

A Report on

Performance Analysis of Flight Management System on SQL Server and MySQL Relational Databases

Submitted by:

Gopi Chand Bodepudi (gbhmf@mail.umkc.edu)
Teja Devarapalli (tdgp2@mail.umkc.edu)
Sravan kumar Pagadala (spvmc@mail.umkc.edu)
Sindhusa Tiyyagura (st2t7@mail.umkc.edu)

Under the esteemed guidance of

Kendall L, Bingham, University of Missouri – Kansas City



University of Missouri – Kansas City

Fall 2019

Submitted on: December 6th 2019

Performance Analysis of Flight Management System on SQL Server and MySQL Relational Databases

Gopi Chand Bodepudi, Teja Devarapalli, Sravan Kumar Pagadala, Sindhusa Tiyyagura

Masters in Computer Science - Fall 2019, University of Missouri – Kansas City

Abstract: Relational DBMS is the most popular tool for data storage. We cannot always use Oracle DB and SQL server as Database for our applications as they are not open source. Now MySQL has gained popularity which is an open source. In simple terms, MySQL is considered good backend tool for Web applications while SQL server is widely used in enterprise level. In this paper we tried to compare the performance of Microsoft SQL server and Oracle MySQL in windows operating system. We have used Flight Management system to compare the two databases. The final analysis shows that SQL server is better performer than MySQL in terms of CPU time and execution time along with several different parameters which will be discussed further in this paper.

Keywords: MS SQL, MySQL, Flight management system, performance analysis.

I. INTRODUCTION

Flight Management System was used as a dataset for the database systems and was designed for maintaining the records of airlines. It records information of travelers and airline details, which includes services available in-Flight details, reservation details, traveler details etc. As this is a complex schema with more tables and relations, we consider that it would be best for analyzing the performance comparisons of MySQL and SQL Server. So, the main contribution of this project is to analyze the performance of two relational databases namely - SQL Server and MySQL server to find out which database is beneficial for applications similar to the Flight management system.

II. WHY MYSQL AND SQL SERVER?

MySQL is second most popular DB which is developed by Oracle and SQL server third most popular DB developed by Microsoft. SQL server is more used in Enterprise level while MySQL is used as database by individuals for their own applications.

Features of Oracle MySQL:

1. Platform independence
2. Execution is very fast.
3. Support multiple data types
4. Supports variable and fixed length records.

Features of Microsoft SQL Server:

1. Enables memory optimization.

2. Provides services like clustering and replication which allow instant recovering and data synchronization between SQL server and other DBMS systems.

III. Our Approach:

a. Design:

In this analysis, the user base is very less so there is low impact on the back-end. However, the results help in finding out which is best to choose based on the requirement among different database systems.

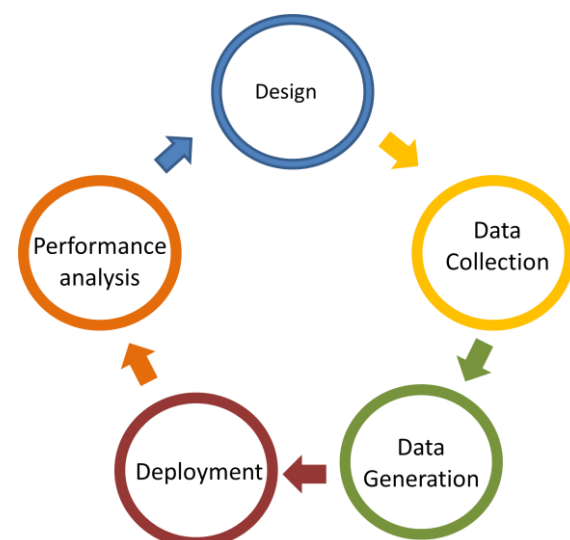


Figure - 1

The main focus of this paper is to find out which relational databases works well for the flight management system. In the first stage, schema of flight management systems is designed after thorough analysis of all the requirements. The UML diagram of the schema is shown in the (Figure - 2).

b. Data Collection:

In this study, the data is collected using RedGate tool which has ability to connect to both relational databases like MySQL and MS SQL server to generate dummy data with data constraints.

