Performance Comparison of NOSQL Database Cassandra and SQL Server for Large Databases

[1]Khalid Mahmood

[1]*Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi Pakistan*

[1]khalidmdar@yahoo.com

***Abstract*--The performance comparison of NoSQL database and a Relational Database Management Systems has been done to identify which database responds faster to specific types of requests and suitability of these databases for different scenarios. Cassandra was taken as sample NoSQL database and its performance was compared with front line relational database SQL Server 2012.**

The new category of databases, the NoSQL databases are horizontally scalable as such these databases are very compatible for use at data centers that require very large size databases with variety of data types. Performance of SQL Server2012 and Cassandra was compared in a limited scenario but it was quite clear that for kind of database required for business, the relational databases are the choice. NoSQL technology is improving at a fast pace and different types of databases are coming into the market. New schema free environments and flexible table designs offer a lot to look forward. The four different types of NoSQL databases are providing specialized utilization for specific technology areas.

***Keywords*---**NoSQL, RDBMS, Distributed Databases, Schema Free, Performance Comparison

I. INTRODUCTION

A) Evolution of Relational Database Management Systems

Relational databases are in for the last 30 years when EF Codd came up with the relational model. The database systems progressed over the years and evolved into formidable systems having no competitor. Big names in this domain are IBM's DB2, Oracle Sybase and SQL Server by Microsoft. Relational Databases excelled in providing strong support for query languages, followed four distinguishing features of Atomicity, Consistency, Isolation and Durability (ACID). Mainstay of the business databases.

B) Emergence of NoSQL Database

There are numerous different types of database that emerged as a result of the advent of social media and large storage available at very low prices. Currently there are three new types of databases that are being used in the industry for processing these data types. Not only SQL NoSQL a generic name for all the databases that are not relational and for informational retrieval, it is not must to use query language. NoSQL databases allow storage and retrieval of data which was not entered in the relational databases. NoSQL databases, being schema-free, support easy replication, attain eventual consistent status and are capable of handling huge amounts of database. The main reason of having a NoSQL database is aimed at having simple design, horizontal scalability and a very fine control on the availability of the database.

C) Main Distinguishing Features of NoSQL Databases

The main distinguishing feature of NoSQL databases is having a different data structures when compared to RDMS. These structures result in making certain NoSQL operations faster compared to relational databases. These databases supports only simple queries compared to relational databases where can be very complex making multiple joins. NoSQL databases do not have a fixed schema unlike the relational databases; rather the schema can be modified at run time, as and when required. These databases have been designed for working on clusters of servers that may or may not be at one location or one data center. The distributed nature of databases restricts achieving the status of; "eventually consistent". There are many different types of NoSQL database. Their suitability depends on its usage to solve specific problems. Different NoSQL data bases have special usage in different domains. Cassandra being one of the most popular and has been the mainstay of Facebook for very long time. In fact Cassandra was developed by Facebook.

D) Column Family Stores

These databases are meant for storing very large size data, particularly when the data is distributed across many servers, may be located at different locations. There can be multiple keys pointing to many columns. These columns are combined into tables called column families. Cassandra and HBase are the popular Column Family Stores.

E) Motivation

Comparative studies exist for same class of databases, like all NoSQL databases [1]. There is a study on comparative performance evaluation of MySQL and MongoDB at University of Edinburgh [2]. On the other hand, various studies have compared Relational Database Management Systems [3]. Comparison of Cassandra vs. Microsoft SQL Server compares the System Properties but not the comparative performance in read and write operations [4].

Performance comparison between Cassandra and SQL Server will provide opportunity to decide when a user should switch from a Relational DBMS to a NoSQL database and vice versa. The focus of this study is to measure time taken on write and read operations in both the databases on single node operations for large database.

F) Experimental Framework

For comparison of the databases three different operations would be used on both the databases. Tables have been created to contain 1,000,000 records considered enough to check performance on a single node. The operations planned are:

1. **Read.** This reads the data against a key from the key-value pair storage in Cassandra and a key in SQL server. This corresponds to Select operation used in Relational databases that corresponds to Read in 'Create, Read, Update, and Delete' (CRUD).

2. **Write.** Data saved in other formats can be written into Cassandra, if related data is not available in the database, it is updated. Write operation combines the Create and Update operations as in relational databases.

3. **Select.** Cassandra supports select statement and SQL like statement using CQL can be used to select the desired data. Simple select statement have a constraint of selecting not more than 15,000 records at a time.

