ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 7, Issue 3, pp: (1-8), Month: July - September 2019, Available at: www.researchpublish.com

Performance Evaluation of Relational Databases and Non-Relational Databases

[1]Dr. Fahad A al Qurashi, [2]Maha H Alotibi

[1]Department of Computer Science King Abdul Aziz University, KAU, Jeddah, Saudi Arabia

[2]Department of Computer Science King Abdul Aziz University, KAU, Jeddah, Saudi Arabia

Abstract: **Currently, it is noticeable that NoSQL databases have been increased in popularity, which was due to the current emphasis on "Big Data". Such databases are believed to implement well again the traditional SQL databases. Therefore, this research aims to investigate independently the performance of SQL databases and some of NoSQL DBs, taking into consideration the key-value stores, compare the Redis database operations such as write, read, and delete implemented on key-value stores with SQL and NoSQL databases. An abstract framework of pair of key-value that support these processes is implemented and designed using all the tested databases. The outcomes of the experiment compute the time taken by these operations. The findings prove that not necessary all NoSQL act better than the SQL databases. Moreover, some of them were found to be much worse. In addition, the evaluation for each database showed that the performing be different with each operation, whereby some show a slow performance to read, but it is fast deleting and writing, and the rest were fast in reading, but slow in the other operations.**

Keywords: **Relational Database, non-relational database, SQL, NoSQL, RavenDB, Cassandra, Redis, MS SQL Express.**

1. INTRODUCTION

Relational Database Management Systems could be used for efficient storing and to query huge data. Common storage database systems were built by focusing on the relational model. The relational models are commonly identified as SQL databases [1]. In contrast to RDBMSs, non-relational databases (NoSQL) were designed often to only allow the eventual consistency to achieve additional improvements in performance and scalability. Lately, we observed that non-relational DBs have significantly increased in their popularity. Such non-relational models are also known as not-only SQL (NOSQL) databases. With the idea that simplicity would result in speed, most of these databases are storing based on simple pairs of key-value. However, along with the growth in availability and accessibility of cheap storage on the internet, huge amounts of unstructured, structured, and semi- structured data are stored and captured for a range of applications. these data known as Big Data [2]. Dealing with such huge quantity of data needs non-centralized, speed, and flexible schemas databases. Thus, NoSQL databases have turned out to be the favored tools for functioning Big Data, which demands to satisfy these needs. This has led to an increase in the number of NoSQL storage offerings. Some implementations of open-source and commercial NoSQL databases are available, such as HBase [3] and BigTable [4].

Thus, considering the several NoSQL offerings, questions have been raised about the variances between the offerings of such databases and their appropriateness in certain applications. Several survey articles therefore have been published to address some of the questions like Han et al. [5] or Tudorica and Bucur [6]. Furthermore, some blogs and online resources are handling these aspects as well. The aim of this paper was mainly to evaluate the performance of Relational Databases and Non-Relational Databases, and compare the key-value operations on both SQL and NoSQL databases. Generally, NoSQL databases were created for enhanced key-value stores, unlike the SQL databases. In this paper, the results indicate that not all of NoSQL databases are better than the SQL databases. In the theoretical section, a brief survey is provided, and in the practical segment, a comparison was made between the write, read and delete operations on the key-value database. It is notable that even within the NoSQL databases, there were some variations between the performances of these processes. Moreover, a slight correlation was noted between the model of data used by each database and their performance.

The remainder of this paper was organized as follows, the 2nd section will discuss the background of SQL, the 3rd section will discuss the background of NoSQL databases, the 4th section will compare between the relational and non-relational databases, the 5th section presents the executed experiment, and the final section will be comprise the results, discussion, and conclusion.

2. RELATIONAL DATABASE

Relational Database was initially developed based on the relational model of data proposed by Edgar F. Codd in 1970, as an application which allows data to be stored in such a way that makes it easily accessible and enables fast data retrieval. The collective data in relational database is arranged in tables according to different categories in rows and columns where each row contains instance of data uniquely to them. A relational database management system (RDBMS) is referring to a systematic arrangement of collective storage of data items, where the data is arranged as tables from the location of data is being retrieved while maintaining the update.

In the RDBMS model, the data in the database is arranged as queries grouped into relational table. The relationship between the tables in terms of generating idea or making assumption about data extraction from stored tables is what actually makes it a 'relation' table.

