

"ANALYTICAL STUDY AND EVALUATION OF STRUCTURED AND NOSQL DATABASES FOR DATASET"

Falgunee Parsana¹, Dr. Achyut C. Patel²

¹Department of CS & IT, Atmiya University

²Smt. M. T. Dhamsania Commerce College, Saurashtra University

¹falguni.parsana@atmiyauni.ac.in, ²acp2809@gmail.com

Abstract

The aim of the study is to evaluate document-oriented MongoDB database with SQL in terms of Performance. For better reliability and efficiency of parallel computing, NoSQL techniques distributes the data randomly on data nodes. This strategy provides phenomenal benefits for general data accessing methods. Hence the migration as well as conversion of RDBMS to NoSQL is now a best choice or attention of many researchers in recent times.

In software development, migration from a RDBMS to NoSQL is a challenge for database administrators as well as programmers. However, due to explosive growth of big data, the well-structured format of Relational Databases may limit the scalability to maintain horizontal scaling. So to gain flexibility as well feasibility of parallel processing, the data storage methods take NoSQL databases into this consideration.

Keyword: NoSQL, RDBMS, Big Data

1. Introduction

RDBMS - Relational Database Management System can be use to effectively store huge amount of data. This system facilitates ACID (i.e., Atomicity, Consistency, Isolation and Durability) transactions. But RDBMS fails to give full transactional uniformity. In contrast, NoSQL - Not only SQL are designed to give further improvements of scalability and performance.

Architecture of traditional database is totally different from NoSQL Database. NoSQL works on more than one machine so performance of NOSQL database is summation of performance of single nodes in the database environment of cluster. This individual node concept is important for performance Tunning and resource building plans. All nodes may not five same amount of performance because due to capacity of its resources such as of disk, bandwidth network and physical main memory.

2. Research Methodology

Organizations now adopt NoSQL Databases because it able to help them to construct database structure quicker, handle enormous level of data, and deal with application more proficiently and suitably. MongoDB models are intended for easy access of statistics.

It provides new intensity of reliability and accessibility which are not providing by further conventional database. Due to high data quantity and throughput, MongoDB can effortlessly implement without a great deal of downtime, and with no changes of the application strategy.

An example is like a booking of ticket at the back of travel reservation model, which involve compound business. On the other hand the core reservation system may run on MySQL, individuals system part with users including serving booking, integrated with social networks, managing session would be improved when we placed in MongoDB. MongoDB came with the thought of giving the new method of data storage.

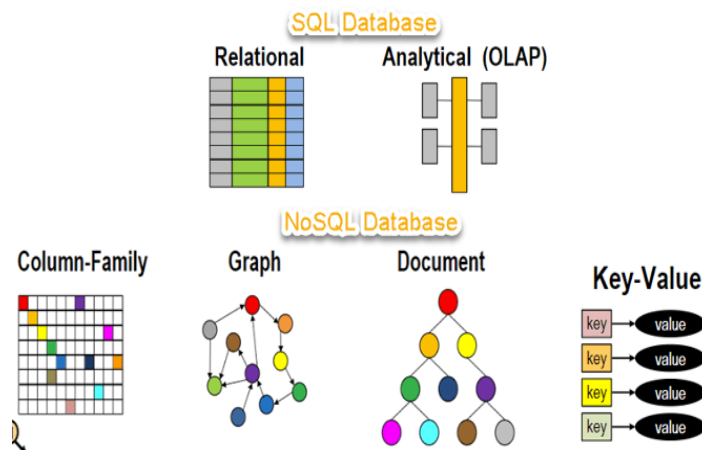


Figure 1

NoSQL association can be used to accumulate data in a active schema, instead of a tabular representation like SQL (Relational database). NOSQL structure like MongoDB customized JSON format into its BSON format, which accumulate the data as a binary format. So due to its binary arrangement provide more reliable and efficient structure in the area of storage space space and quicker speed.

3. Research objectives

As per the specific demands, Relational Database Management System store data in a very different way comparatively. Data store on relational data structure is highly structured and normalized where the data in NoSQL is typically unstructured. This dissimilarity of data structure helps to meet the particular loads of these two systems. This NoSQL Databases are most scalable with high accessibility due to simpler data model but also not gives consistency all times. On other hand, the SQL systems are not simply scalable with high availability due to multipart data model but consistency will

become guaranteed.

