A Quantitative Performance Analysis between Mongodb and Oracle NoSQL

Sarita Padhy
Velocis System pvtLtd
Delhi, INDIA
Email Id: spadhy8@gmail.com

G Mayil Muthu Kumaran
Scientist-F
National Informatics Center, Delhi, INDIA
Email Id: muthu@nic.in

***Abstract*—NoSQL, referred to "Not only SQL", is trending as a great competitor to Relational Database with its capacity to handle unstructured and semi structured data better than traditional RDBMS. Relational Database management system stores data in tabular format in the form of tables i.e. rows and columns, Alternatively, NoSQL is a schema less database which stores data in key-value format. NoSQL databases are used to store unstructured and semi structured data such as unstructured data is users session, log data, messaging, videos, audios, images, which are growing rapidly than structured data, which doesn't fit in traditional relational databases. NoSQL database is primarily designed to provide more flexibility, speed and scalability. This paper focuses on two prominent NoSQL databases i.e. mongo DB and Oracle NoSQL and have compared the time analysis based on the insert, delete and update operations on data in each database for same datasets. The criteria for comparison includes basic data model, characteristics & database performance on same type of datasets.**

Keywords—NoSQL; Mongodb; Oracle NoSQL; Database Management System; RDBMS

I. INTRODUCTION

Database management system (DBMS) is a system software platform that stores and retrieves a collection of related data. DBMS system is generally used in areas of application specific data management, where performance, availability and scalability take highest priority. The traditional relational database stores the structured data in tabular format in a schema format. NoSQL is an alternative method of relational database, which stores the unstructured or semi structured data (e.g. images, videos, social media data etc.) in schema less format i.e. in the form of document type, key-value pair, graph based and column based. According to estimates, the volume of business data doubles in every 1.2 years. The increase of unstructured data collected by business is undeniable. NoSQL database can handle large number of unstructured or semi structured data, which was difficult to achieve in traditional relational database. NoSQL handles large volume of data, supports dynamic schemas, auto-Sharding, and horizontal scaling which is very difficult to manage in traditional database.

With this fast growing technology (points on MongoDB and oracle NoSQL), NoSQL brings the new revolution in the database sphere. Oracle NoSQL is an advanced key value model which stores the data in the form of JSON and which provides functionalities such as high reliable, flexible data model and high availability through automatic load balancing. Whereas, MongoDB is a document type model which stores the data in JSON/BSON format and provides functionalities such as high performance, high availability through replication and high scalability through Sharding technology.

II. NOSQL DATA MODELS

A. Document Model

Document model stores the data in document format in a structure like JSON (JavaScript Object Notation) which contains one or more fields, where each field contains a typed value, such as a string, date, binary, decimal value, array etc.

B. Key Value Model

Data is stored in a key- value pair for representing polymorphic and unstructured data and data can only be queried by the key.

C. Graph Model

Graph model uses nodes and edges to store and represent data respectively. In graph model nodes is storing the actual data and edges are representing to connect to different nodes. This model is useful for computing complex data.

III. TECHNICAL OVERVIEW

A. MongoDB

MongoDB [1] is the most widely used document type NoSQL database. MongoDB stores data in flexible, JSON-like document format, where fields can vary from document to document and data structure can be changed over time.

- MongoDB community server is free and open-source, published under Server side public side and Apache License 2.0
- The document model maps to the objects in application code, which makes data easy to work with.

978-93-80544-34-2$31.00 ©2019 IEEE

- It supports adhoc queries, indexing, and real time aggregation to access and analyse the data.
- It provides high availability, horizontal scaling, and geographic distribution for easy to use.

B. Oracle NoSQL

The Oracle NoSQL [2] Database, "No Single Point of Failure" architecture, is an advanced key value document data model where the key value is divided into two parts i.e. one is the Minor key, which stores the data locality and other one is the Major key which is used for distributing actual data.

- It supports Schema Evolutiontechnique which can be achieved by JSON API [3]. Schema evolution provides read and write access to the older version of data, even if the data has been modified to newer version.
- It provides Table Evaluation technique which can be achieved by Table API. Table API, which is very different from the relational table, actually stores the metadata, not the actual data.
- Table API allows the designing of complicated data structures such as nested tables, tables with arrays, tables with parent and child relationships.
- It provides highly reliable, flexible and high availability data management across a set of storage nodes.
- If the data format doesn't fit into key value pair, it provides environment to run SQL queries. The major difference between oracle RDBMS and oracle NoSQL in the context of SQL is, that Oracle NoSQL supports arrays and maps datatypes.
- It supports fixed schema, adhoc and schema less data structure.