As a result, we could observe several different perspectives of the database in such way that the matching database can be seen in many dissimilar ways. Structured Query Language (SQL) is the most commonly used for the management of Relational Database by RDBMS data stores. The initial work of RDBMS is based on relational algebra and relational calculus and its divided element such as predicates, statement and queries.

The followings are the benefits of the RDBMS based on relational database model:

- Database is self-recording (documenting) because all or most of the data is actually stored in the database rather than in applications.
- It is easier to access, retrieve, update and maintain the database.
- Information retrieval, reporting and summarization supports are provided.
- The effectiveness of tabular form of the database while being structured with co-relative tables resulting in the database to be more predictable in nature.

2.1. Relational Database Tools

Oracle and MySQL are two of the most popular and commonly used relational databases. The use of MySQL is more widespread in the web oriented environment due to it being extremely fast. As for Oracle, it is more extensively used by ERP, insurance, banking and finance companies, where there is a need to fulfill large database requirement [2], solving complex and difficult problems and supporting large online transaction processing (OLTP) environments. Although, in general, they work in the same manner, there are some small differences that set them apart [2].

Oracle is a costly and expensive database that might be afforded by only large organizations. On the other hand, MySQL is an open source system which making it a more affordable solution for any organizations or user. However, despite the cost, MySQL is far behind in terms of the additional facilities and robustness which are supported by Oracle due to it being an open source system [3].

·In addition, Oracle has a large table space, snapshots, role management, synonyms and packages MySQL lacks any of the mentioned features [2].

· The syntax of Oracle is highly flexible whereby it utilizes integrated programming language such as PL/SQL. In that sense, Oracle has wider command structure than MySQL.

Oracle is also equipped with tighter and more bounded security with slightly more advanced parameters for security enhancements via creating profiles, external authentication and local authentication, whereas MySQL would only provide three security parameters, namely, user location, user id and user password [2,3].

· However, MySQL is more case sensitive especially for database and table names. Case sensitivity is not an issue for Oracle.

· Oracle uses XML for data transfer of which the languages are not supported by MySQL.

· VARCHAR and CHAR character type option is available in MySQL while Oracle provides 4 types of character type data sets which are as follows: CHAR, NCHAR, VARCHAR2, and NVARCHAR2 [2]

3. NON-RELATIONAL DATABASES

Non-relational database can be referring to a class of systems that succeed databases which is significantly different from the relational systems; where it would not utilize relations tables as its storage structure, therefore joins operations cannot be performed. It also would not use SQL as its query language, not be able to guarantee ACID properties and can be scaled horizontally. Today, there are a lot of classifications for NOSQL databases available. One of the classifications is based on a theorem (CAP theorem) which was discussed in the introduction section

[5]. NoSQL is one of an example of the description for databases that refuse the tradition of operating relational models such as the Relational Database Management Systems (RDBMS). On the other hand, the NoSQL databases family focuses more on delivering more distributed and scalable solutions when handling massive amounts of data which is only made possible by a more consistent relaxed model and less schema- oriented database designs in comparison to RDBMS [7]. The NoSQL would be a better pick for cases where the data to be kept does not conform to any easily deniable schema or in the case of the sparse tables in an RDBMS database (i.e. having a lot of columns where the column is only used by several rows). Furthermore, the main concern when dealing with relational model is that the data is expected to be stored in separate tables which are connected by logical relations and used for joining the tables when a more complex query arrives [5] which only works best for smaller size tables. There will likely be a drop in the performances with the increasing size of data and the number of required joins. NoSQL databases are highly improved for deal with huge data for cases when the particular elements are not closely related; therefore, it would be beneficial in terms of costing as it removes the need for expensive joins.

3.1 Tools of Non-Relational Database:

_ Key-value store:

The data are stored as key-value pairs. This data structure, ‘‘Hash table” as it more commonly known as, is where the keys retrieve the data. Some of the most well-known examples of key-value stores are Redis1, and Memcached2.

_ Document store:

The data are stored in collections that contain key-value pairs which encapsulate key value pairs in JSON (Javascript Object Notation) or JSON like documents. Some of the most well-known examples of document stores are MongoDB3, CouchDB4. Since the values are not opaque to the system, data can be queried by values and keys [8].