Data in RDBMs is stored in form of Tables and tables are the two-dimensional approach which is combination of rows and columns. Tables are interrelated with each other into the database through Primary key as well as foreign key. This interrelation is facilitating by SQL Joins.

So to aim of such proposed work is to evaluate the performance issues, scalability, high availability and flexible database model.

4. Problem Definition

Traditional and Unstructured databases use two completely different technologies. One important difference is that, Data model of unstructured database is different from the structured database. The traditional databases keep data in tables form and joint them in the help of primary keys and foreign keys. Such structure is highly normalized and schema full. The NoSQL i.e., Unstructured databases stores their data in the form of Documents. It stores the data differently. The data which are stored is schema free and De-normalized highly. Both these methods give different performances.

There are numerous structured and unstructured Databases which are presently in use. This paper mainly focusing on evaluating performance and analytical study of MySQL and NoSQL MongoDB document based model. The motivation for choosing MySQL and MongoDB is that it meet all necessities of current relational databases and becomes easy to manage and gives better performance.

5. Performance Evaluation

1. Transaction in MongoDB

MongoDB command Line Interpreter provides to insert, update and delete operations on Documents in a collection.

INSERT Transaction:

Insert a Single Document:

It inserts one or many documents into a single collection. If collection does not exist then it creates a new collection.

<code>db.collection.insertOne(document)</code>	Inserts a single Document into a collection
--	---

For example:

The following example inserts one document into the HospitalData collection. If the document does not specifying `_id` column then MongoDB inserts the `_id` column with Objectid value to this new document.

```
db.hospitalData.insertOne(
{Item: "medicine", quantity: 20, tags: ["fever"]}
)
```

READ Transaction:

To retrieve certain data from collections, here `find ()` method can be used.

<code>db.collection.find(document)</code>	Find() method can be used to find a data from a collection based to certain condition.
---	--

For example:

```
db.collection.find(
{ customer_id: {$gt: 200}
})
```

UPDATE Transaction:

To modify one or more documents into a collection then the `updateOne()` and `UpdateMany()` methods can be used.

<code>db.collection.updateOne(document)</code>	UpdateOne() method can be used to modify a single document into a collection.
<code>db.collection.updateMany(document)</code>	UpdateMany() method can be used to modify a multiple documents into a collection.

For example:

```
db.customerdetails.updateOne(
{ cust_id: {$gt: 60},
{$set: {cname: "Falguni"}  })
```

DELETE Transaction:

To delete one or more documents from a collection then the `deleteOne()` and `deleteMany()` methods can be used.

<code>db.collection.deleteOne(document)</code>	<code>deleteOne()</code> method can be used to delete a single document from a collection.
<code>db.collection.deleteMany(document)</code>	<code>deleteMany()</code> method can be used to delete a multiple documents from a collection.

For example:

```
db.customerInfo.deleteOne(
{
  first_name: "Falguni"
})
```

TO ESTIMATE TIME EXAMINATION

I have performed test cases in order to evaluate the time utilization for Oracle NoSQL and MongoDB catalogue on 958, 4300 and 5000 datasets correspondingly.

To evaluate Time estimation for INSERT transaction.

TABLE: moment in time (ms) of inserting records

Qty of Records	MongoDB	Oracle NoSQL
958	120	142
4300	258	220
5000	227	255

The below figure:3 displays Column-chart pictorial representation of Insert operation on oracle NoSQL and MongoDB.

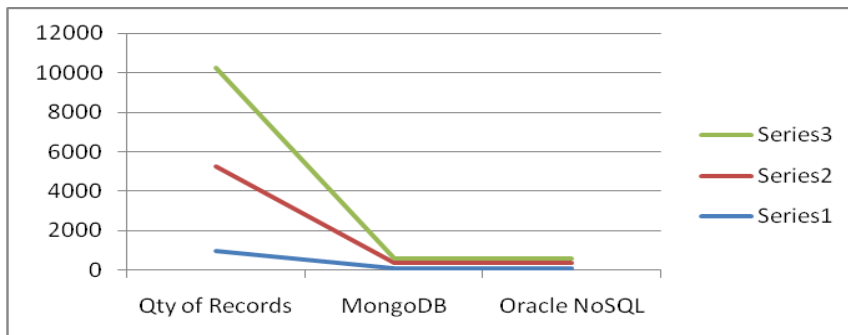


Figure: 2

2. Operations in Oracle NOSQL

Oracle NOSQL architecture has its own command line Interpreter (CIL) to run various queries.

