MVM: MySQL Versus MongoDB

Purva Grover and Rahul Johari

Abstract The Literature survey exhibits lack of quality research work in field of database(s) when it comes to task of performing comparison between real-world database entities. In this research paper, we have compared and contrast between the two open-source RDBMS (Relational database management system): MySQL and MongoDB. Comparison between two databases was done on basis of database operations, such as insertion, deletion, selection, projection et al. It is true that selection of the database in application depends entirely on database operations and we observed that in some database operations and applications MySQL performed better than MongoDB whereas in some applications MongoDB resulted in better performance. For the evaluation and analysis we obtained the real-time traces of diabetic dataset comprising of 100,000 records with 51 columns and put it to test for efficiency and performance to both the RDBMS and in the end the database operation execution time was recorded and analyzed.

Keywords MySQL · MongoDB · Queries · Performance of mySQL and mongoDB · Syntax of queries

Purva Grover (✉) · Rahul Johari
University School of Information and Communication Technology,
Guru Gobind Singh Indraprastha University, Drawka Sector 14, Delhi, India
e-mail: purva.usict.00416@ipu.ac.in

Rahul Johari
e-mail: rahul@ipu.ac.in
URL: http://ipu.ac.in/

M. Pant et al. (eds.), *Proceedings of Fifth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing 436, DOI 10.1007/978-981-10-0448-3_75

1 Introduction

1.1 Computer Applications

In most of the computer application, we require the database to archive the data for the future use and analysis. For these applications the success of the application are highly depended on the following activities: (1) Insertion of a record; (2) Selection of a particular record; (3) Querying database (4) Creation of Summary tables (5) Deletion of a particular record et al. The users of the applications are mostly concerned with these activities. In this research paper, we had taken two open-source database under GNU license viz. MySQL and MongoDB.

1.2 MySQL Databases

The word MySQL [1] can be broken down into two words My and SQL. The SQL stands for structured query language. MySQL is most widely used relational database management system. MySQL is available in two editions: (1) Open-source MySQL Community Server (2) The Proprietary Enterprise Server In this research work, we are using the Open-source MySQL Community Server for over study.

1.3 MongoDB

MongoDB [2] is the most popular NoSQL [3] database system. MongoDB comes under the document-oriented database type of NoSQL. MongoDB supports dynamic schema which makes the integration of data easier for some of the applications.

1.4 Comparisons Between MySQL and MongoDB

For making the comparisons between MySQL [1] and MongoDB [2] we are using the dataset Diabetes 130-US hospitals for years 1999–2008 Data Set—Beata Strack, Jonathan P. DeShazo, Chris Gennings, Juan L. Olmo, Sebastian Ventura, Krzysztof J. Cios, and John N. Clore, Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records, BioMed Research International, vol. 2014, Article ID 781670, 11 pages, 2014. The dataset contains the following attributes: Frequency, admission source id, number inpatient, acarbose, glimepiride pioglitazone, encounter id, time in hospital, diag 1, miglitol, metformin rosiglitazone, patient nbr, payer code, diag 2, troglitazone, metformin pioglitazone, race, medical specialty, diag 3, tolazamide, change, gender, num lab procedures, number diagnoses, examide, diabetesMed, age, num procedures, max glu serum, citoglipton, readmitted, weight, num medications, A1Cresult, insulin, admission type id, number outpatient, metformin, glyburide metformin, discharge disposition id, number emergency, repaglinide, glipizide metformin.

2 Problem Statement

Whether we choose relational database MySQL for our application or go for the NoSQL database MongoDB? We need our application to perform exceptionally well in the selection and aggregation operation. The performance of insertion and deletion should be taking medium time.

