

RDBMS IA-1

Topic : A Quantitative Performance Analysis between MongoDB and Oracle NoSQL

Created By:
Devansh Shah
1911052

What is a Database?

- A **database** is an organized collection of data.
- The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.
- **Dynamic websites** on the World Wide Web nowadays which are handled through databases.
- There are many **databases available** like MySQL, Sybase, Oracle, MongoDB, Informix, PostgreSQL, SQL Server, etc.



Types Of Database

There are different types of database :

- SQL Database (Relational Database)
- NoSQL Database



SQL Database:

- SQL Database are also called as relational database.
- The data is stored in the form of tables.
- Uses SQL for retrieving data from the database.
- A relational database contains the following components:
 - Table
 - Record/ Tuple
 - Column name /Attribute
 - Schema
 - Keys



	emp_id	last_name	first_name	hire_date	job_title
1	1	Smith	James	2016-03-01 00:00:00.000	Staff Accountant
2	2	Williams	Roberta	2004-02-07 00:00:00.000	Sr. Software Engineer
3	3	Weinberg	Jeff	2007-01-02 00:00:00.000	Human Resource Manger
4	4	Franklin	Victoria	2010-07-02 00:00:00.000	Operations Manager

NoSQL Database:

- NoSQL database are often called as “Not Only SQL” database.
- The data is stored in the form of documents.
- Alternative to traditional relational database.
- Have dynamic Schema.

```
{  
  "_id": "5cf0029caff5056591b0ce7d",  
  "firstname": "Jane",  
  "lastname": "Wu",  
  "address": {  
    "street": "1 Circle Rd",  
    "city": "Los Angeles",  
    "state": "CA",  
    "zip": "90404"  
  },  
  "hobbies": ["surfing", "coding"]  
}
```

MongoDB Database:

- MongoDB 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.
- Uses Document-Model.
- Ad hoc queries, real time aggregation provide powerful ways to access and analyze your data.
- MongoDB provides `db.collection.estimatedDocumentCount()`



CRUD Operations:

1] Create :

- Insert operation inserts the new document to the collection.
- Methods - insert() , insertOne() , insertmany([]).

TABLE I. INSERT OPERATION IN MONGODB

SYNTAX	DESCRIPTION
<code>db.collection.insert(document)</code>	Insert document into a collection

<code>db.collection.insertOne(document)</code>	InsertOne() inserts a document into a collection
<code>db.collection.insertMany(document)</code>	InsertMany() inserts more than one document into a collection.

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

2] Read:

- Find function reads the data from the database.
- Methods - find() , findOne().

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:

- Update function is used to update the data in the database.
- Methods - `updateOne()` , `updateMany()` .

TABLE III. UPDATE OPERATION IN MONGODB

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

Ex:

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

4] Delete:

- Delete function is used to delete the data from the database.
- Methods - `deleteOne()` , `deleteMany()` .

4) Delete Operation:

TABLE IV. DELETE OPERATION IN MONGODB

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

Syntax:

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

Ex:

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

Oracle NoSQL Database:

- As a true multi-model data store, the Oracle NoSQL Database provides several different options for data modeling:
- **Tables** – Using this modeling option, developers specify table definitions using a SQL data definition language, very similar to the relational database. Command line SQL for querying the Oracle NoSQL Database or the Oracle NoSQL Database APIs can be used for table data access.
- **JSON documents** – Using this modeling option, developers can store JSON objects in the Oracle NoSQL Database and use SQL to access these objects.



CRUD Operations:

1) Create:

- Create command is used for creating the tables for the database.

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:

- Put command is used for inserting the data into the table.

2) Insert Operation[7]:

TABLE VI. INSERT OPERATION IN ORACLE NoSQL

SYNTAX	DESCRIPTION
<pre>put table -name <tablename> Tablename → add-value-field column_name -value column_name</pre>	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:

- Update statement in Oracle NoSQL is similar to the standard SQL update statement.