II. LETERATURE REVIEW

A) NoSQL: Non-Relational Databases of Next-Generation

Distributed databases are now the standard for storage of data for the Web2 applications being used by all front line social media organizations like Google, Facebook, LinkedIn, Twitter and Yahoo!. All these are processing very large databases of the scale petabytes. Although RDBMS do provide simplicity, robustness and performance, they have limitation of flexibility to scale with database application, be it on the Grid or an implementation on the cloud. These next generation databases are generally distributed, open source and very much scalable horizontally. NOSQL databases are non-relational, support easy replication and are mostly schema-free, no-join, and support easy replication. A good understanding of the design of non-relational database, comparison between relational and NOSQL architecture has identified important research directions in this important area [5].

B) A Distributed Storage System - Cassandra

Cassandra has been developed to be used as decentralized, distributed storage for very large databases that are spread over numerous commodity servers, providing reliable service catering for failure of one or more nodes. Cassandra can run on machines that may be spread across multiple locations with likelihood of multiple failures. Cassandra can manage the persistent state with these failures providing scalability and reliability of different systems that are using this service. Though Cassandra is similar to relational database to certain extent, it does not have features of RDBMS but provides dynamic control over how data has been laid out. Facebook the largest social networking platform, serves millions of users uses tens of thousands of servers which are housed in data centers across the globe [6].

C) Performance comparison of NoSQL databases and SQL Express

Most of NoSQL databases store data as key-value pairs on the premise. High speed Internet and cheap storage has encouraged capturing all kind of semi-structured, and unstructured data from variety of applications in the organization. A new term Big Data has emerged that includes all kinds of data coming from multiple sources. Processing of Big Data requires speed, flexible schemas, and distributed databases. Comparison was focused on read, write, and delete operations on the multiple key-value databases. It was found that there were wide variation of performance within NoSQL databases. It was also observed that there was practically no correlation in the data model used and the corresponding performance.

Comparison has been primarily made between NoSQL databases that include Couch base, MongoDB, Cassandra, Hypertable, Couch DB and Raven DB and a relational database SQL Express. It was found that not all NoSQL databases perform better than the SQL Express. Within NoSQL databases, a wide deviation was found depending upon the type of operation. [7]

D) RDBMS to NoSQL, a Continuous Evolution

Relational databases are not proving suitable for new generation web based applications supporting millions of users and data distributed across multiple servers. The new technologies named NoSQL database have now developed enough and offer very cost effective solutions for mobile and web applications. The new NoSQL databases support applications with large transaction volumes but need or can perform with low latency. To meet the complexity of database for web applications, companies started building their own databases for their special workload. These in-house developments are the main inspiration behind the current NoSQL databases. The authors have correctly identified the scenarios when the organization should move out from relational databases to NoSQL databases. The main focus would be type of application that has been written, the kind of queries that the users expect and any variations that may be expected in database design [8].

III. METHODOLOGY FOR PERFORMANCE COMPARISONEVIEW

A) Configuration

The tests were run on a machine with I3 processor. SQL Server 2012 was used for performance comparison. For Cassandra, DataStax installation of Cassandra 3.4 was used. PC configurations was as under:

System Type:	x64-based PC
Processor:	Intel(R) Core(TM) i3-2310M CPU @ 2.10GHz, 2 Core(s), 4 Logical Processor(s)
Total Physical Memory:	3.94 GB
OS:	Microsoft Windows 10 Professional

B) Relational database Management Systems: SQL Server 2012

For testing the performance, AdventureWorks2012 sample database in SQL Server 2012 was used. A special table salesorderheader1m was created using SQL Server SalesOrderHeader table by running SQL script and a special table was built for running the tests. The table had a row count of 1,000,000 with 26 columns of different data types and Data Size of 226.844 MB.

This table was specially created to carry out tests on a single processor machine for ease of handling. The test started with an empty database created in Cassandra with same attributes but assigned Cassandra data types. The data from SQL Server was exported to CSV file for subsequent loading onto Cassandra. Tests for Cassandra and SQL Server checked for any ongoing processes, and waited until those completed before continuing. This was done because Cassandra and even SQL Server performance degraded when some application was already running.

C) Cassandra Setup

The Cassandra version 3.4 was installed from DataStax. It received the IP from the personal computer used for the test purpose. For Cassandra testing, Keyspace and column family were created by running the commands via the cqlsh command line utility. For creation of Keyspace following command was used.

```
CREATE KEYSPACE testdat WITH REPLICATION =
{'class': 'NetworkTopologyStrategy', 'dc1' : 3 };
```

Column family .salesorderheader1m was created with same attributes of SQL Server table.

IV. COMPARATIVE ANALYSIS

For performance analyses, following three tests were carried out.

a. Comparison of Read Performance of csv into database
b. Comparison of Write Performance of database table into csv format
c. Comparison of SELECT Performance for limited number of records

A) Import Data Analysis

Importing of database into SQL Server and Cassandra, the performance of SQL server was far superior. While Cassandra took on average over 321 seconds, SQL Server took on average about 34.42 seconds. These results show that SQL Server has better throughput when exporting data to another data type as shown in figure 1.