CREATE Transaction:

<pre>Create table <tableName> (columnName1 datatype, columnName2 datatype...)</pre>	<p>To create a new table, create command can be used.</p>
---	---

For example:

```
create table employee
(
empid integer,
ename string
)
```

DELETE Transaction:

This operation can be performed by multidelete() method. Based on Primary Key, records can be deleted.

<pre>tableName.delete(primarykeycolumn, value)</pre>	<p>Delete command can be used to delete or remove records from the table. Here primary key has to be provided while you delete a row.</p>
--	---

For example:

```
EmployeeInfo.delete(empid, null, null)
```

UPDATE Transaction:

Oracle NoSQL update command is similar to traditional SQL update command but it manage rich data model. It directly updates the data at the server side.

Update <tableName> [<table-alias>] Set columnName=Expression Where Condition [<returning_clause>];	To update the rows into a table, Update command can be used.
---	--

For example:

```
update Employee
set E.ename="falguni"
where E.empid=2 Returning empid, E.ename;
```

Conclusion

The major implication is we will examine the performance helps in formative how the data should be stored and manage. Our focus on this paper is to evaluate the performance between a well-known document-oriented type of NoSQL databases i.e., MongoDB with the Microsoft SQL server as a relational Database.

We will also compares the basic concepts and expressions used by two databases, as well as, perform some tests for evaluating the speed of these two databases in terms of insert operation, simple and complex queries.

References

1. D. Pritchett, "Base: An acid alternative," *Queue* 6, no. 3 (2008): pp. 48-55.
2. Dadapeer, N. M.Indravasan, and G. Adarsh "A Survey on Security of NoSQL Databases". *International Journal of Innovative Research in Computer and Communication Engineering*. ISSN (Online): 2320-980, Vol. 4, Issue 4, April 2016.
3. A B M Moniruzzaman and S. A. Hossain. "NoSQL Database: New Era of Databases for Big data Analytics, Classification, Characteristics and Comparison". *International Journal of Database Theory and Application*, ISSN: 2005-4270, Vol. 6, No. 4, August, 2013.
4. F. Zhu, J. Liu, and L. Xu: "A Fast and High Throughput SQL Query System for Big Data". *International Conference on Web Information Systems Engineering (WISE)*, pp 783-788-2012.

5. S. Khan and Prof. Vanita Mane. "SQL Support over MongoDB using Metadata", *International Journal of Scientific and Research Publications*, Volume 3, Issue 10, October 2013.
6. R. Lawrence. "Integration and virtualization of relational SQL and NoSQL systems including MySQL and MongoDB". *International Conference in Computational Science and Computational Intelligence (CSCI)*, volume 1, pp 285-290, March 2014.
7. M. Hanine, A. Bendarag and O. Boutkhoul, "Data Migration Methodology from relational to NoSQL Databases", *International journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol: 9, No: 12, 2014
8. M. Potey, M. Digrase, G. Deshmukh, and M. Nerkar. "Database Migration from Structured Database to Non-Structured Database". *International Conference on Recent Trends & Advancements in Engineering Technology (ICRTAET) 2015*.
9. A. Abdullah and Q. Zhuge. "From Relational Databases to NoSQL Databases: Performance Evaluation". *Research Journal of Applied Sciences, Engineering and Technology*, Vol: 11 No: 4 pp: 434-439, 2015
10. W. L. Low, J. Lee, and P. Teoh., "Conceptual Mappings to Convert Relational into NoSQL Databases". *18th International Conference on Enterprise Information Systems*, January 2016.
11. A. Goyal, A. Swaminathan, R. Pande and V. Attar, "Crossplatform (RDBMS to NoSQL) database validation tool using bloom filter," *2016 International Conference on Recent Trends in Information Technology (ICRTIT)*, Chennai, 2016, pp. 1-5.
12. J. Kepner et al., "Associative array model of SQL, NoSQL, and NewSQL databases," *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, 2016, pp. 1-9.
13. P. Sareen and P. Kuma, "NoSQL Database and its Comparison with SQL Database". *International Journal of Computer Science & Communication Networks*, Vol: 5 No: 5, pp 293-298, 2011.