ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

Study on Handling Semi-structured Data using NoSQL Database

Sapthami R, Anisha P Rodrigues
PG Scholar, Dept of CSE, NMAMIT, Nitte
Assistant Professor, Dept of CSE, NMAMIT, Nitte

Abstract*— Huge Volume and Variety of data Such as unstructured and semi structured data is being produced by many web 2.0 sources such as Facebook, Google, yahoo!. Handling this variety of data that don't have consistent format is a big challenge. Relational databases are used to handle limited amount of structured data, but web 2.0 sources produce huge volume of unstructured and semi structured data which cannot be handled by traditional relational databases. This is where NoSQL databases came into picture. NoSQL database is used to store and retrieve huge volume of semi structured and unstructured data more efficiently. In this paper we present how to handle unstructured and semi structured data using one of the NoSQL database, MongoDB.

***Index Terms*— MongoDB, NoSQL database, semi-structured data.**

I. INTRODUCTION

Web 2.0 technologies like Facebook, Twitter, Google and Yahoo! etc, produce and at the same time deal with huge volume of unstructured and semi structured data. This varieties of data cannot be handled by relational databases. Relational databases are used to store and retrieve limited volume of structured data. Even though relational databases are widely used and provide good performance when they deal with structured data having fixed schema, but they are not efficient to handle unstructured and semi structured data. In order to handle semi structured and unstructured data we need more than relational database. To deal with this problem NoSQL databases are introduced.

NoSQL database is non relational database stands for Not Only SQL. NoSQL was first introduced by Carlo Strozzi in 1998 and refers to non-relational database, and then the term was later reintroduced by Eric Evans in 2009. NoSQL databases are used to overcome the problem of storing and retrieving unstructured and semi structured data. Because the data produced by different web applications are not always structured data. NoSQL databases have very simple data model and are schema free i.e the data is not stored using fixed table schemas as in relational databases. NoSQL databases do not support ACID properties for transactions as in relational database but still provide high performance and scalability.

There are different data models of NoSQL databases they are as follows:

A. Key value database

Here the data is stored in form of key value pair i.e it is organised as an associative array. A unique key is used to retrieve the data. Less time is used to execute the query since the data retrieval is based on unique key. Here the data model is schema free because the values stored here are independent with no limit on length which is suitable for handling semi structured and unstructured data. At runtime new values can be added without affecting already existing values in the database. Voldemort, Scalaris, Raik and Redis are examples for key value database.

B. Column oriented database

Google's big data and Apache HBase, part of Hadoop are both column oriented databases. Due to success of Google's Big Table, the column oriented databases are motivated. In column oriented, column values are stored contiguously where as in relational database row values are stored contiguously. Column oriented storage for some operations like aggregation, support for ad-hoc and dynamic query provide better performance when compared to relational database. here columns are grouped together to form column family, this column family must be pre defined but new values can be added without effecting existing one, which make it flexible. HBase, Hypertable and Cassandra are examples for column oriented database.

C. Document oriented database

Here the data model consists of collections and document. the data is stored in the form of documents which in turn consist of a unique key and corresponding values. A collection contains set of document and database contains set of collection. These documents have different fields and these fields can be added at any time. The documents can be in any format like JSON (Java Script Object Notation), XML (extensible Markup Language) etc. In relational databases the every record stored in tables have same fields where in document oriented database every records need not be same. In key value database, the data is retrieved based on key, here we cannot get the specified field as we can do in document oriented databases. In document oriented database we can retrieve entire document or we can get the values from specified field. MongoDB, CouchDB and SimpleDB are examples for Document oriented database.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

D. Graph database

Here the data model consists of networked structure which contains nodes and edges. Edges are used to represent relationship between nodes. The set of key value pair are nodes. Here schemas are not pre defined, this makes graph database more flexible. Joins are not required in graph database. edges are traversed from one node to another to draw the relationship between them. In social networking websites, the graph database are made use where relationship among data is important. Neo4j, GraphDB and Titan etc are example for graph database.

II. LITERATURE SURVEY

The author in [1] compared two leading database i.e SQL database and NoSQL database. In this paper comparison made between two databases based on their features and also discussed tools that are used for relational and non relational database. The authors have pointed that SQL database is beneficial when reliability, flexibility, robustness and scalability is taken into consideration. But when it comes to web application that produces huge volume of unstructured and semi structured data, SQL cannot handle it. To overcome this NoSQL databases are made use.

Hecht and Jablonski [2] discussed about the various classes of NoSQL databases. These papers compare their data models, query possibilities, concurrency control, and partitioning and replication opportunities. Which results in choosing right class of NoSQL database depending on the application the developer make use?

In [3] the authors have compared relational database and NoSQL database classes based on data modelling and query syntax. This experiment is explained using a case study of news website like Slashdot. They have mentioned detailed description about key-value store, column-oriented store, document-oriented store and graph store. Here for comparison they have taken MongoDB from document oriented database, Neo4j from graph database of NoSQL database. And in SQL, PostgreSQL is used. In SQL ER diagram is used for modelling the data but in NoSQL no specific data modeling technique is available.