Using RedGate tool, the data is generated in MS SQL server database system. To have consistent data in MySQL server, transferring the data from MS SQL to MySQL server by generating the scripts in MS SQL and then executing the scripts in MySQL server. In the counter case, using the RedGate tool on both servers results in inconsistent data.

The data is generated in incremental order such as 10K, 20K, 30K records using RedGate tool.

c. Data Preparation:

Data cleaning is the process of finding and cleaning the unwanted, corrupted data from the auto-generated data. Making sure that the inconsistent data is eliminated from the tables and is not included for analysing performance.

The tool has the capacity to change the inconsistent and corrupted data. Data constraints like integer ranges, regular expressions were added to remove the inconsistent data.

d. Deployment:

Different queries and scripts are executed on databases and recorded execution times, memory utilization, CPU time and utilization.

All these records are later used to find out which database is more beneficial for similar type of applications based on execution times and memory Utilization.

e. Performance Analysis:

After executing the queries, all the results are recorded in either tables or graphs for further processing. The data is scaled from 10K to 30K in incremental order and analysed the performance metrics at each stage. These entries are used to do performance comparison between databases MS SQL server and MySQL server.

IV. PERFORMANCE ANALYSIS:

We have used the following parameters for performance analysis.

- Throughput (Reads, Writes, Read & Writes)
- Latency (Reads, Writes, Read & Writes)
- CPU Usage
- Memory Usage
- Query cost
- CPU time
- Execution time

We have written Python programs for calculating Throughput and Latency. By taking timestamps before and after the execution of operations and calculating the difference between timestamps gives Throughput and Latency.

1. Throughput for reads, writes, reads & writes. (Figure – 3a)

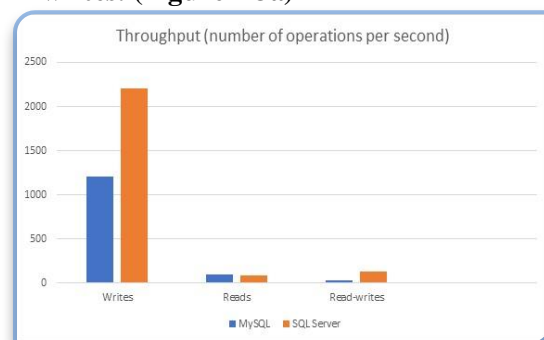


Figure – 3(a)

We have done the above analysis on a table inserting 20k records. More throughputs indicate faster performance of Database. For Writes and Read-writes SQL server outperformed MySQL but for Reads MySQL slightly performed better than SQL server

