



MONGODB VS MYSQL: A Comparative Study of MongoDB and MySQL Based on Their Performance

Shrikant Patel, Sanjay Kumar, Sandhya Katiyar,
Raju Shanmugam and Rahul Chaudhary

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

October 27, 2020

MONGODB VS MYSQL: A Comparative Study of MongoDB and MySQL Based on Their Performance

Shrikant Patel^{1*}, Sanjay Kumar², Sandhya Katiyar³, Raju Shanmugam⁴, Rahul Chaudhary⁵

^{1,2,4} School of Computing Science and Engineering, Galgotias University, Greater Noida, India

³Galgotias College of Engineering & Technology, Greater Noida, India

⁵Bhai Parmanand Institute of Business Studies, Delhi, India

patelshrikant@rediffmail.com

Abstract—A database refers to data structure used for systematical and organized collection of data. Nowadays database system follows mostly two different algorithms i.e. SQL and NoSQL. In SQL databases schema is predefined, fixed and vertically scalable whereas in NoSQL databases, schema is dynamic and horizontally scalable. Also, SQL databases are table-based databases such as MySQL, Oracle, MS SQL whereas NoSQL databases are document based, key-value based, graph databases such as MongoDB, CouchDB so on. Here, two comparison systems one is desktop application using python language and other is web application using PHP is used to analyze the performance difference between MongoDB and MySQL based on their Insert, Read, Update and Delete operations. In Observation of results its seen that for Insertion, Updating and deletion operations MongoDB performs much better than MySQL and for selection (data fetching) operation, although there is not major difference between them still average time of MongoDB select operation is less than average time of MySQL select operation. So, we have concluded that MongoDB performance is better than MySQL performance.

Keywords—SQL, NoSQL, MySQL, MongoDB

1. Introduction

A database is an organized collection of structured information or data, typically stored electronically in a computer system. A database is usually controlled by database management system. Now-a-days DBMS available on mainly based on two algorithms one is SQL like MySQL and other is NoSQL like MongoDB.

MySQL is a popular open-source relational database management system (RDBMS) that is distributed, developed, and supported by Oracle Corporation. The relational systems like, MySQL stores data in tabular form and uses structured query language (SQL) for accessing of data. In MySQL, the programmer should pre-define the schema based on requirements and set up rules to control the relationships between fields in the record [2]. The related information may be stored in different

tables, but they are associated by the use of joins. Thus, data duplication can be minimized.

MongoDB is an open-source database developed by MongoDB Inc. MongoDB stores data in JSON-like documents that can vary in structure. Related information can be stored together for fast query access through the MongoDB query language. MongoDB uses dynamic schemas, which helps to create records without first defining the structure, such as the attributes or the data types. It is possible to change the structure of records by simply adding new attributes or deleting existing fields. This model helps to represent hierarchical relationships, to store arrays, and other more complex structures very easily. Documents in a record need not have an identical set of fields [3].

In this paper we have compared both MySQL and MongoDB for their performance on basis of insert, select(fetch), update and delete operations of databases. Our comparison systems, Desktop comparison system based on python and Web comparison system based on PHP. Both comparison systems perform these four identical operations Insert into database, select(retrieve) from database, update in database and delete from database on same database in both database systems, MongoDB server and MySQL server.

2. PROBLEM DEFINITION

A database system is very crucial and important in creation as well as management of data. The task of data creation and data management is required by every server-based applications whether desktop application or web application. Now-a-days there are many databases systems available running on different principals. Two major principals of them: one is SQL based database systems like MySQL and other is NoSQL based database systems like MongoDB. MongoDB is document storage database (key-value based) and MySQL is table and foreign-key related database. MySQL and MongoDB both have some fundamental differences because of these fundamental differences there are some applications where table-based database system like MySQL always performs better and there are some applications where document-based database systems always performs better. But there are some applications also where any of both type of database system can be used. The main concern here is which one to choose for such type of applications.