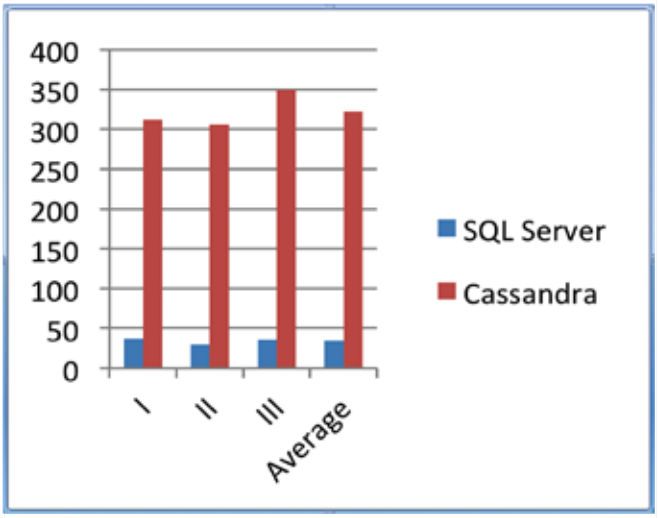

Fig. (1). Import of Data by SQL Server and Cassandra

B) Export Data Analysis

In exporting data, the performance of SQL Server was again very superior with respect to Cassandra. Whereas, Cassandra took on average of 322.6 Seconds to export 1,000,000 records, SQL Server took just 20.3 Seconds as shown in figure 2.

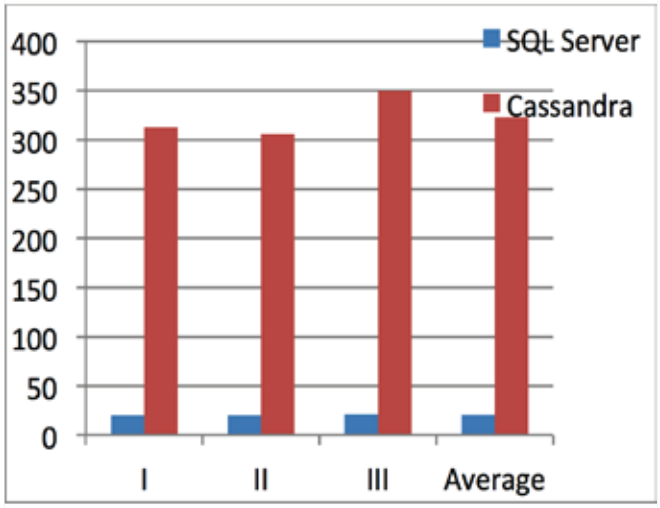

Fig. (2). Export of Data by SQL Server and Cassandra

C) Select / Read Records

Time was measured for select command for all the attributes of the table. While SQL Server was comfortable in selecting and displaying all the records at a time, for Cassandra, had to specify a limit which had a maximum value of 15,000. Thus three measurements were taken to select 5,000, 10,000 and 15,000 records for both the databases. Again SQL Server performance was better than Cassandra. Cassandra in cql shell had a page limit of 100 records which could not be increased. For this Devcenter for Cassandra had to be installed. This installation was rather tricky as it required 64 bit JVM environment. The tests for Cassandra were run and the comparative results are as shown below in figure 3:

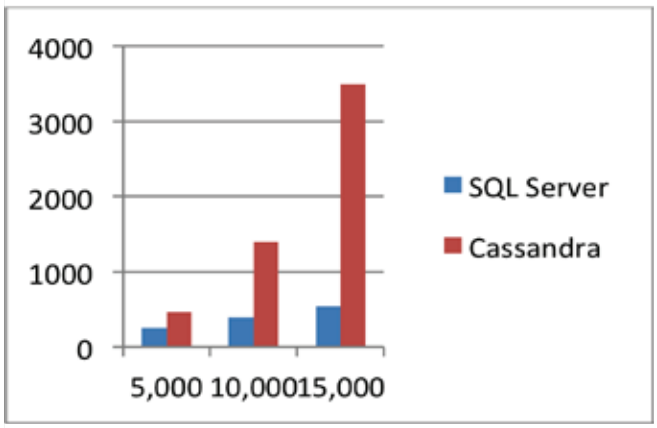

Fig. (3). Select /Read Records by SQL Server and Cassandra

V. CONCLUSION AND FUTURE WORKS

A) Conclusion

This study was an aimed at investigating and comparing performance and scaling of a NoSQL database and a Relational Database Management Systems. The performance of the both the databases was explored to a limited extent, to find which database responds faster to specific types of requests and suitability of these databases for different scenarios. Which technology is more suitable than the other and under what circumstances? The relational databases developed in 80s with specific structures in mind and built to have tables with columns and rows and pre-defined schema. The most important aspect, the database schema gives a logical view of the database and relations between tables, thus allowing creating databases which are very quick to respond and easy to design, with guaranteed reliability and technically no duplication. The new category of databases, the NoSQL databases, are relatively new and have become popular as they provide horizontal scalability which has made these databases very suitable for data centers that require very large.