IV. DIFFERENT IN SYNTAX

A. CRUD Operations in MongoDB

To run queries, mongoDB [4] uses its own Command Line Interpreter (CIL). A Table in mongoDB is called a collection. It supports insert, delete and update documents in a collection.

1) Insert Operation:

In MongoDB, Create operation creates new documents to a collection whereas Insert operation insets the new document to a collection and if the collection doesn't exist, then it creates a new collection. The insert operation inserts document into a single collection and all write operations are atomic on the level of a single document.

The insert Operation can be performed by two methods using insert() and save().

TABLE I. INSERT OPERATION IN MONGODB

SYNTAX	DESCRIPTION
db.collection.insert(document)	Insert document into a collection
db.collection.insertOne(document)	InsertOne() inserts a document into a collection
db.collection.insertMany(document)	InsertMany() inserts more than one document into a collection.

Ex:
```
db.customer.insertOne
({
 customer_id:101,
name: "Sammy"
})
```

2) Read Operation:

TABLE II. SELECT OPERATION IN MONGODB

SYNTAX	DESCRIPTION
db.collection.find(document)	Find() is used to query the documents from a collection based on certain criteria.

Ex:
```
db.customer.find
({
customer_id: {$gt: 105}
})
```

3) Update Operation:

TABLE III. UPDATE OPERATION IN MONGODB

SYNTAX	DESCRIPTION
db.collection.updateOne(document)	UpdateOne () updates a document into a collection
db.collection.updateMany(document)	UpdateMany() updates more than one document into a collection.

Ex:
```
db.customer.updateOne
        ({
        customer_id: {$gt: 105},
        {$set: { name:"John"}       })
```

4) Delete Operation:

TABLE IV. DELETE OPERATION IN MONGODB

SYNTAX	DESCRIPTION
db.collection.deleteOne(document)	deleteOne() deletes a document from a collection
db.collection.deleteMany(document)	deleteMany() deletes more than one document from a collection.

Syntax:

```
db.collection.deleteOne(document)
        db.collection.deleteMany(document)
```

Ex:
```
db.customer.deleteOne
        ({
        first_name: "Sammy"
        })
```

B. CRUD Operations in Oracle NoSQL

Oracle NoSQL also uses its own command line Interpreter (CIL) to run the queries which is present in KVStore. Oracle NoSQL supports the traditional SQL queries. [5]

1) Create Operation[6]:

TABLE V. CREATE OPERATION IN ORACLE NOSQL

SYNTAX	DESCRIPTION
Create table <tablename>ifnot exists (column1 datatype, column2 datatype…..)	Create table is used to create a new table.

Ex: Create table customer ifnot exists
(customer_id integer, column2 string)

2) Insert Operation[7]:

TABLE VI. INSERT OPERATION IN ORACLE NOSQL

SYNTAX	DESCRIPTION
put table -name <tablename> Tablename → add-value-field column_name -value column_name	To insert data into the table, put command is used.

Ex:
put table – name customer
customer →add –value -field customer_id -value 1
customer →add-value-field name -value John

3) Update Operation[8]:

Oracle NoSQL update operation eliminates *read-modify-write* cycle. In Read-modify-write cycle, it fetches the whole user data at the client end, computes and then sends it back to the server. Oracle NoSQL update the data at the server-side directly.

Update statement in Oracle NoSQL is similar to the standard SQL update statement but it handles rich data model.

TABLE VII. UPDATE OPERATION IN ORACLE NOSQL

SYNTAX	DESCRIPTION
UPDATE <table_name> [AS <table_alias>] <update_clause>[<update_clause>]* Where <expr> [<returning_clause>];	- Update command is used to update the rows in a table
Table evolve –name table_name Add-field –type datatype –name column_name	- Evolve command is used to add new column/field to an existing table.

Ex: update customer C
Set C.name = John
Where C.customer_id = 121
Returning customer_id, C.name;

Or

Table evolve –name customer
add –field –type string –name customer_name

4) Delete Operation[9]:

Delete operation can be done by multidelete() method in oracle NoSQL. Rows can be deleted based on the primary key.

TABLE VIII. DELETE OPERATION IN ORACLE NOSQL

SYNTAX	DESCRIPTION
table_name.delete(primarykey,value)	-To delete a row from the table, Delete command is used. For deleting a row, primary key has to be provided.
Table evolve –name table_name Remove –field –name column_name	To delete a field in a table, evolve command is used.

Ex:
customer.delete(customer_id, null, null)

Or

table evolve –name table_name
remove –field –name customer_name

V. TIME ANALYSIS

Test cases were performed in order to estimate and compare the time consumption for Oracle NoSql and Mongo Database on 1000, 5000, 10000, 50000, 100000 datasets respectively. Same datasets were used to perform the operations in a single instance and not in a cluster.