In this paper we have compared both, MongoDB and MySQL for their basic operations: insert, select, update and delete. So that we can give an idea that which one database system can be opt for such type of applications where both type of database system seems suitable for application's need.

3. METHODOLOGY

Here we have two database, one database named `mongodb_vs_mysql` in MySQL database system and other is in MongoDB with the same name `mongodb_vs_mysql`. In MySQL, `mongodb_vs_mysql` database consists of one table named `user info`

having more than 100000 records. And in MongoDB, mongodb_vs_mysql database consists of one collection which having more than 100000 documents.

```
mysql> use mongodb_vs_mysql
Database changed
mysql> SELECT count(*) FROM user_info;
+-----+
| count(*) |
+-----+
|    111021 |
+-----+
1 row in set (0.23 sec)
```

Figure 1: Records Count in MySQL

```
> use mongodb_vs_mysql
switched to db mongodb_vs_mysql
> db.user_info.find().count()
111021
>
```

Figure 2: Documents count in MongoDB

In MySQL user_info table has 9 columns to store user information similarly In MongoDB user_info collection has 9 key-value pairs in each document.

```
mysql> use mongodb_vs_mysql
Database changed
mysql> SELECT * FROM user_info LIMIT 10;
```

login_id	login_password	firstname	lastname	gender	mobile	email	dob	nationality
Abel	Schlabaugh	Krystina	Schlabaugh	Male	9254678130	kschlabaugh@schlabaugh.co.uk	1992-02-22	India
Abraham	Cratch	Abraham	Cratch	Female	9624478246	acratch@gmail.com	1996-11-18	India
Adelina	Zurcher	Jerry	Zurcher	Male	9888480009	jzurcher@zurcher.org	1992-02-22	India
Adell	Zagen	Serina	Zagen	Male	9948480409	szagen@aol.com	1992-02-22	other
Adelle	Syrop	Eulah	Syrop	Male	9876479929	eulah@syrop.co.uk	1993-03-23	India
Adolph	Feagins	Gilberto	Feagins	Female	9390609037	gfeagins@gmail.com	1994-04-24	other
Agustin	Chavous	Roslyn	Chavous	Male	9246508810	roslyn.chavous@chavous.org	1992-02-22	India
Ahmad	Knodel	Sherron	Knodel	Female	9822909718	sherron@knodel.co.uk	1995-05-25	India
Ahmed	Discipio	Xochitl	Discipio	Female	9792479368	xdiscipio@gmail.com	1997-08-30	India
Aja	Kippley	Deonna	Kippley	Female	9924480249	deonna_kippley@hotmail.com	2002-10-08	India

```
10 rows in set (0.05 sec)
```

Figure 3: Sample of data records in MySQL

```
> use mongodb_vs_mysql
switched to db mongodb_vs_mysql
> db.user_info.find().pretty()
{
  "_id" : "Ulysses",
  "login_password" : "Mcwalters",
  "firstname" : "Ulysses",
  "lastname" : "Mcwalters",
  "gender" : "Male",
  "mobile" : "6798543210",
  "email" : "ulysses@hotmail.com",
  "dob" : "1992-02-22",
  "nationality" : "India"
}
```

Figure 4: Sample of data documents in MongoDB

To compare performance of both: MongoDB and MySQL we have used two comparison systems one is desktop application based on Python:

Figure 5: MySQL Insert Operation

```
start_time1 = time.time()
mydb = mysql.connector.connect(host="localhost",user="rahul",passwd="mahi")
cursor=mydb.cursor()
cursor.execute("use mongodb_vs_mysql")
cursor.execute(f"insert into user_info values('{self.insert_search_id.get().strip()}',\
'{self.insert_search_password.get().strip()}', '{self.insert_firstname.get().strip()}',\
'{self.insert_lastname.get().strip()}', '{self.insert_gender.get().strip()}',\
'{self.insert_mobile.get().strip()}', '{self.insert_email.get().strip()}',\
'{self.insert_dob_dateentry.get_date()}', '{self.insert_nationality.get().strip()}')")
mydb.commit()
start_time2 = time.time()
```