The study tested, compared and analyzed the performance of the two databases SQL Server2012 and Cassandra. The experiments that were done on the two databases on just a PC with 4 GB Ram and an I3 processor were constrained. Complex queries could not be run on the machine. Remarkable performance difference was in the import and export of data. The data that took over three minutes using Cassandra was exported in just about 20 seconds. Same was the case of import of csv file with one million records that took about similar time with the two databases, for selecting data through an SQL commands on SQL Server 2012 and CQL on Cassandra had remarkable performance difference. The queries that fetched 5000 records, SQL Server performed twice as fast. When the number of records fetched increased to 10,000, the time had increased by three times. Increasing the data extracted to 15,000 records, the time increased by 7 time. This aspect could not be further checked as Cassandra gave error for selecting over 15,000 records.

For comprehensive performance, a good option would be acquire resources from cloud where NoSQL database could enjoy the distributed environment and could demonstrate enhanced processing capability. Machines with larger RAM and multiprocessing environment are more conducive for NoSQL databases whereas Relational Databases would perform much better on a single server with extended processing capability. NoSQL technology is evolving and improving every day with new schema free environments and very flexible database table designs. The four different types of NoSQL databases are providing specialized utilization for

specific technology areas. Whereas the Key-value stores are somehow the simple type of database management systems, they store pairs of keys and values. The data can be retrieved only when the key to the record is known. Not suitable for very complex database designs, their simple architecture makes these systems suitable in specific applications. Their main applications are in embedded systems. They are also used in in-process databases where high performance is the key. The Column Family databases are good for storing very large size databases, especially for distributed environment when the data is distributed over many servers. Multiple keys pointing to multiple columns may be generally arranged into column families. Cassandra is one very popular Column Family Store.

Another NoSQL databases, the Document-Oriented databases facilitate storage, retrieval and managing semi-structured data. These are a kind of key-value stores. The difference is how they process the data; a key-value store considers the data to be somewhat transparent to the database, but a document-oriented system may use the internal structure of the document to retrieve metadata used by database engine. Couch DB, Mongo DB are popular Document Databases. The fourth type of NoSQL database, the Graph database uses graph data model that is flexible and can be very comfortably scaled across multiple servers. Again, Graph Databases do not offer any advanced query processing like SQL and thus avoid overtime in handling joins. To run queries on such databases is specific to the data model. Neo4J, Infinite Graph and InfoGrid are popular Graph Databases.

B) Future Work

For a clear line of thinking, there is need for further investigation into defining the linkage between the type of database and its applications in the industry. The four databases discussed above are offering options for different applications but the processes are so very complex that new users are reluctant to adopt the new technologies and as such are not benefiting from the new databases. There is need to develop database systems with IDEs like the relational databases and a standardized language interface for convenient handling.

REFERENCES

[1] Benchmarking Top NoSQL Databases Apache Cassandra, Couch base, HBase, and MongoDB, [Online]. Available: http://www.endpoint.com/.

[2] C. Hadjigeorgiou, "RDBMS vs NoSQL: Performance and Scaling Comparison." M.S. dissertation, The University of Edinburgh, 2013.

[3] Y. Bassil, "A comparative study on the performance of the Top DBMS systems," *Journal of Computer Science & Research*, vol. 1, no. 1, pp: 20-31, 2012.

[4] System Properties Comparison Cassandra vs. Microsoft SQL Server, [Online]. Available: http://db-engines.com/en/system/Cassandra%3BMicrosoft+SQL+-Server

[5] R. P. Padhy, M. R. Patra, and S. C. Satapathy, "RDBMS to NoSQL: Reviewing some next-generation non-relational databases." *International Journal of Advanced Engineering Science and Technologies*, vol. 11, no. 1, pp: 15-30, 2011.

[6] Lakshman and P. Malik. "Cassandra: a decentralized structured storage system." *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp: 35-40, 2010. Doi: 10.1145/1773912.1773922

[7] Y. Li and S. Manoharan. "A performance comparison of SQL and NoSQL databases." In *Proceedings of IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, 2013.

[8] C. Nance, T. Losser, R. lype and G. Harmon, "Nosql vs rdbms - why there is room for both." In *Proceedings of the Southern Association for Information Systems Conference*, 2013.