3 Related Work

In [4] author(s) had undertaken the study and analysis of the real-time disaster dataset belonging to states of India from 2010 to 2013. The data was imported from excel file into the MongoDB (NoSQL Document Database). Data was extracted from MongoDB by writing the NoSQL queries than the graphs were plotted using the extracted data. In [5] author(s) discuss about the history and the emergence of big data, how traditional DBMS could not compete with large data set and what are the issues and challenges of big data and the tools currently being used to implement and analyze the big data. In [6] author(s) gave the overview of Big data and its related components like cloud computing, distributed computing, data mining, etc. The paper introduced 3 V limitation of relational databases system which is at the roots of origin of Big Data. The paper discussed Big Data application(s), Big Data datasets and Big Data Tools. In [7] author(s) shown the analysis of large wireless network dataset which possessed real-time movement traces of the nodes. The author(s) designed relational algebra and SQL queries which were ran in the hive editor of hadoop software after importing it into hadoop software file browser against the user inputs from a web page. In [8] author(s) had beautifully contrasted between the MongoDB, NoSQL open-source database and Oracle, relational commercial database on the basis of syntaxes and query execution in milliseconds. In [9] author(s) had proposed CK-Means algorithm to analysis data stored in distributed clusters and had presented new strategy to parallelize K-Means++. They had made comparison between K-Means (existing clustering analysis algorithm), SKMeans (existing clustering analysis algorithm) and their new CK-Means algorithm. In [10] author(s) had introduced Top 10 algorithms in data mining. These top 10 algorithms covers the topics: association analysis, classification, clustering, link mining, and statistical learning of data mining. This paper describes these 10 algorithms, their impact and further research areas in algorithms.

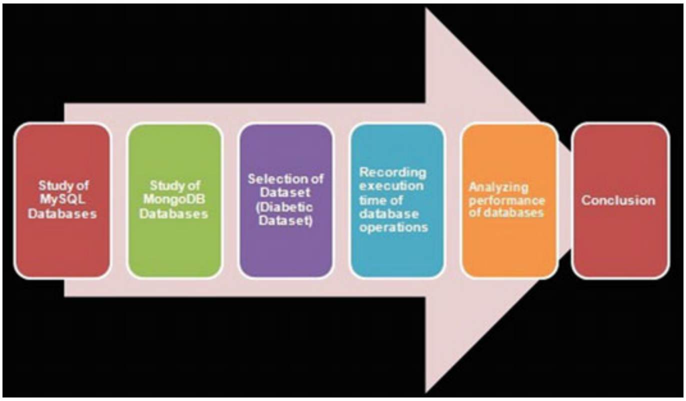
Study of MySQL Databases
Study of MongoDB Databases
Selection of Dataset (Diabetic Dataset)
Recording execution time of database operations
Analyzing performance of databases
Conclusion

Fig. 1 Block diagram indicating the steps followed

4 Motivation and Methodology Adopted

Selecting the suitable database for the software is a big task therefore, in this research paper, we are contrasting between MySQL and MongoDB on various database operations. The flowchart in Fig. 1 gives the overview of the methodology adopted for this research paper.

5 Simulation Performed

In this research paper as discussed we had chosen two databases MySQL and MongoDB. For these two databases we had recorded the query execution time in seconds for various databases operation. Table 1 gives the execution time for creation of schema and Insertion on table. Table 2 gives the execution time for deletion in table and dropping of schema. Tables 3, 4 and 5 contains the syntax of queries of MongoDB and MySQL. Tables 6, 7 and 8 gives the execution time for 10,000, 50,000 and 100,000 records, respectively. Table 9 gives the execution time for 100,000, 50,000, 100,000 records for various number of condition. Using the values in Table 1 we had plotted the line graph in Fig. 2. Using the values in Table 3 we had plotted the line graph in Fig. 2. Using the values in Tables 6, 7 and 8 we had plotted the line and bar graphs in Figs. 3, 4, 5, 6, 7 and 8 to analysis the performance of MySQL and MongoDB.