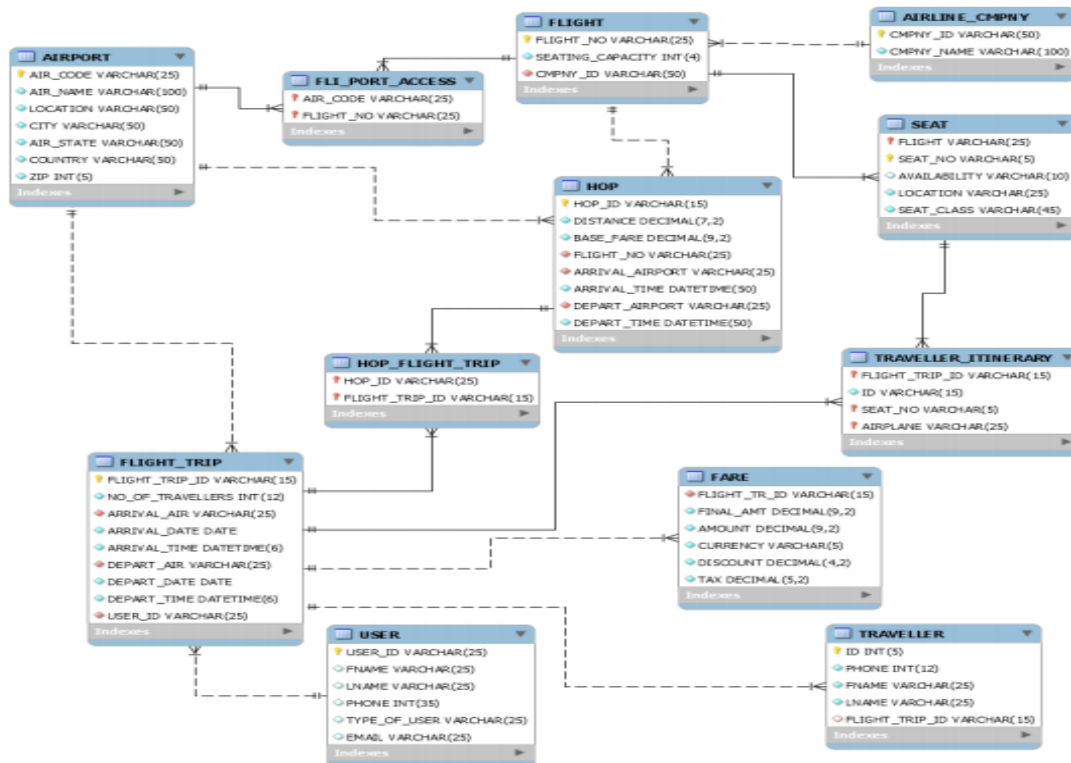


Figure - 2

2. How throughput changes on increasing data scalability for SQL Server? (Figure-3b)

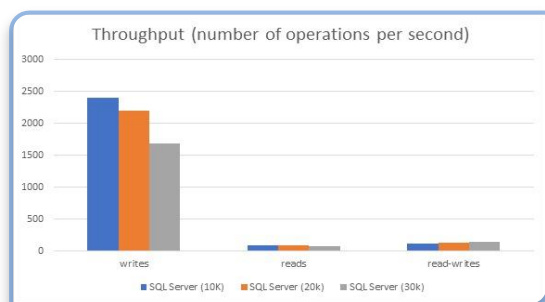


Figure - 3(b)

To test the scalability of the database, we have increased the data load from 10K to 20K to 30K. For Writes and Reads, the performance decreases gradually for SQL server. But for Read-writes the performance has got better.

3. How throughput changes on increasing data scalability for MySQL server? (Figure - 3c)

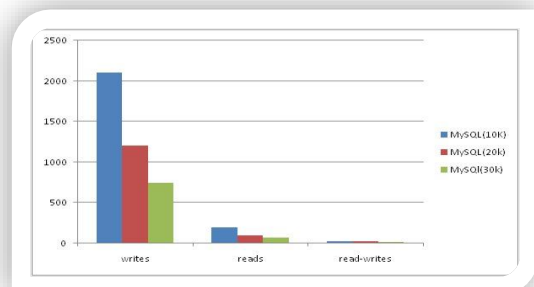


Figure - 3(c)

For MySQL also, the performance decreases gradually for all operations. But when compared to SQL server the performance change is more in MySQL. It indicates that SQL server is better choice for bigger systems.

4. Latency for reads, writes, reads and writes on both databases SQL server and MySQL. (Figure - 3d)

We have done the above analysis on a table inserting 20k records. Less Latency indicates faster performance of Database. For Writes and Read-writes SQL server outperformed MySQL but for Reads MySQL slightly performed better than SQL server.

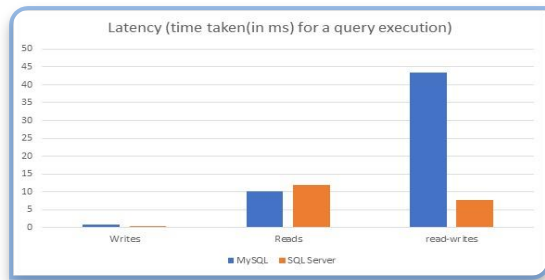


Figure – 3(d)

5. Latency comparisons of SQL server in terms of data scalability for 10k, 20k, 30k records. (Figure – 3e)

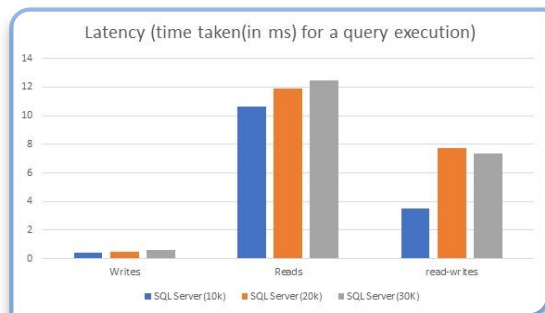


Figure – 3(e)

To test the scalability of the database, we have increased the data load from 10K to 20K to 30K. For Writes and Reads, the performance decreases gradually for SQL server. But for Read-writes the performance has got better.

