s_id:”200A20”}) |

 

### C. Based on Query Execution speed/performance

Use GUI Tool for Find performance: HeidiSQL 8.3 for MySQL ,Robomongo 0.8.3 for MongoDB .
In [2], I have performed testing comparisons with NoSQL( MongoDB) with MySQL database with the help PHP. They have performed testing by using the some raw database for comparisons query speed with performance. The given graph shows the result of testing. In performance testing, the authors have inserted 100 to 50,000 same type row with the help loop in php then easily inserted . The cost time of MongoDB and MySQL were recorded as shown in graph. Two important factors for which MongoDB was preferred over MySQL:

- data fetch Speed

From the graph, we notice that MongoDB spends less time than MySQL, for a large amount of information as shown in figure 2. It leaves MongoDB 30×-50× faster than MySQL as sown in[3].

Number of Parallel Clients 5 Time in seconds

| | Total Rows | Rows / client | SQL Time | Mongo Time | Sql Ops/sec | Mongo Ops/sec |
|---|---|---|---|---|---|---|
| Basic Query | 50 | 10 | 0.1 | 0.08 | 500 | 625 |
| with index | 500 | 100 | 0.38 | 0.1 | 1,316 | 5,000 |
| | 5,000 | 1,000 | 2.8 | 1.2 | 1,786 | 4,167 |
| | 25,000 | 5,000 | 14 | 4 | 1,786 | 6,250 |
| | 50,000 | 10,000 | 28 | 10.4 | 1,786 | 4,808 |

*Figure : 3.1 QUERY SPEED COMPARISONS*

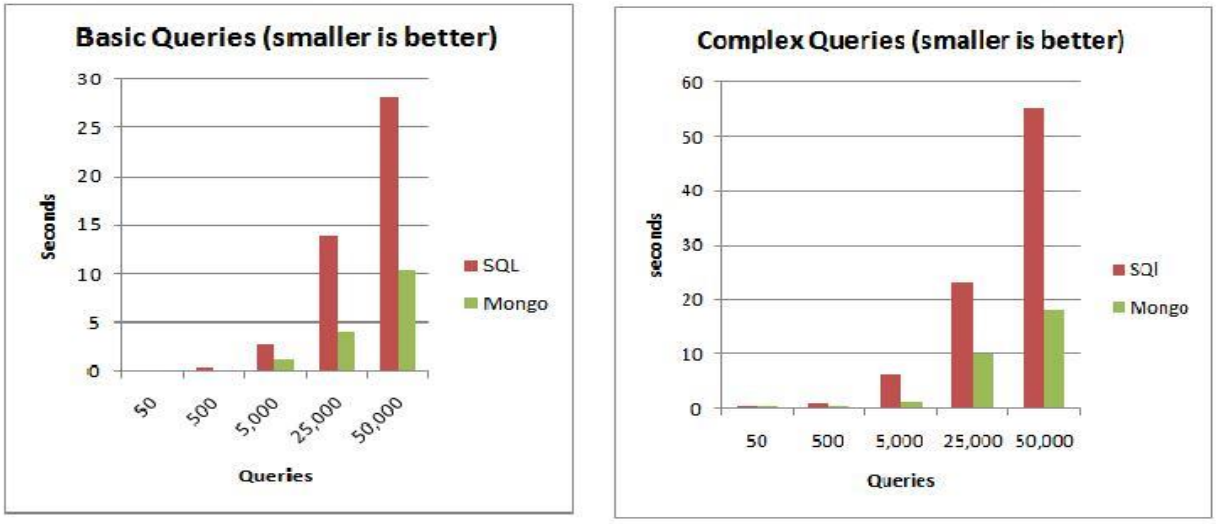


*Figure :3.2 BASIC AND COMPLEX QUERY TIME FOR MYSQL AND MONGODB*

## IV. Methodology of Work

In this system, [7]a method is used to integrate these two types of database by adding an interface between the Application layer like use PHP Language and Database layer. The middleware between the two consist of Metadata which consist of different-2 types of packages. This system useded to