Table 1 Creation of schema and insertion in table

Number of records	MySQL (s)		MongoDB (s)
	Creation of table	Importing data	Importing data
1	0.30	0.11	0.015
10	0.23	0.21	0.047
100	0.32	0.20	0.110
1000	0.29	0.63	0.238
5000	0.34	1.86	0.780
10,000	0.31	1.96	1.607
20,000	0.23	2.46	3.510
30,000	0.28	3.21	5.290
40,000	0.27	3.59	6.800
50,000	0.30	5.29	8.900
60,000	0.25	4.99	20.000
70,000	0.22	5.87	13.630
80,000	0.36	6.54	12.886
90,000	0.28	7.46	53.870
100,000	0.28	6.86	23.618

Table 2 Deletion in table and dropping of schema

Number of records	MySQL (s)		MongoDB (s)	
	Dropping of table	Removing of data	Dropping of collection	Removing of records
1	0.16	0.1	0.031	0.032
10	0.16	0.13	0.032	0.032
100	0.12	0.16	0.033	0.047
1000	0.12	0.16	0.032	0.203
5000	0.22	0.13	0.094	0.359
10,000	0.14	0.46	0.047	0.593
20,000	0.34	0.55	0.047	0.78
30,000	0.17	0.67	0.031	0.842
40,000	0.14	1.09	0.078	0.78
50,000	0.17	1.44	0.063	1.045
60,000	0.16	2.14	0.062	1.232
70,000	0.23	2.12	0.063	1.732
80,000	0.11	2.09	0.046	2.652
90,000	0.11	2.43	0.062	3.604
100,000	0.14	1.89	0.032	5.039

Table 3 Syntax for distinct values query

Missing values %	Column names	MySQL	MongoDB
No missing values	Gender	SELECT distinct(‘gender’) FROM ‘total10000’;	db.Total10000. distinct (“gender”)
1 % missing value	Diagnosis 3	SELECT distinct(‘diag 3’) FROM ‘total10000’;	db.Total10000. distinct (“diag 3”)
2 % missing value	Race	SELECT distinct(‘race’) FROM ‘total10000’;	db.Total10000. distinct (“race”)
52 % missing value	Payer code	SELECT distinct(‘payer code’) FROM ‘total10000’;	db.Total10000. distinct (“payer code”)
53 % missing value	Medical specialty	SELECT distinct(‘medicalspecialty’) FROM ‘total10000’;	db.Total10000. distinct (“medicalspecialty”)
97 % missing value	Weight	SELECT distinct(‘weight’) FROM ‘total10000’;	db.Total10000. distinct (“weight”)

Table 4 Syntax for summary tables query

Missing values %	Column names	MySQL	MongoDB
No missing Values	Gender	SELECT gender,Count(*) FROM ‘total10000’ group by gender;	db.Total10000.group({key: {gender:1},cond: {},reduce:function (curr, result) {result.total += curr. Frequency;},initial: {total: 0}})
1 % missing Value	Diagnosis 3	SELECT diag3,Count(*) FROM ‘total10000’ group by diag 3;	db.Total10000. group({key: {diag3: 1}, cond: {}, reduce: function(curr, result) {result.total += curr.Frequency;}, initial: {total: 0}})
2 % missing Value	Race	SELECT race,Count(*) FROM ‘total10000’ group by race;	db.Total10000. group({key: {race: 1}, cond: {}, reduce: function(curr, result) {result.total += curr.Frequency;}, initial: {total: 0}})

6 Result Obtained

Below are the results obtained from the above simulations:

1. Figure 2 shows that MySQL is far better than MongoDB for creation and insertion operation. For deletion operation, Till 70,000 records the performance of MongoDB is better than MySQL, but after 70,000 records the performance of MySQL is better than MongoDB.