6. Latency comparisons of MySQL in terms of data scalability for 10k, 20k, 30k records. (Figure – 3f)

For MySQL also, the performance decreases gradually for all operations. But when compared to SQL server the performance change is more in MySQL. It indicates that SQL server is better choice for bigger systems.

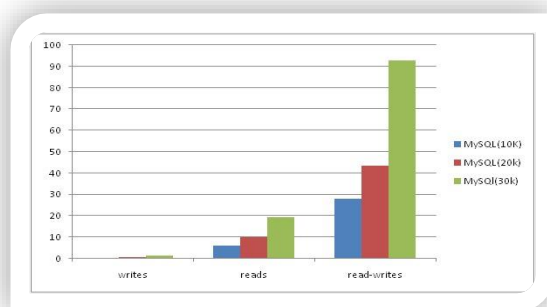


Figure – 3(f)

7. Simple Query with WHERE clause. (Figure – 4a, 4b)

```
SELECTR arrival_date,COUNT(*) as
no_of_passengers
FROM flight_trip
WHERE arrival_date = '23-10-2019'
```

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+2%	40KB	0.1859	125	433
MySQL	+3%	43KB	1.13	31	352

Figure – 4a

Using Indexes:

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+1%	40KB	0.1859	120	123
MySQL	+1%	43KB	1.13	28	232

Figure – 4b

We have written a simple query with where clause for checking other parameters shown above.

CPU usage, Memory usage, Query cost is more for MySQL, but CPU time and execution time is less for MySQL. This indicates that MySQL executes faster for Reads but occupies more memory than SQL server

In the second case we have created indexes on the columns we use in where condition and observed CPU usage, CPU and execution time decreases. So, when we use indexes the execution is faster without using indexes but there is no change in the memory usage.

8. List of trip details given Source and Destination.

Query with multiple JOINS and WHERE condition. (Figure – 5a, 5b)

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+3%	56KB	2.4349	109	431
MySQL	+3%	43KB	1.13	110	436

Figure – 5a

Using Indexes:

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+1%	56KB	2.4349	16	212
MySQL	+1%	43KB	1.12	21	220

Figure – 5b

Now we have written a query with multi joins. The CPU and Execution time is almost same for both but when compared with previous cases MySQL performance has decreased. As expected using Indexes yield faster results.

9. Use Case: List of trip details given the Source and Destination address.

Optimization: Query with Multiple JOINS and conditions with JOIN. (Figure – 6a, 6b)

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+3%	48KB	2.17	47	124
MySQL	+3%	39KB	2.11	53	132

Figure – 6a

Using Indexes:

	CPU Usage	Memory Usage	Query cost	CPU Time	Execution time
SQL Server	+1%	48KB	2.17	16	22
MySQL	+1%	39KB	2.10	21	43

Figure – 6b

In this case, we have optimized the above query. We can see that all parameters got improved because of optimization (takes less memory and executes fast).

V. CONCLUSION

From the analysis, Graphs and tables illustrates that in terms of execution time, SQL Server shows better performance than MySQL. There is double increase in the execution times for MySQL in case of increase in the data scalability.

SQL server performed poorly while scaling up but not as high as MySQL.

MS SQL server is faster than MySQL. Both databases have same CPU utilization. MS SQL server consumes more primary memory than MySQL. While using Indexes, there is huge decrease in the CPU Usage and the execution time.

In our future work, comparisons between different operating system platforms like Windows and UNIX in terms of execution time and memory usage. For further enhancement and better results, comparisons of performance can be done with different database systems and also increasing the comparison parameters.

VI. REFERENCES

- [1]. A Comparative Study on the Performance of the Top DBMS Systems, by Youssef Bassil in 2012.
- [2]. Comparative Performance Analysis of MySQL and SQL Server Relational Database Management Systems in Windows Environment, by Amlanjyoti Saikia¹, Sherin Joy², Dhondup Dolma³, Roseline Mary. R4 in 2015.