A. Time evaluation for Insert operation:

The below table and graph shows the performance of insert operation on mongoDB and oracle NoSQL Databases. It can be inferred from TABLE IX that mongoDB is found to perform slightly better than oracle NoSQL with increase in size of data.

TABLE IX. INSERTING TIME(MS)

No. of records	MongoDB	Oracle NoSQL
1,000	126	150
5,000	300	256
10,000	426	359
50,000	2247	1900
1,00,000	4966	5286

The below fig.1 shows the graphical representation of insertion operation on mongoDB and oracle NoSQL.

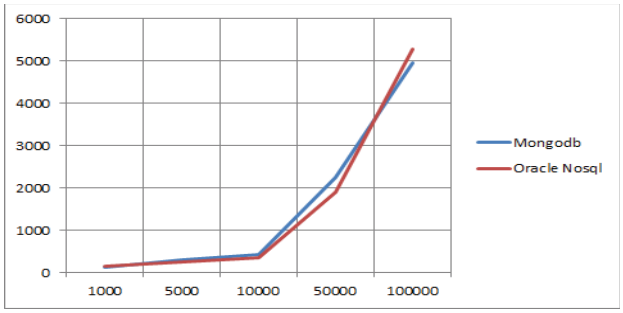

Fig. 1. Time analysis for Insert Operation

Next part is to show the performance of update operation on mongoDB and oracle NoSQL.

B. *Time analysis for Update operation:*

The below table and graph show the performance of update operation on mongoDB and oracle NoSQL Databases. It can be inferred from TABLE X that mongoDBis found to perform slightly better than oracle NoSQL while performing update operation.

TABLE X. UPDATING TIME(MS)

No. of records	MongoDB	Oracle NoSQL
1,000	17	19
5,000	152	178
10,000	565	750
50,000	1862	1926
1,00,000	2875	2945

The below fig.2 shows the graphical representation of update operation on mongoDB and oracle NoSQL.

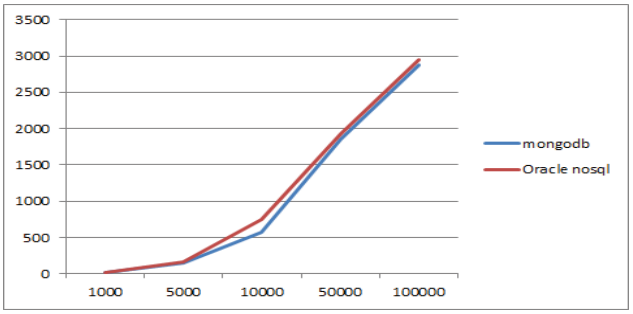

Fig. 2. Time analysis for Update Operation

C. *Time analysis for Delete operation:*

The below table and graph show the performance of delete operation on mongoDB and oracle NoSQL databases. It can be inferred from TABLE XI that mongoDBis found to perform slightly better than oracle NoSQL with increase in size of data.

TABLE XI. DELETING TIME(MS)

No. of records	MongoDB	Oracle NoSQL
1,000	10	15
5,000	65	60
10,000	135	125
50,000	579	628
1,00,000	1458	1859

The below fig:3 shows the graphical representation of delete operation on mongoDB and oracle NoSQL.

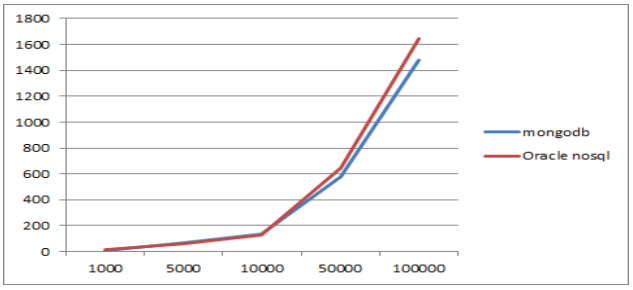

Fig. 3. Time analysis for Delete Operation

VI. MONGO DB VS ORACLE NOSQL

A. *PROS*

TABLE XII. PROS OF MONGODB AND ORACLE NOSQL.