Table 5 Syntax for selection condition query

Number of conditions	Column names	MySQL	MongoDB
1 condition	encounterid	SELECT * FROM 'total10000' WHERE 'encounterid' = "248916";	db.Total10000. find(encounterid: 248916)
2 condition	encounterid, patientnbr	SELECT * FROM 'total10000' WHERE 'encounterid' = "236316" and 'patientnbr' = "40523301";	db.Total10000. find ({encounterid: 236316,patientnbr: 40523301})
3 condition	encounterid, patientnbr, gender	SELECT * FROM 'total10000' WHERE 'encounterid' = "253380" and 'patientnbr' = "56480238" and 'gender' = "Female";	db.Total10000. find ({encounterid: 253380,patientnbr: 56480238,gender:"Female"})

Table 6 Query execution time for 10,000 records

Query executed for	MySQL (s)		MongoDB (s)	
	Distinct values	Summary tables	Distinct values	Summary tables
No missing values	0.04	0.39	0.078	0.390
1 % missing values	0.05	0.05	0.109	0.639
2 % missing values	0.05	0.03	0.078	0.406
52 % missing values	0.03	0.05	0.093	0.406
53 % missing values	0.05	0.05	0.125	0.421
97 % missing values	0.05	0.05	0.078	0.437

Table 7 Query execution time for 50,000 records

Query executed for	MySQL (s)		MongoDB (s)	
	Distinct values	Summary tables	Distinct values	Summary tables
No missing values	0.19	0.19	0.203	1.545
1 % missing values	0.09	0.14	0.359	1.716
2 % missing values	0.14	0.16	0.156	1.311
52 % missing values	0.11	0.08	0.156	1.295
53 % missing values	0.12	0.11	0.405	1.42
97 % missing values	0.14	0.16	0.375	1.28

Table 8 Query execution time for 100,000 records

Query executed for	MySQL (s)		MongoDB (s)	
	Distinct values	Summary tables	Distinct values	Summary tables
No missing values	0.23	0.25	0.25	2.434
1 % missing values	0.19	0.23	0.608	3.12
2 % missing values	0.2	0.25	0.234	2.605
52 % missing values	0.14	0.17	0.312	2.636
53 % missing values	0.22	0.23	0.811	2.855
97 % missing values	0.16	0.2	0.764	2.465

Table 9 Deletion in table and dropping of Schema

Number of records	MySQL (s)			MongoDB (s)		
	Number of conditions			Number of conditions		
	1	2	3	1	2	3
100,000	0.09	0.03	0.13	0.069	0.078	0.078
500,000	0.20	0.14	0.19	0.110	0.078	0.078
1,000,000	0.30	0.31	0.33	0.184	0.125	0.125

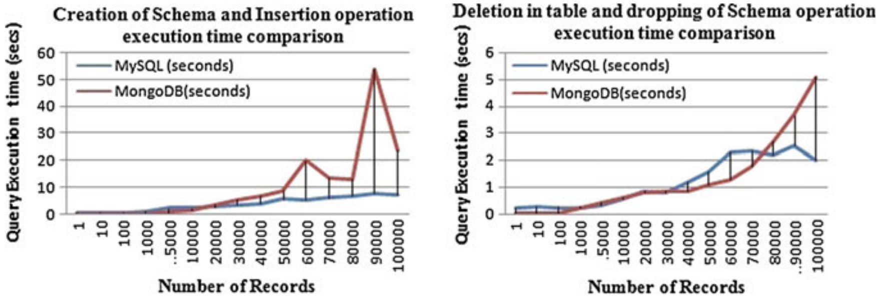

Fig. 2 MySQL and MongoDB comparisons on creation, insertion, and deletion operation

2. Figures 3 and 4 shows that the query execution time increases as the number of records increases.
3. Figure 5 shows that MySQL is better than MongoDB for distinct values query for various number of records.
4. Figure 6 shows that MySQL is better than MongoDB for summary tables queries for various number of records.
5. Figure 7 shows for MySQL selection query execution time as the number of records increases the selection query takes more time to select the records. To select the record on basis of three conditions takes more time than to select the