Figure 6: MongoDB Insert Operation

```
start_time1 = time.time()
c = MongoClient('localhost',27017)
db=c.mongodb_vs_mysql
result=db.user_info.insert_one({'_id':self.insert_search_id.get().strip(),
                               'login_password':self.insert_search_password.get().strip(),
                               'firstname':self.insert_firstname.get().strip(),
                               'lastname':self.insert_lastname.get().strip(),
                               'gender':self.insert_gender.get(),
                               'mobile':self.insert_mobile.get().strip(), 'email':self.insert_email.get().strip(),
                               'dob':str(self.insert_dob_dateentry.get_date()),
                               'nationality':self.insert_nationality.get()})
start_time2 = time.time()
```

Figure7: MySQL Select Operation

```
start_time1 = time.time()
mydb = mysql.connector.connect(host="localhost",user="rahul",passwd="mahi")
cursor=mydb.cursor()
cursor.execute("use mongodb_vs_mysql")
cursor.execute(f"select * from user_info where login_id='{self.search_id.get().strip()}'\
collate utf8mb4_bin and login_password='{self.search_password.get().strip()}' collate utf8mb4_bin")
result=cursor.fetchone()
start_time2 = time.time()
```

Figure 8: MongoDB Select Operation

```
start_time1 = time.time()
c = MongoClient('localhost',27017)
db=c.mongodb_vs_mysql
result=db.user_info.find_one({'_id':self.search_id.get().strip(), 'login_password':self.search_password.get().strip()})
start_time2 = time.time()
```

Figure 9: MySQL Update Operation

```
start_time1 = time.time()
mydb = mysql.connector.connect(host="localhost",user="rahul",passwd="mahi",client_flags=mysql.connector.constants.ClientFlag.FOUND_ROWS)
cursor=mydb.cursor()
cursor.execute("use mongodb_vs_mysql")
cursor.execute(f"update user_info set {self.selected_field_for_update}='{self.selected_field_for_update_data}' \
where login_id='{self.update_search_id.get().strip()}' collate utf8mb4_bin")
mydb.commit()
start_time2 = time.time()
```

Figure 10: MongoDB Update Operation

```
start_time1 = time.time()
c = MongoClient('localhost',27017)
db=c.mongodb_vs_mysql
result=db.user_info.update_one({'_id':self.update_search_id.get().strip()},
                               {'$set':{self.selected_field_for_update:self.selected_field_for_update_data}})
start_time2 = time.time()
```

Figure 11: MySQL Delete Operation

```
start_time1 = time.time()
mydb = mysql.connector.connect(host="localhost",user="rahul",passwd="mahi")
cursor=mydb.cursor()
cursor.execute("use mongodb_vs_mysql")
cursor.execute(f"delete from user_info where login_id='{self.delete_search_id.get().strip()}' collate utf8mb4_bin")
mydb.commit()
start_time2 = time.time()
```

