Performance Analysis of Queries in RDBMS vs NoSQL

Sirish Shetty B
Department of Information and Communication Technology
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal, India
sirish.shetty@manipal.edu

Akshay KC
Department of Information and Communication Technology
Manipal Institute of Technology
Manipal Academy of Higher Education
Manipal, India
akshay.kc@manipal.edu

***Abstract*—Due to the enormous amount of information getting stored in databases, an extra effort is needed to handle the information so that it is not lost in any way. To avoid the loss of information various methods were introduced and one such technique is introduction of NoSQL using MongoDB. In this paper, a comparison is made between structured Oracle and an unstructured MongoDB, with few identified operations in a controlled environment. The result obtained is then used to realize better performance of the structured or unstructured methods.**

***Index Terms*—Relational databases, Non-relational databases, Oracle, MySQL, NoSQL, MongoDB, RDBMS**

I. Introduction

Nowadays a lot of data is required to be stored and used for businesses be it E-commerce, Banking or Medical applications. The programmer has to decide the best way to store and retrieve the data required for such applications. In order to decide the best way, factors like scalability and performance are to be considered while choosing the database. Oracle, MySQL are relational databases and have been around for decades. MongoDB is relatively new and it is a non-relational database. The difference between relational and non-relational database is that the data is stored in a structured way in a relational database whereas in a non-relational database the data is stored in a semi-structured way.

The relational model was introduced in 1970s which offers structuring, using and keeping the data in a mathematic conformed way. It is an extended version of the initial designs of flat model or network model and any other models with an introduction to the relations. The advantage of relations is that it brings the group benefits such as tables-data which contains the information in a structured way, or relating the inputs by employing attribute values.

Since a major amount of research and development is happening in the database systems which follow through the relational model, they work very reliably and effectively. Because of the usage experience of these tools, database applications have become the first choice of acute usages. They cannot allow loss of any kind of information. Relational databases are scalable, or flexible and provide a lot of facilities with a minimum effort from the user side, even though it is slightly difficult to form and handle the data. That is why it might become less efficient while handling the large amount of data with a larger velocity.

This led to the invention of an efficient and easier way to handle the heavy data called NoSQL. NoSQL provides a higher scalability, a better and pliable data model and extremely superior performance. Even though MongoDB was first founded in 2007 by Dwight Merriman, Eliot Horowitz, and Kevin Ryan the team behind DoubleClick, it is one of the trending open-source document-based database application. It stores each record as a document, and a group of the document makes a collection. Modern application require Big Data, Fast feature development and flexible deployment, these features are provided by MongoDB [1].

For the purpose of this study, Oracle and MongoDB databases are considered and compared against each other for the performance of different types of queries. Oracle stores the data is structured way meaning the data fields to be stored are already predefined. Whereas in the case of MongoDB, not all the data stored need to be predefined. All records in MongoDB need not have the same number of data fields - there can be additional fields in some records or lesser in some records.

In this study, a comparison is performed for aspects of the queries like insert, update, delete and search in oracle against MongoDB. For the purpose of illustration, we take the example of storing a record of company employee details.

Following is the structure of the paper. Section II provides the literature survey performed prior to the analysis. Section III details the motivation followed by Section IV which describes the methodology used for comparing the various aspects of MongoDB and Oracle. Section V details the analysis performed on the results obtained followed by Section VI which concludes the paper.

II. Related Works

This section provides the related works.

Rajat et al [2] compared SQL and MongoDB and have detailed on the current database models such as non - relational and relational models. They have also performed an analysis between the two databases for various data sets. The analysis construes the time taken by the two databases for joins, insertions and retrievals. They claim to have showed

978-1-7281-0283-2/19/$31.00 ©2019 IEEE

how MySQL performs better for simple queries with smaller datasets while MongoDB works better for efficient complex queries with large data sets.