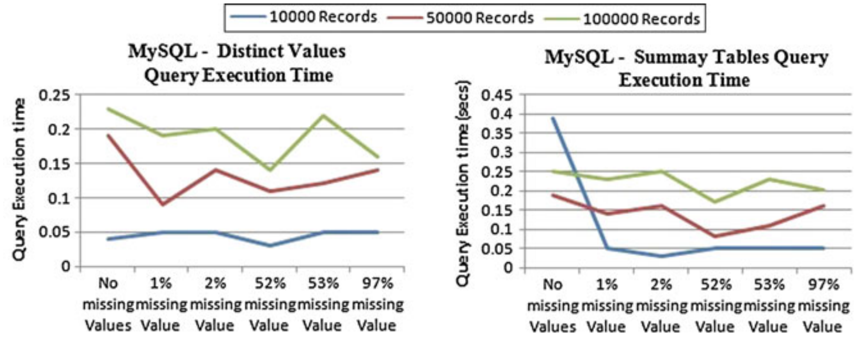

Fig. 3 MySQL database performance for various number of records

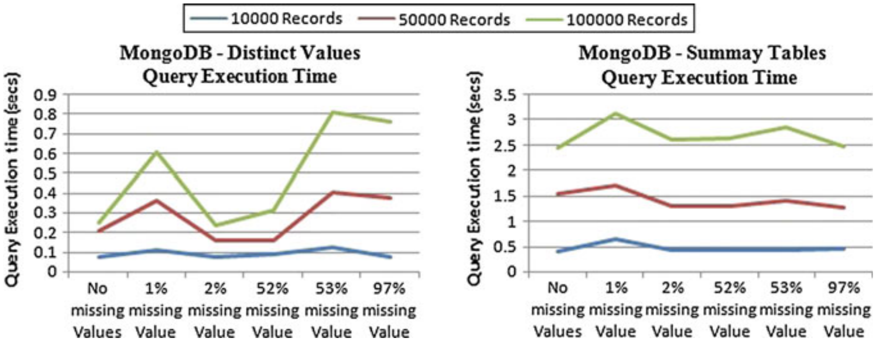

Fig. 4 MongoDB database performance for various number of records

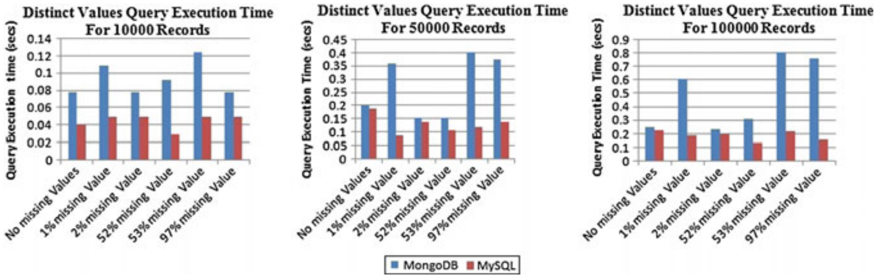

Fig. 5 MySQL and MongoDB comparison on distinct values query for various number of records

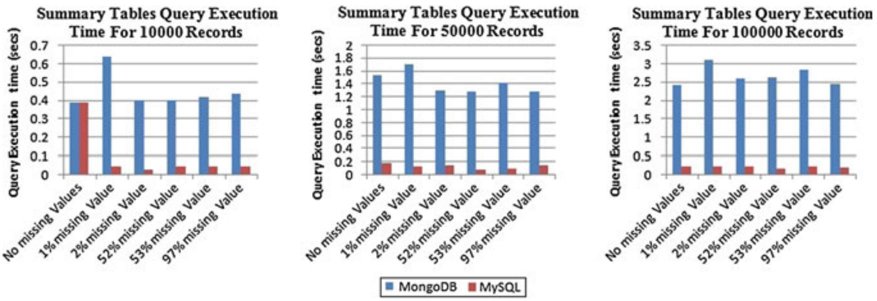

Fig. 6 MySQL and MongoDB comparison on summary tables query for various number of records