_ Column family:

The data are stored as a set of rows and columns where columns are grouped according to the relationship of data [9]. Some of the most well-known examples of document stores are Cassandra5, HBase6.

_ Graph database:

This type of databases is best used to represent data in the graph form. The most well-known example of graph databases is Neo4j7.

4. COMPARISON BETWEEN RELATIONAL AND NON-RELATIONAL

Here we will highlight and conclude the main dissimilarities between this two databases types as follows [10]

Table 1: Relational and Non-Relational database.

Relational database	**Non-** ***relational database***
Less scalable	Highly scalable
Caching done by help of special infrastructure	Enhanced the performance by caching data into system memory

Transactional	Multiple operation transactions can be implemented.
Low data throughput	High data throughput
Multiple column index.	Single index, key value store
Provide better consistency than non-relational databases	Compromise on consistency
Provides acid properties	Provide base properties
Eliminates duplication of record, preventing inconsistent data from occupying the database	"Data duplication is allowed which threatens data integrity. Also, updating one record means updating all other duplicate records"
"Its organization allows a query language to use the primary key shared between tables to efficiently and quickly return and collate the requested records"	"As they require more than one pass, Searching in Non-relational database is inefficient especially those based on multiple criteria"

5. RELATED WORK

Bartholomew [11]discuss the comparison of SQL and NoSQL databases, each with brief history and the use case of the databases. Tiwari further elaborate a detailed introduction on NoSQL databases with a comparison on the basis of the following features: (1) scalability, (2) transactional integrity and consistency, (3) data modeling, (4) query support, and (5) access and interface availability [10]. Hecht and Jablonski conducted a survey on the use case of NoSQL databases

[8] in which comparisons of NoSQL databases are made according to their query possibilities, data models, replication opportunities, partitioning and concurrency controls.

Boicea et al. [12] investigate Oracle and MongoDB databases in the comparisons of SQL and NoSQL database performing was conducted through Elapsed time to insert data, update data, and delete data. Various amount of data between 10 to 1,000,000 records were tested in these experiments. It was reported that for all operations, MongoDB provides better performance than Oracle.

Another attempt by Li and Manoharan [13] was performed to compare the performances of NoSQL databases by running five experiments: (1) Time to instantiate database bucket, (2) time to read values corresponding to given keys, (3) time to write key-value pairs, (4) time to delete key-value pairs, and (5) time to fetch all keys. Similar to the works from Boicea et al., various data from 10 records to 100,000 records were tested in these experiments. The databases tested are MongoDB, RavenDB9, CouchDB, Cassandra, Hypertable10, Couchbase11, and MS SQL Express12. It was found that Couchbase and MongoDB are the fastest two for read, write, and delete operations. They also note that Couchbase lacks fetching all keys from database.

Abramova et al. [9] engage in the use Yahoo! Cloud Serving Benchmark [14] in order to make evaluation and comparison of the performance of NoSQL databases 600,000 records were randomly generated and used with different workloads by changing ratios of read, update and insert operations. The databases used in the experimental evaluation are Redis, Cassandra, HBase, MongoDB, and OrientDB8. The in-memory database Redis was found to provide the best performance while column family databases, Cassandra and HBase showed good performances in update operations since they are optimized for such operations.

Jablonski and Hecht prepare a use-case oriented survey of NoSQL databases [15]. The worries in selecting a NoSQL database to fit a specific use-case was identified which was the main focus in their work. They use as the basis for their comparison the data model, support for queries, concurrency controls, partitioning, and replication. Many NoSQL databases, including CouchDB, MongoDB, HBase and Cassandra are compared for the purpose of their research.

6. EXPERIMENTAL FRAMEWORK AND RESULTS

In the current work, we will evaluate the performance of relational and non-relational databases. This work was based on the information obtained from a previous research [13] with the addition of the redis DB results to the comparison. The computer features were as follow: Windows 7 operating system Ultimate 64-bit, Intel Core i3 (2.4 GHz) processor, and 4 GB RAM memory. The data set for the experiment were auto-generated key-value pairs that were of the form (kN, vN), whereby N is a sequence number. It should be noted that the times were counting by averaging over five runs.