In [4] author discussed about NoSQL database and relational database. And evaluated new system based on their data model, consistency and storage mechanism, availability and query support. This paper provided detailed description about different classes of NoSQL. And also examples of each classes are discussed.

In [5] the author has tried to present comparative study on NoSQL and SQL database. For the comparison the author have chosen MongoDB from NoSQL database and MySQL from SQL database. They have justified why MongoDB is better than MySQL database and also highlighted advantages of using NoSQL database when compared to SQL database based on various operation performed on two database.

In [6] the author discussed the need for database that is able to store and process big data efficiently. Mainly in large applications like search engines and SNS, using relational database is inadequent to store and query dynamic user data. So NoSQL databases are created. The paper discuss about the features of NoSQL database and data model. And also discuss about classification of these data model based on the CAP theorem

III. MONGODB

MongoDB is one of the document oriented database not a relational database. Where data is stored in form of documents. This type of database contains collections and documents. Each collection contains set of document and each database contains set of collections. There are no concepts of schema and tables. It doesn't have transaction, ACID compliance, joins, foreign keys. Since there is no fixed schema removing and adding of fields in documents are easier in MongoDB. MongoDB is highly scalable, highly available and provide high performance. MongoDB stores the data with dynamic schemas in BSON format written in C++. There are MongoDB driver for other languages like Java, Ruby, Python, C, C++, C# etc.

In MongoDB collections and documents are thought of tables and rows in relational database.

Table 1: RDBMS v/s MONGODB Terms

RDBMS	MONGODB
Database	Database
Table	Collection
Rows	BSON Document
Primary key	Primary key
Index	Index
Join	Embedded documents
Column	Fields

The figure 1 shows the collection in the database created within MongoDB.

```
> db.student.find();
{ "_id" : ObjectId("56dbdbf6857dd9bald6ad544"), "name" : "abc", "branch" : "CSE"
, "age" : 20 }
{ "_id" : ObjectId("56dbdc9d857dd9bald6ad545"), "name" : "abc1", "branch" : "CSE
", "age" : 21, "mobile no" : 8967425841 }
```

Fig 1: collection in Mongodb

The above figure shows the set of documents inside a collection of MongoDB. In above figure the unique ID with corresponding value is present. An unique id is a primary key, which helps to retrieve values easily. Here unique ID is generated by MongoDB to each and every document, if ID is not given by the programmer. In relational database the primary key has to be manually mentioned to each and every record by the programmer. Or else primary key contain will occur which indicates primary key field not null. So we have

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

to include primary key for every record in relational database. We can observe that the fields within the documents are not same. In first document there are fields like unique ID, name, branch and age. In second document an extra field named mobile number is added. This indicates that the fields can be added dynamically according to the availability of information. Which in turn says that there is no concept of pre-defined schema. But in relational databases such as MySQL database, Oracle database, MS SQL etc, the number of fields in each record should be same. Since traditional relational databases follows pre-defined schema structure. There is no constrain on data stored in MongoDB collections as in relational databases.

A. Data Modelling and Quering in MongoDB

Some document oriented database such as CouchDB and RavenDB store data in form of JSON format, but MongoDB stores the data based on BSON format where BSON is Binary JSON that will enable binary serialization. It is easy to map the object structures if it is stored in JSON or BSON format. Boolean, float, integer, date, binary types and strings are supported by BSON.

The basic CRUD operation of MongoDB that are used to create, read, update and delete the data within the MongoDB collections. Create operation, used to create database. Use command is used to create the database.

The above, will create the new database, if the database dose not exit, or else it will return existing database. To see the database we have to insert document into MongoDB collection. For example, If the database is college and students are collections. And documents are student information such as name of student, age, phone number and address.

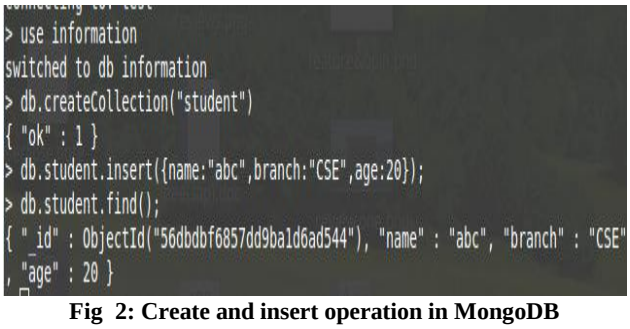

Fig 2: Create and insert operation in MongoDB

The figure 2 shows the list of students details within the collection "students". In RDBMS the command to create the database is,

CREATE TABLE student (
Usn_id INT NOT NULL,
name VARCHAR(50),
branch VARCHAR(50),
age INT);

Above is the schema before inserting values into the relational database such as MySQL, Oracle database etc.