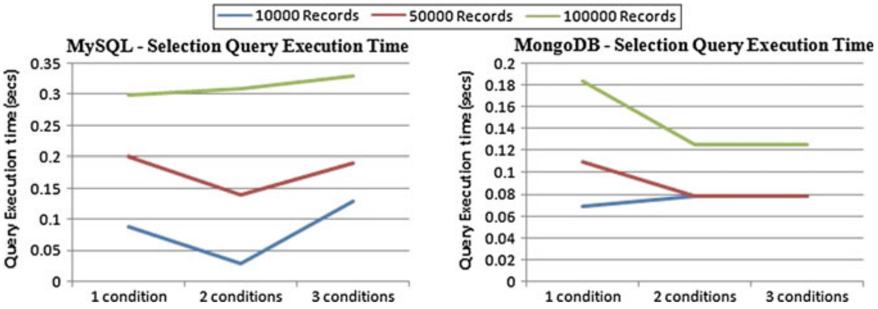

Fig. 7 MySQL and MongoDB performance on selection condition query for various number of records

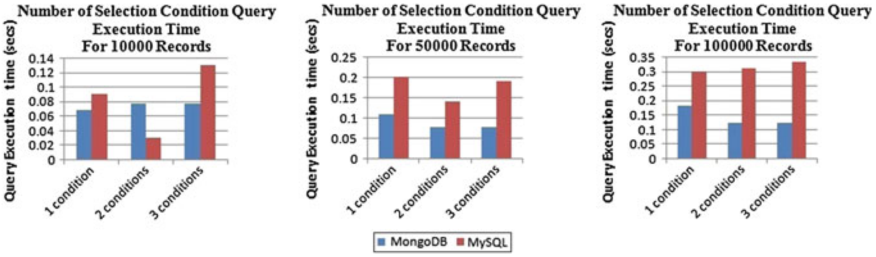

Fig. 8 MySQL and MongoDB comparison on selection condition query for various number of records

record on one or two conditions. There is very less difference in query execution time in MongoDB selection query execution time when selection is based on two and three conditions.

6. Figure 8 shows that MongoDB is better than MySQL for selection queries for various number of records.

7 Conclusion

The simulation results shows that "selection operation" is better performed in MongoDB for infinite numbers of records where as for "summary table operation" in MySQL exhibits better performance than in MongoDB. For Insertion of records there is no need for building schema in MongoDB whereas for MySQL it is necessary to built schema before inserting the records. The "Syntax for queries" is simpler and easier in MySQL as compared to MongoDB.

8 Future Work

In Future, same simulation of queries can be performed on more than 100,000 records. The simulation can also take into account more database operations. The contrast between the two open-source database technologies can be done by building the web application.

References

1. http://dev.mysql.com/downloads/mysql
2. https://www.mongodb.org/
3. https://en.wikipedia.org/wiki/NoSQL
4. Grover, P., Johari, R.: Disaster Big Data AnalysiS WIPS Symposium. In 3rd International Conference on Big Data Analytics (BDA), Delhi University, Springer, Dec 2014
5. Bhardwaj, V., Johari, R.: Big Data Analysis: Issues and Challenges. In IEEE International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), VIIT, Visakhapatnam, Andhra Pradesh, Jan 2015
6. Grover, P., Johari, R.: BCD: Big Data, Cloud Computing and Distributed Computing. In IEEE Global Conference on Communication Technologies (GCCT-2015) Kanyakumari, TamilNadu, April 2015
7. Bhardwaj, V., Johari, R.; Priti bhardwaj query execution evaluation in wireless network using MyHadoop. In 4th IEEE International Conference on Reliability, Infocom Technologies and Optimization (ICRITO 2015), AMITY University, Sept 2015
8. Boicea, A., Radulescu, F.: Laura Ioana Agapin: MongoDB vs Oracle—database comparison: In: 2012 Third International Conference on Emerging Intelligent Data and Web Technologies (2012)
9. Esteves, R.M., Hacker, T., Rong, C.: A new approach for accurate distributed cluster analysis for Big Data: competitive K-Means. Int. J. Big Data Intell. **1**(1/2), 50–64 (2014)
10. Wu, C., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z., Steinbach, M., Hand, D.J., Steinberg, D.: In: 1 Top 10 Algorithms in Data Mining. Springer, London (2007)