Dipina et al. [3] performed a performance evaluation of MySQL and MongoDB where MySQL is an example of relational database and MongoDB is an example of non relational databases. They have performed the analysis for insert and search operations upto a 25000 records. They claimed that relational databases suffer from horizontal scaling whereas NoSQL performs better than MySQL.

Mayur et al. [4] have tried to emphasize the document oriented databases MongoDB and stressed on the need of NoSQL usages. They have taken up a case study to compare the MongoDB and SQL databases. They claimed that when the data is large the load balancing becomes difficult in SQL databases whereas MongoDB has an inbuilt load balancer which takes care the load baclacing in case of large amount of data.

Győrödi et al. [5] have performed a comparative study of MySQL with MongoDB and claimed that MongoDB works better. They have compared the insertion, updation and querying operations in both MySQL and MongoDB.

III. Motivation

This study is based on the development of the dynamic platform, for various operations, which are the user preferences. Relational dabases are static in nature since using a relational database, the users must follow a structured method. Even though the subforum is different for each forum of users, the technique followed to implement must be same for each user.

To foreground the advantages and benefits of the non-relational database MongoDB, it is compared with relational database Oracle. Basic four operations were performed on both databases:

1) Insert
2) Delete
3) Select (Query)
4) Update

IV. Methodology

For the purpose of illustration following example of storing records of company employee details is taken:

1) In oracle two tables with following schema were created:

```
emp_details(Id, fname, lname, address,
                    mobile, pan, aadhar)
emp_fin_details (Id, accntno, salary,
                    bankname)
```

2) But in mongoDB, a single table is created with following schema:

```
emp_all_details(Id, fname, lname,
      address, mobile, pan, aadhar,
         accntno, salary, bankname)
```

Data insertion started with the initial 10 records and then the performance comparison was performed. Then the table records were increased to 100, 1000 and 10,000 respectively. To populate the tuples into the tables, a PL/SQL block with loop is written, and the timing before insertion and after insertion were recorded. Then the difference in these timings gives the actual time of the operation. Similarly, for MongoDB, documents were populated and timings were calculated.

Similarly to perform delete, select and update operation, a PL/SQL block is written for each of them for Oracle and for MongoDB and in the very same way the timings are calculated.

In order to achieve higher performance of the operations like insert, delete and select, relational model aspect in mongoDB is compromised as compared to oracle. All the study conducted in this paper is performed in a controlled environment with specific system specifications with Windows 10 64bit OS, Processor Intel Core i5, RAM 4GB.

V. Results and Analysis

Table I shows the time in milliseconds for Oracle and MongoDB for different number of records varying from 10 to 10000.

TABLE I
Execution time in milliseconds for Oracle and MongoDB

		No. of Records			
		10	**100**	**1000**	**10000**
Oracle	**Insert**	0	6	37	350
	Delete	6	13	51	798
	Select	0	0	7	21
	Update	0	0	1	5
MongoDB	**Insert**	0	20	29	69
	Delete	0	1	11	364
	Select	0	0	0	8
	Update	0	1	8	70

The main objective here is to compare and analyze which database performs better for the identified operations.

A. Hypothesis Testing

For each operation performed in relational and nonrelational databases hypothesis testing can be performed. Procedure adopted to perform the hypothesis testing is ANOVA for all the operations.

1) For Inserting: Table II provides the timings in milliseconds needed to insert the records into databases.

TABLE II
Execution time to insert in milliseconds for Oracle and MongoDB

	10	100	1000	10000
Oracle	0	6	37	350
MongoDB	0	20	29	69

Following hypothesis can be stated:

H_0: Performance of Oracle for inserting is not better than the performance of MongoDB.

H_1: Performance of Oracle for inserting is better than the performance of MongoDB.

The prerequisite values calculated with reference to inserting is given in Table III.