The databases used in this experiment were RavenDB, Cassandra, Redis, and MS SQL Express. RavenDB is a document oriented database, and the offering of flexible schema storage was considered as it is formatted for serialized objects. These databases also offered data sharing and replication. The Apache Cassandra database stores data according to the column family data model, and it is originally a product of Facebook [10]. Some of its key characteristics include data replication to increase fault-tolerance and de- centralization to reduce failures. Redis is a non- relational open source database that is used as a message broker database and cache. The version detail of database implementations is shown in table 2.

Table 2: used database and their versions.

Database	Version
RavenDB	960
Cassandra	1.1.2
Redis	3.0.7
Microsoft SQL Server Express	10.50.1600.1

First of all, we will evaluate the performance of Redis database since we execute this database, so our focus will be in it. Figure 1 show the Redis performance with the 3 operation (write, read, delete).

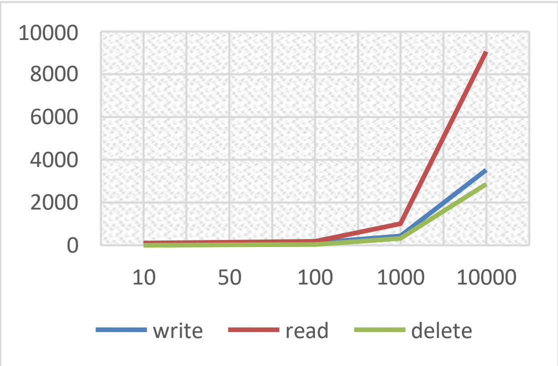

Figure 1: Redis database evaluation in (MS)

Redis provide fastest time with delete operation. And the slowest operation was read. we observed that Read and delete operation performed closely. in general, as the number of records increase, the time taken increase.

In the first experiment, the time taken to write values that match the presented keys from the databases was measured. The results of reading operations are summarized in Table 3. In the tables, the operations number refers to the number of times the reading operation was executed in the test. Moreover, it is the same as the number of key-value pairs in the store.

Table 3: Time for Writing of Different Database (MS)

Database	Number of operations				
	10	50	100	1000	10000
RavenDB	570	898	1213	6939	71343
Cassandra	117	160	212	1200	9801
Redis	86.2	93.7	118.4	433.8	3519
MS SQL Express	30	94	129	1790	15588

After sorting through delete performance, the list of databases were are follows: SQL Express, Redis, Cassandra, and RavenDB. We found that the writing performances of RavenDB and Cassandra was worse than that of SQL Express. Moreover, SQL Express is acting better than other NoSQL databases.

Table 4 summarizes the results of the second experiment, whereby the time taken for reading key- value pairs to the databases was computed. It amounts to adding the key-value pair to the database.

Table 4: Time for Reading of Different Databases (MS)

Database	number of operations				
	10	50	100	1000	10000
RavenDB	140	351	539	4730	47459
Cassandra	115	230	354	2385	19758
Redis	108.8	137.2	194.2	1014.8	9058
MS SQL Express	13	23	46	277	1968

After sorting according to read performance, the list of databases were as follows: Redis, Cassandra and RavenDB. Cassandra is a column-family database; RavenDB is a document-oriented database, and Redis is

a key-family database. The performance of read operation in SQL Express was found to be better than other NoSQL databases.

Table 5 summarizes the results of the third experiment,

i.e. computing the time taken to delete key-value pairs to the databases. It amounts to adding the key-value pair to the database.

Table 5: Time for Deleting of Different Database (MS)

Database	Number of operations				
	10	50	100	1000	10000
RavenDB	90	499	809	8342	87562
Cassandra	33	95	130	1061	9230
Redis	3.6	17	39.2	321.4	2862.8
MS SQL Express	11	32	57	360	3571

After sorting according to delete performance, the list of databases were as follows: Redis, SQL Express, Cassandra and RavenDB. It is observed that the write performances of RavenDB and Cassandra were worse than that of SQL Express. Moreover, SQL Express is acting better than other NoSQL databases. The delete performance of Redis was seen to be better than that of SQL Express and the other NoSQL databases.