Insert values into student ("123", "abc","CSE", 20);

If at all we are inserting student details in one of the relational database such as MySQL. Than after creating the database and table, we can't just insert the values, we have to first define the structure of the table and then insert the values. once the schema is defined no extra fields or records should be inserted.

Whereas it's not the case in mongodb, if we insert 5 fields in one document, than we can add extra field say 6th field. In the above example if we want to add branch field for the student than we can add i.e we are not strict to a schema. Because there no constrain on data or no fixed schema in mongodb.

Read operation is used to read the data object. Here find () method is used to read operation. There are many ways to read the data, find () method is primary way of reading.

db.users.find({name:" abc"});

Here in the above query, the find() method will retrieve the document related to the key value pair. In relational database select operation is used to read the data. Same query cane be written in RDBMS as,

*SELECT **
FROM student
WHERE name="abc";

Update operation is used to update the data in documents. Here MongoDB database use update() method to update. Update method will accept 2 parameters, the query and the update.

db.users.update({name:"abc"},{age:20});

In RDBMS, the following query is written as

UPDATE student
SET subject age='abc'
WHERE name='abc'

Here in the above query, the update method will have all values in document and just change the value age.

Delete operation, which delete the document within the collection. Remove () method is used.

db.users.remove({name:"abc"});

In RDBMS, the delete operation is written as,

DELETE FROM student
WHERE name='abc';

There is difference in inserting the values into mongodb and relational database. The below figure shows the time taken to insert 10, 100, 1000, 10000, 100000, 1000000 documents into MongoDB.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 10, April 2016

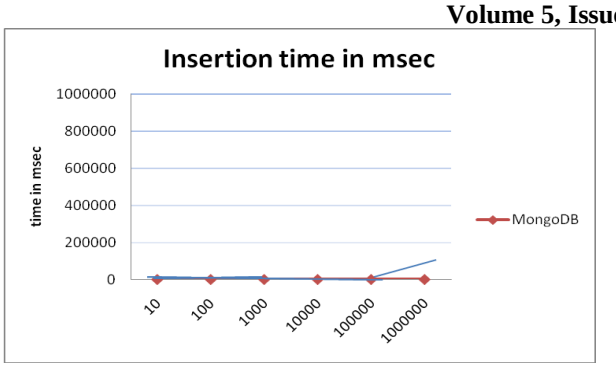

Fig 1: insertion time (msec) in MongoDB

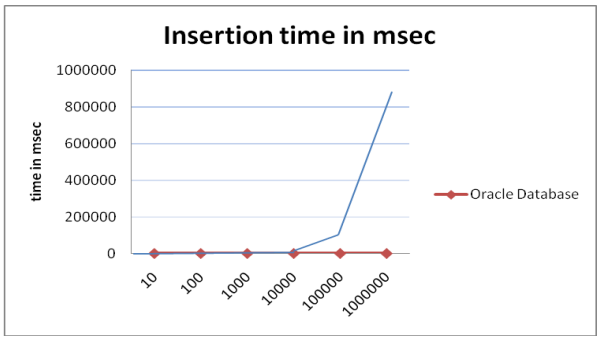

Fig 2: insertion time (msec) in Oracle Database

The figure shows the inserting time in oracle database, here we can see that time taken to insert large amount of data is bigger. Relational database works well with small data. In mongodb time taken to insert large amount of data is small, this grantees that mongodb is more effecient than relational database.

IV. CONCLUSION

In this paper, NoSQL data models like key value, document oriented, column oriented and graph database are discussed. And main focus is done on one of the document oriented database, MongoDB. Discussed the data model and querying process in both relational database and document oriented NoSQL database.

REFERENCES

[1] Nishthe Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, Dishant Gosian: "An Survey and Comparison of Relational and Non-Relational Database" International Journal of Engineering Research & Technology, pp.1-6, 2015.

[2] Robin Hecht, Stefan Jablonski : "NoSQL Evaluation A Use Case Oriented Survey". In: International Conference on cloud and service computing, pp.336-341, 2011.

[3] Karamjit Kaur, Rinkle Rani, "Modeling and Querying Data in NoSQL Databases". In: IEEE International Conference on Big Data, pp. 1-7, 2013.

[4] Cornelia Gy_rodi, Robert Gy_rodi, George Pecherle, Andrada Olah, " A Comparative Study : MongoDB vs MySQL", In Proceeding of International Conference on Engineering of Modern Electric Systems, pp.1-6, June 2015.

[5] Rick Cattell : "Scalable SQL and NoSQL Data Stores". ACM SIGMOD Rec. 39(4), pp.12-27 , 2010.

[6] Neal Leavitt : "Will NoSQL database live upto their promise?",Computer 43.2, pp.12-14, 2010

[7] Jing Han, Haihong E, Guan Le : "Survey on NoSQL Database",Pervasive computing and applications(ICPCA), 2011.