implement a package which acts as an interface between php application layer and NoSQL database (MongoDB).The system makes use of MongoDB NoSQL Database and the reason for big database problem and timing problem. The system is primarily designed for embedded php based web application which requires database access. The database command fired from php based web application is given as input to the interface acting as a middleware(Metadata).The interface parses the input and reformats the code in the format that is been required by the back-end database (MongoDB) NoSQL Database.The reformatted code consists of functions that directs back-end database to implement and maintain the database.In form of result and shown it, the back-end databases will response to the middleware with the results. As a result, the middleware responses to the PHP web application in terms of successor an exception in case offailure. Hence to act as middleware the system has to perform conversion from one form to another form of result generated. Thus, the system has to store some information about the format conversion rules and the data structure format like JSON. In turn, the system has to store the metadata for supporting the working showing with the help of diagram.

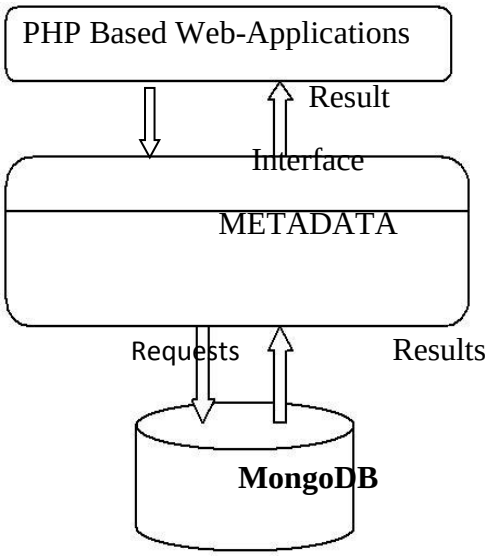


*Figure : 4 DATABASE INTEGRATION ARCHITECTURE*

In our methodology of work, we make use of standard SQL query language to use commands to the database like insert, delete,drop,select,create. Due to the need, of huge amount of data storage we make use of one of the boosting invented technology of NoSQL database i.e. MongoDB,Cassandra. MongoDB stores data in JSON structure.so get result in different-2 format it automatically . The reason for using SQL query language is that it is a mostly useable and standard database as well many organizations in the it fields used for developing system throughout the world are expert in programming using SQL language.

## V. CONCLUSION

Relational Database Management system won't go away for big databases application because have a drawback to our applications like time consuming and execution speed and scaling of queries, they are define compulsory. But the storage requirement for the new generation of applications are huge different from heritage applications.We can choose NoSQL(MongoDB) instead of MySQL because of two factors, ease of use and timing performance. We conclude that if your web application is data intensive and stores lots of big-data, queries lots of data, and generally lives database, then you better do that efficiently or have resources (i.e. money) to burn .it's a worth doing for all organizations for developing applications Lastly, the report concludes by method define a database integration method by using a middleware between the two layers. In this method, application does not have to consider about the complexity of underlying database layer with mongoDB Databases .there data distribution and storage. They have to use the basic SQL query language to get result from the database and all the format conversion rules will be done by the Metadata because data fetch from database in the form of JSON format. The system was proposed because MongoDB has newly come into existence, whereas the standard SQL language has been over years and, therefore if we merge the two we can use the features of both the database. Although, NoSQL(MongoDB) has the advantage of horizontal expansion, but for complex SQL requests, it cannot support them very well. For the Query based on KEY/VALUE and massive data storage requirements, NOSQL is a very worth doing choice for me and all other developers and organizations who's developed big applications.

## REFERENCES

1. Berg K., Seymour T., and Coel R. History of Databases. International Journal of Management and Information Services, 17, 2013.
2. Zhu Wei-ping, Li Ming-xin, Chen Huan, "Using MongoDB to Implement Textbook Management System instead of MySQL",IEEE,978-1-61284-486,2011
3. Sanobar Khan, Prof. Vanita Mane" SQL Support over MongoDB using Metadata",IJSRP, Volume 3, Issue 10, October 2013
4. MongoDB High+Performance+Benchmark+White+Paper_final paper March 2015
5. MongoDB Books http://mongodb.org/display/doc/
6. mongodb with php Http://www.tutorialspoint.com/mongodb/mongodb_tutorial.pdf
7. Lior Okman, Nurit Gal-Oz, Jenny Abramov,"Security Issues in NoSQL databases",IEEE, 978-0-7695-4600-1, 2011

 