TABLE III
PREREQUISITE VALUES TO CALCULATE $F_{calculated}$

Source of Variation	SS	d.f	MS
Between Sample	9453.13	(2-1) = 1	9453.13 / 1 = 9453.13
Within Sample	87813.75	(8-2) = 6	87813.75 / 6 = 14635.63
Total	97266.88	(8-1) = 7	

Based on the prerequisite value F-Ratio is calculated as

$$F - Ratio = \frac{MSbetween}{MSwithin}$$
$$= \frac{9453.13}{14635.63} = 0.65$$

Hence, the $F_{calculated}$ is 0.65. $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$. Therefore the H_0 is accepted which means the performance of Oracle is not better than MongoDB for inserting.

2) For Deleting: Table IV provides the timings in milliseconds needed to delete the records into databases.

TABLE IV
EXECUTION TIME TO DELETE IN MILLISECONDS FOR ORACLE AND MONGODB

	10	100	1000	10000
Oracle	6	13	51	798
MongoDB	0	1	11	364

Following hypothesis can be stated:

H_0: Performance of Oracle for deleting is not better than the performance of MongoDB.

H_1: Performance of Oracle for deleting is better than the performance of MongoDB.

The prerequisite values calculated with reference to deleting is given in Table V.

TABLE V
PREREQUISITE VALUES TO CALCULATE $F_{calculated}$ FOR DELETE OPERATION

Source of Variation	SS	d.f	MS
Between Sample	30258	(2-1) = 1	30258 / 1 = 30258
Within Sample	548528	(8-2) = 6	548528 / 6 = 91421.33
Total	578786	(8-1) = 7	

Based on the prerequisite value F-Ratio is calculated as

$$F - Ratio = \frac{MSbetween}{MSwithin}$$
$$= \frac{30258}{91421.33} = 0.33$$

Hence, the $F_{calculated}$ is 0.33. $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$. Therefore the H_0 is accepted which means the performance of Oracle is not better than MongoDB for deleting.

3) For Selecting: Table VI provides the timings in milliseconds needed to retrieve the records into databases.

TABLE VI
EXECUTION TIME FOR SELECT OPERATION IN MILLISECONDS FOR ORACLE AND MONGODB

	10	100	1000	10000
Oracle	0	0	7	21
MongoDB	0	0	0	8

Following hypothesis can be stated:

H_0: Performance of Oracle for select operation is not better than the performance of MongoDB.

H_1: Performance of Oracle for select operation is better than the performance of MongoDB.

The prerequisite values calculated with reference to select operation is given in Table VII.

TABLE VII
PREREQUISITE VALUES TO CALCULATE $F_{calculated}$ FOR SELECT OPERATION

Source of Variation	SS	d.f	MS
Between Sample	50	(2-1) = 1	50 / 1 = 50
Within Sample	342	(8-2) = 6	342 / 6 = 57
Total	392	(8-1) = 7	

Based on the prerequisite value F-Ratio is calculated as

$$F - Ratio = \frac{MSbetween}{MSwithin}$$
$$= \frac{50}{57} = 0.88$$

Hence, the $F_{calculated}$ is 0.88. $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$. Therefore the H_0 is accepted which means the performance of Oracle is not better than MongoDB for select operation.

4) For Updating: Table VIII provides the timings in milliseconds needed to update the records into databases.

TABLE VIII
EXECUTION TIME TO UPDATE IN MILLISECONDS FOR ORACLE AND MONGODB

	10	100	1000	10000
Oracle	0	0	1	5
MongoDB	0	1	8	70

Following hypothesis can be stated:

H_0: Performance of MongoDB for updating is not better than the performance of Oracle.

H_1: Performance of MongoDB for updating is better than the performance of Oracle.

TABLE IX
PREREQUISITE VALUES TO CALCULATE $F_{calculated}$ FOR UPDATE OPERATION

Source of Variation	SS	d.f	MS
Between Sample	666.13	(2-1) = 1	666.13 / 1 = 666.13
Within Sample	3421.75	(8-2) = 6	3421.75 / 6 = 570.29
Total	4087.875	(8-1) = 7	

The prerequisite values calculated with reference to updating is given in Table IX.