Figure 12: MongoDB Delete Operation

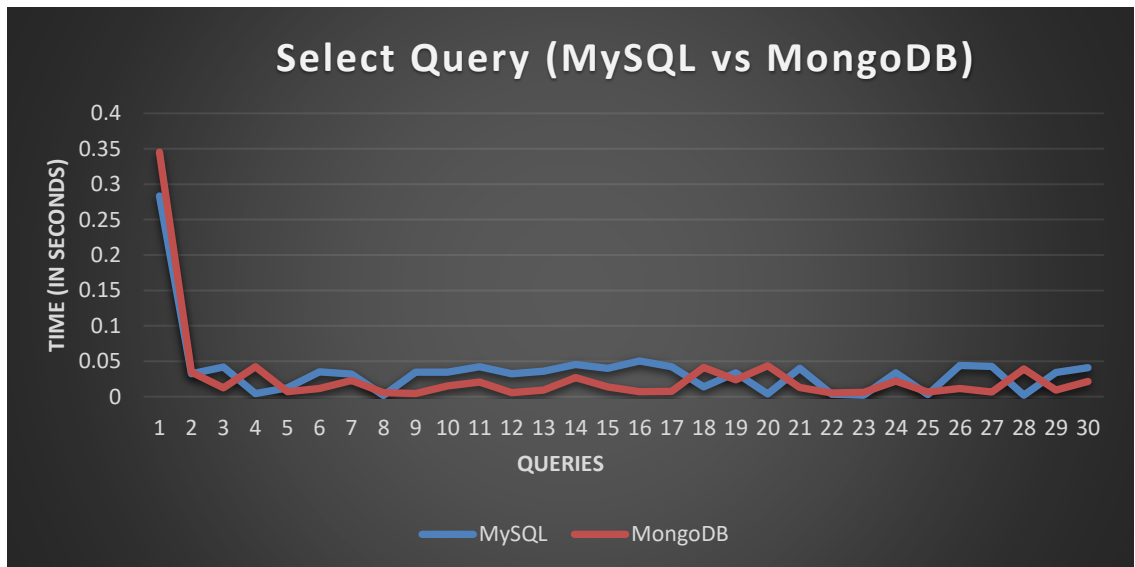
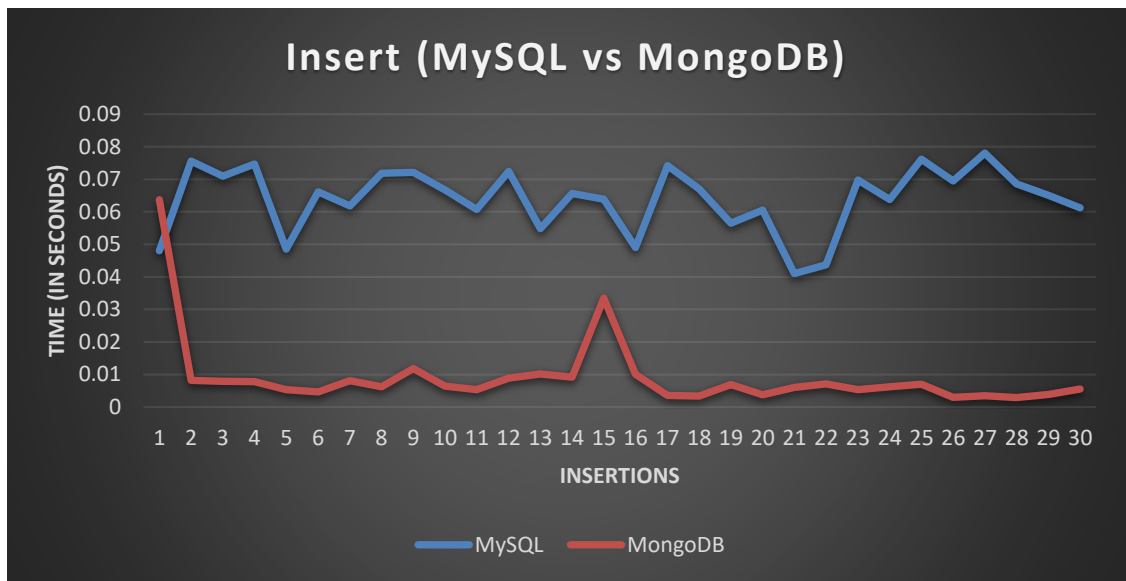
```
start_time1 = time.time()
c = MongoClient('localhost',27017)
db=c.mongodb_vs_mysql
result=db.user_info.delete_one({'_id':self.delete_search_id.get().strip()})
start_time2 = time.time()
```