Sno	MongoDB	Oracle NoSQL
1	It is a document-oriented NoSQL database which provides functionalities such as high performance, high availability through replication and high scalability through Sharding technology.	It is a distributed key-value pair database which provides high reliable, flexible data model and high availability through automatic load balancing. [10] [11]
2	It is very flexible due to which addition/deletion of data has very less or no impact on the application.	It provides powerful features like schema evolution and table API.
3	It supports adhoc queries. [12]	It provides simple programming language integration.
4	It supports rich query language, data aggregation, and text-search and geospatial queries. [13]	It supports full text-search, Time-To-Live, SQL query and geospatial queries. [14]
5	MongoDB Community edition is licensed under server side public side and Apache License 2.0	Oracle NoSQL Community edition is licensed under Apache License 2.0
6	MongoDB is developed by MongoDB Inc.	Oracle NoSQL is developed by Oracle corporation.
7	MongoDB [15] provides driver for C, C++, C#, java, node.js, perl, php, python, Motor, Ruby, Mongoid, and Scala.	Oracle NoSQL provides driver for C,C#, java, javascript, python.

B. *CONS*

TABLE XIII. CONS OF MONGODB AND ORACLE NOSQL.

Sno	MongoDB	Oracle NoSQL
1	It doesn't support join operation. The document size is limited to 16MB. [16]	The maximum table size is 5TB.
2	The index key limit is must be less than 1024 bytes. Nesting of documents for more than 100 levels is not possible.	The index size is limited to 64 bytes.

VII. LIMITATION

NoSQL provides a lot of advantages over relational databases, yet Data security is a major challenge in both traditional & NoSQL databases. NoSQL supports data security, but still security is a limiting factor for NoSQL deployment. NoSQL doesn't perform on ACID transactions, which ensures data consistency in the database. But NoSQL follows "eventual consistency" that means data in one database node may go out of sync with data on another node. The learning curve of NoSQL database is steeper.

The goal for NoSQL database is designed to provide a solution which doesn't require any administration. But in reality, NoSQL databases needs to be administrated by technical skilled persons. NoSQL databases are still new, so

every NoSQL developer is still learning the skills. Over time this situation will be resolve.

VIII. FUTURE SCOPE

This paper focuses on the time analysis based on the insert, delete and update operations of two NoSQL databases i.e.mongo DB and Oracle NoSQL in a single node. In future, performance analysis between MongoDB and oracle NoSQL can be performed in a cluster node.

IX. CONCLUSION

This paper delivers quantitative performance analysis between mongoDB and oracle NoSQL. This paper can assist the readers to choose the best NoSQL option between mongoDB and Oracle NoSQL based on their requirements.

Oracle NoSQL provides better distribution model. It is best for implementing storage nodes designed to provide flexibility, availability, then implementing on storage and distribution model. MongoDBon the other hand, relies on sharding for Scale in and scaling out, further it involves designating a specific server to hold certain chunks of the data when the data increases continuously.

REFERENCES

[1] Retrieved from https://docs.oracle.com/en/database/other-databases/nosql-database/18.1/tutorials.html

[2] Retrieved from https://docs.oracle.com/database/nosql-12.1.3.0/GettingStartedGuideTables/tablesapi.html

[3] Retrieved from https://docs.oracle.com/database/nosql-12.1.3.0/GettingStartedGuideTables/tablesapi.html

[4] Retrieved from https://www.oracle.com/webfolder/technetwork/tutorials/obe/nosql/bulkput/index.html#section2s1

[5] Retrieved from https://docs.oracle.com/cd/NOSQL/html/SQLForNoSQL/updates.html

[6] Retrieved from https://docs.oracle.com/database/nosql-12.1.4.0/GettingStartedGuideTables/recorddelete.html

[7] Retrieved from https://blogs.oracle.com/oracleuniversity/explore-features-benefits-of-the-oracle-nosql-database-take-oracle-university-training

[8] Retrieved from https://db-engines.com/en/system/MongoDB%3BOracle+NoSQL

[9] Retrieved from https://en.wikipedia.org/wiki/MongoDB

[10] Retrieved from https://en.wikipedia.org/wiki/Oracle_NoSQL_Database

[11] Retrieved from https://en.wikipedia.org/wiki/Oracle_NoSQL_Database

[12] Retrieved from https://docs.mongodb.com/ecosystem/drivers/

[13] Retrieved from https://docs.mongodb.com/manual/reference/limits/

[14] Retrieved from https://docs.mongodb.com/manual/crud/

[15] Retrieved from https://docs.oracle.com/cd/NOSQL/index.html

[16] Alexandru Boicea,Florin Radulescu, Laura Ioana Agapin "MongoDB vs Oracle - databasecomparison",IEEE

[17] V.Anand by "MongoDB and Oracle NoSQL: A technical critique for design decisions",IEEE

[18] R. Poljak, P. Poš_i_ and D. Jakši by "Comparative Analysis of the Selected RelationalDatabase Management Systems",IEEE