Based on the prerequisite value F-Ratio is calculated as

$$F - Ratio = \frac{MSbetween}{MSwithin}$$

$$= \frac{666.13}{570.29} = 1.12$$

Hence, the $F_{calculated}$ is 1.12 . $F_{tabulated}$ is 5.99 i.e $F_{calculated} < F_{tabulated}$. Therefore the H_0 is accepted which means the performance of MongoDB is not better than Oracle for updating.

Fig. 1, Fig. 2, Fig. 3, and Fig. 4 depicts the graphs for inserting, deleting, select operation and updating based on the time in milliseconds respectively. As it can be observed from the graphs that for insert, delete and select operations the time taken by oracle system is comparatively high whereas time taken by oracle during update it is less.

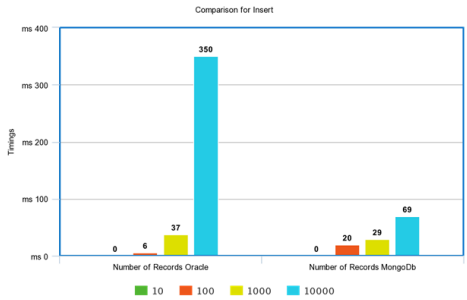

Fig. 1. Comparison for Insert

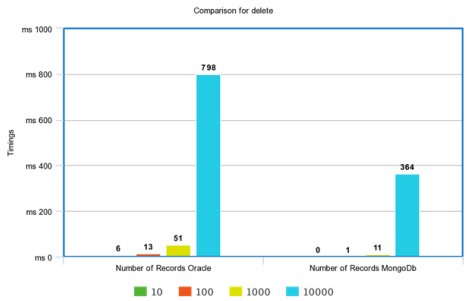

Fig. 2. Comparison for Delete

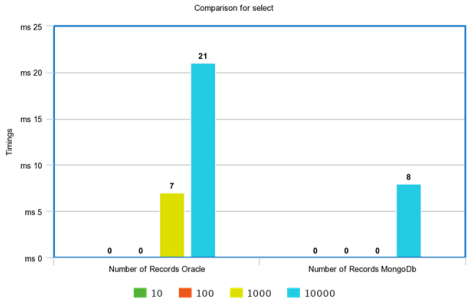

Fig. 3. Comparison for Select

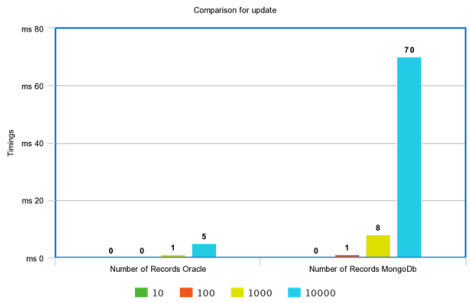

Fig. 4. Comparison for Update

VI. CONCLUSION

A comparison study has been made in this paper to realize that the non relational databases perform better than the relational databases. Considering the data set which was taken for the experiment, it can be verified from the hypothesis testing that, MongoDb performs better only for insert, delete and select operation, whereas for update operation Oracle still performs better in the given environment.

REFERENCES

[1] MongoDB. [Online]. Available: https://www.mongodb.com/what-is-mongodb

[2] R. Aghi, S. Mehta, R. Chauhan, S. Chaudhary, and N. Bohra, "A comprehensive comparison of SQL and MongoDB databases," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 5, pp. 325 – 330, February 2015.

[3] B. Dipina Damodaran, S. Salim, and S. M. Vargese, "Performance evaluation of mysql and mongodb databases," *International Journal on Cybernetics & Informatics (IJCI)*, vol. 5, pp. 387 – 394, April 2016.

[4] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, "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 *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Feb 2017, pp. 325–330.

[5] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, "A comparative study: Mongodb vs. mysql," in *2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, June 2015, pp. 1–6.