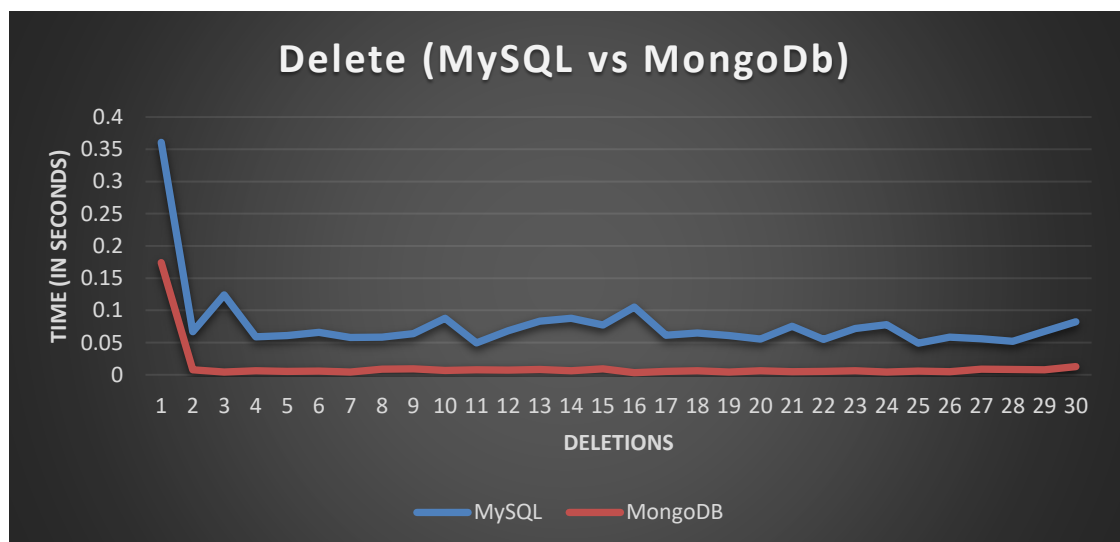
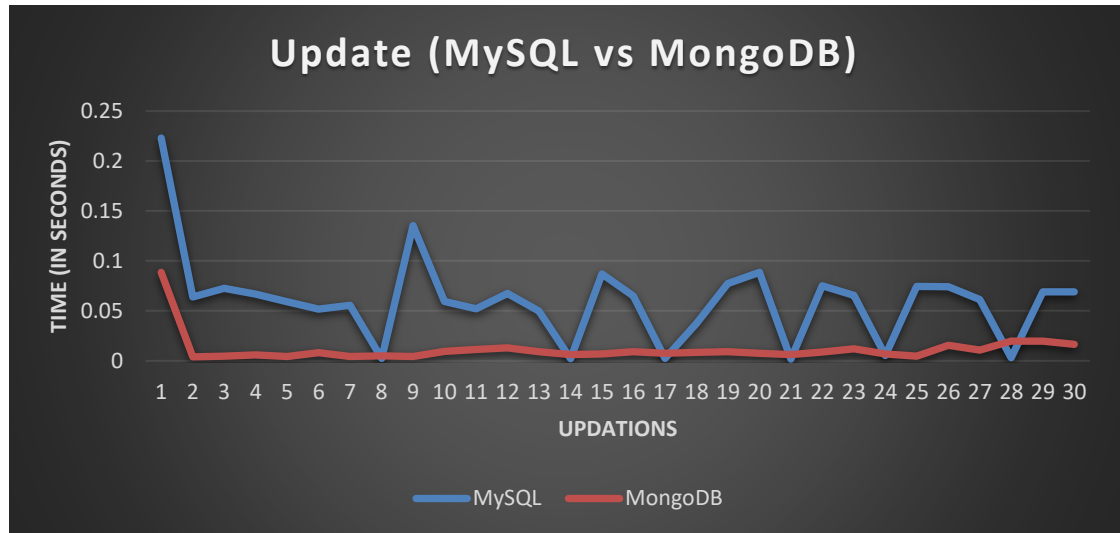
Similarly, four operation performed in web application comparison system also using PHP server-side scripting language. After performing all these operations on both database systems for different 30 -30 records and documents respectively on MySQL and MongoDB. We have collected operation execution time for both the databases separately for all operations: Insert, Select, Update and delete. After that observed result analyzed and difference between both are recorded for result.