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;

4] Delete:

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

TABLE VIII. DELETE OPERATION IN ORACLE NoSQL

SYNTAX	DESCRIPTION
<code>table_name.delete(primarykey,value)</code>	-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)
```

Time Analysis:

➤ Time Analysis for Insert operation:

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.

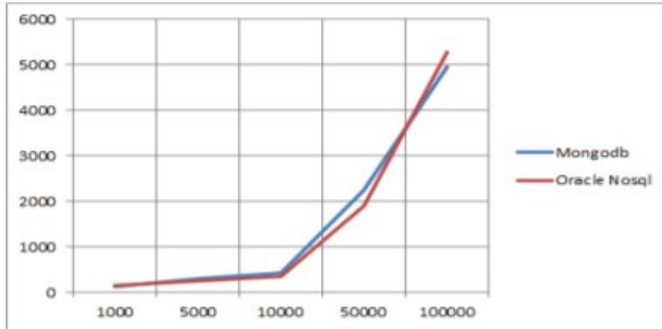


Fig. 1. Time analysis for Insert Operation

➤ Time Analysis for 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.

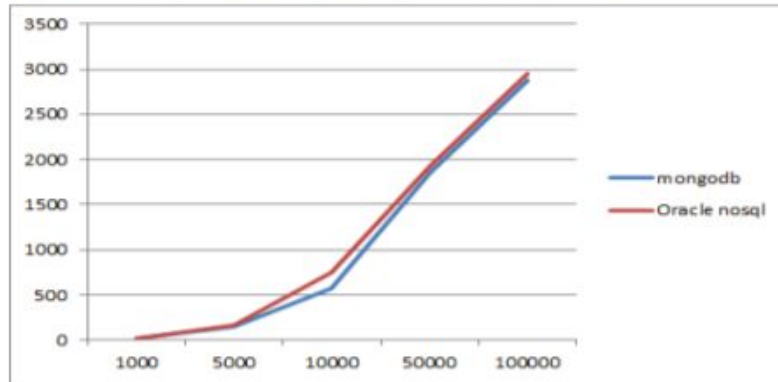


Fig. 2. Time analysis for Update Operation

➤ Time Analysis for Delete operation:

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.

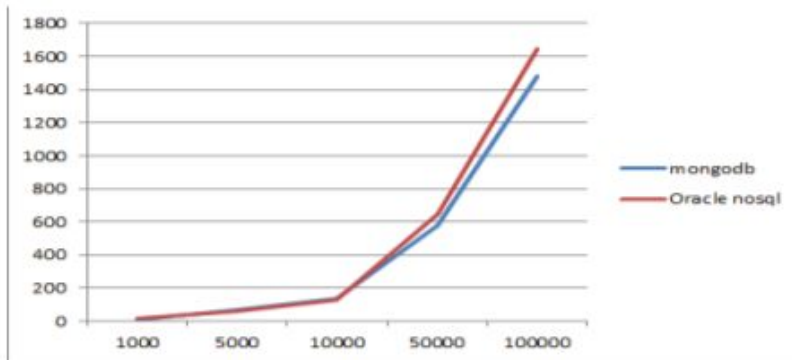


Fig. 3. Time analysis for Delete Operation

Properties:

MongoDB :

- Existence of the collection is not necessary for the creation of document
- It is very flexible due to which addition/deletion of data has very less or no impact on the application
- Uses BSON to store data.
- MongoDB is developed by MongoDB Inc.

Oracle NoSQL :

- The table definition has to be created first, after which you can continue to create the rows
- It provides powerful features like table API.
- Uses JSON to store data.
- Oracle NoSQL is developed by Oracle corporation

Conclusion:

- Using MongoDB, applications built can be faster
- Processing of data is faster of MongoDB as compared to Oracle NoSQL
- Oracle NoSQL wins when it comes to tools it has to offer and possible options to create data models.



Implementation Strategy:

- Perform CRUD operations on both of the databases.



References:

- <https://docs.mongodb.com/manual/crud/>
- <https://www.oracle.com/technetwork/database/nosqldb/learnmore/nosql-database-498041.pdf>
- <https://www.mongodb.com/nosql-explained/nosql-vs-sql>
- <https://severalnines.com/database-blog/battle-nosql-databases-comparing-mongodb-and-oracle-nosql>



END :

THANK YOU !!