7. DISCUSSION AND CONCLUSION

The main concept behind NoSQL databases are they based on BASE uniformity pattern rather than ACID uniformity pattern which may giving up some consistency to offer extra high performance, availability, and scalability. Currently there are more than 225 NoSQL databases which offer several characteristics and features . In this paper, we compare the implementation of key-value store on SQL and NoSQL databases we provide evaluation for one database from each type: RavenDB as a document store, Cassandra as column family, Redis as key-value stores, and MS SQL Express as relational database management system. he time taken to complete various data operations are experimented. The results got from the experiments could be listed as follow:

_ MS SQL Express clearly provides the best write performance in term of elapsed time then Redis come. RavenDB Performance significantly decreases as the size of the records increases by reason of locking mechanism of RavenDB.

_As the data increase we observed that Redis provides fairly best performance than other databases for the read operation. MS SQL Express got best result with small number of records but clearly provides the worse read performance as the records increase . in my opinion that the relational database architecture of MS SQL Express is the main reason behind this difference.

_ Redis provides clearly the best performance, When it comes to delete a key-value pair while RavenDB provides the worst performance.

In this test RavenDB do not act in good way with the write, read and delete operations. Casandra reasonably is good for delete and write operations, but is slow on read operation. MS SQL Express and Redis could be consider as the fastest two overall for delete, read, and write operations. Note that we have not tested databases for more complex operations. The database rankings we noted may not be held when it comes to complex operations.

Up till now, we observe that not all NoSQL DSs act better than the SQL DSs we tested. we see that even within NoSQL DBs there is on a large-scale variation in performance based on the operation's type "such as write, read and delete" . Also, slight association between the data model and performance each database uses is observed

However, NoSQL as well as SQL databases deliver dissimilar features and one cannot replace another. So, If the systems could hand over some consistency, then NoSQL DB may be a good option to give more scalability, high performance and availability. In other hand If the systems are not adaptable in term of consistency, then the RDMS is the correct option. As future work, compare different databases based on different terms

REFERENCES

[1] K. Kline, SQL in a nutshell, 3rd ed. O'Reilly Media, November 2008.

[2] P. Warden, Big Data Glossary. O'Reilly Media, September 2011.

[3] L. George, HBase: The Definitive Guide. O'Reilly Media, August 2011.

[4] F. Chang, J. Dean, S. Ghemawat, Wallach, M. Burrows,T. Chandra, and A. Fikes, "Bigtable: a distributed storage system for structured data," in Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation – Volume 7, ser. OSDI '06. Berkeley, CA, USA: USENIX Association, 2006, pp. 15–15.

[5] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," in Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, oct. 2011, pp. 363 –366. [6] P. Warden, Big Data Glossary. O'Reilly Media, September 2011.

[6] C. Bucur and B. Tudorica , "A comparison between several NoSQL databases with comments and notes," in Roedunet International Conference (RoEduNet), 2011 10th, june 2011, pp. 1 –5.

[7] PClarence J M Tauro, Aravindh S, Shreeharsha A.B "*Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases"* International Journal of Computer Applications (0975 – 888) Volume 48– No.20, June 2012.

[8] *R. Hecht and S. Jablonski, "NoSQL evaluation: A use case oriented survey", 2011 International Conference on Cloud and Service Computing, 2011..*

[9] Abramova, V., Bernardino, J., & Furtado, P. (2014). Which nosql database? a performance overview. Open Journal of Databases (OJDB), 1(2), 17-24.

[10] Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (2012). A survey and comparison of relational and non-relational database. International Journal of Engineering Research & Technology, 1(6).

[11] Bartholomew, D., 2010. SQL vs. NoSQL. Linux J., 54– 59 (doi: FacetnMap: A Scalabled Browse and Search Visualization).

[12] Boicea, A., Radulescu, F., & Agapin, L. I. (2012, September). MongoDB vs Oracle-Database Comparison. In EIDWT (pp. 330-335).

[13] Li, Y., & Manoharan, S. (2013, August). A performance comparison of SQL and NoSQL databases.In Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on (pp. 15- 19). IEEE.

[14] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., & Sears, R. (2010, June). Benchmarking cloud serving systems with YCSB. In Proceedings of the 1st ACM symposium on Cloud computing (pp. 143-154). ACM.

[15] Hecht, R., & Jablonski, S. (2011, December). NoSQL evaluation: A use case oriented survey. In Cloud and Service Computing (CSC), 2011 International Conference on (pp. 336-341). IEEE.