4. PRACTICAL RESULTS

In Order to analyse performance difference between MySQL and MongoDB we have performed 30 Insert operation, 30 select operations, 30 update operations 30 delete operations for 30 different records in MySQL (having 100000+ records) and for same data having 30 documents in MongoDB (having 100000+ documents).

Observed Results are as following:





Average Time (in seconds)				
	Insertion	Selecting	Updating	Deletion
MySQL	0.0639	0.0366	0.0606	0.0787
MongoDB	0.0091	0.0280	0.0115	0.0121

Table 1: Average time for all operation in MySQL and MongoDB

7. CONCLUSION

In this paper, we have gone through the performance evaluation between MySQL and MongoDB for all CRUD operations. After Results we have observed that for insertion, updating and deletion operations **MongoDB performs much better than MySQL** and also observed that for selection (fetching data) operation there's not major difference in both still MongoDB average selection time is less in comparison to MySQL.

So, we reached to a conclusion that CRUD operations of MongoDB database system are faster than MySQL database system. It is clearly can be seen from the observed results.

As to summaries it can be said that SQL, Relational Database Management System (RDBMS) are slower in CRUD operations as compared to NoSQL, Non-Relational Database Management Systems. So, whenever we have applications where any of both type of database system can be used. Then we can opt MongoDB (NoSQL) as database requirement for better performance .

8. REFERENCES

1. Dipina Damodaran B, Shirin Salim and Surekha Mariam Vargese, "MONGODB VS MYSQL: A Comparative Study Of Performance In Super Market Management System", International Journal of Computational Science and Information Technology (IJCSITY) Vol.4, No.2, May 2016.
2. Kamil Kolonko, "Performance comparison of the most popular relational and non-relational database management systems", Master of Science in Software Engineering February 2018.
3. Vidushi Jain VIT University-Vellore, Aviral Upadhyay VIT University-Vellore, "MongoDB and NoSQL Databases", International Journal of Computer Applications (0975 – 8887) Volume 167 – No.10, June 2017.
4. Jing Han, Haihong E, Guan Le, Jian Du - Survey on NoSQL Database IEEE2011.
5. Lior Okman, Nurit Gal-Oz, Yaron Gonen, Ehud Gudes, Jenny Abramov - Security Issues in NoSQL Databases IEEE2011
6. Tianyu Jia, Xiaomeng Zhao, Zheng Wang, Dahan Gong and Guiguang Ding - Model Transformation and Data Migration from Relational Database to MongoDB IEEE 2016.
7. Mukesh Babulal Arethiya, Meenakshi Garg, "Comparative analysis of simple and aggregate query in MongoDB", International Journal of Advance Research, Ideas and Innovations in Technology.

8. Dileepa Jayathilake, Charith Sooriaarachchi, Thilok Gunawardena, Buddhika Kulasuriya and ThusithaDayaratne, “A Study Into the Capabilities of NoSQLDatabases in Handling a Highly HeterogeneousTree”,International Journal of Computational Science and Information Technology (IJCSITY) Vol.4, No.2, May 2016.
9. Elmsari and Navathe, “Fundamentals of Database Systems”, Pearson Education, 5th Ed., 2006.
10. Ullman J. D., “Principals of database systems”, Galgotia .
11. Malhar Lathkar,"Python Data Persistence: With SQL and NOSQL Databases".
12. Andreas Meier (Author), Michael Kaufmann (Contributor),"SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management".
13. Shannon Bradshaw (Author), Eoin Brazil (Author), Kristina Chodorow (Author),"MongoDB: The Definitive Guide: Powerful and Scalable Data 4