

Using the Advantages of NOSQL: A Case Study on MongoDB

Divya Chauhan

Department of Computer Science,
Himachal Pradesh University,
Summerhill, Shimla, India
dvcherish90@gmail.com

K. L. Bansal

Department of Computer Science
Himachal Pradesh University
Summerhill, Shimla, India
kishorilalbansal@yahoo.co.in

Abstract— With such a big volume of data growing tremendously every day, the storage of information, support and maintenance have become difficult. A further level of difficulty is added by the variety of data being captured. The data stored and updated on daily bases is in the form of logs, audio, video, sensor data and so on, which is not easy to be stored and queried using relational database. The paper gives the overview of NOSQL databases which provide more scalability and efficiency in storage and access of the data. A case study on MongoDB is done as to show the representational format and querying process of NOSQL database. The concepts of MongoDB are compared to the relational databases.

Keywords- Big data, NOSQL, MongoDB, JSON

I. INTRODUCTION

Few years back, relational databases were used widely for the application which had structured data to store and frequent reads but limited write. But with the advent of time the data is growing at such an exponential rate that it is believed the 90% of the total data is generated only in last two years. The types of data to be stored and high read and write rate has demanded a more scalable solution. Relational databases were not able to provide the horizontal scalability which is a big requirement of today's scenario. Horizontal scaling refers to the ability to distribute the data and read/write operation amongst the servers. This led to the introduction of term NOSQL databases to the technical world. The term NOSQL was first used by Carlo Strozzi in 1998 for his RDBMS, Strozzi NOSQL. NOSQL refers to Not Only SQL. It uses query language to access data but in a simpler way and gives addition features like it does not support a fixed schema and supports a high number of reads and writes operation per second without using joins to relate data.

NOSQL is now used in many fields like social networks, search engines, geospatial analysis, molecular modeling, data warehousing and caching. It does not provide ACID guarantees but follows BASE. Base stands for Basically Available, Soft state and Eventually consistent.

The main objective of this paper is show how the data is represented and queried in MongoDB. The paper is organized as follows: section 2 describes few related works. section 3 contains the classification of NOSQL databases. Section 4 gives the overview of MongoDB and its features. Also compares it to relational database. Section 5 gives the JSON representation of collection. Last section concludes the paper.

II. RELATED WORK

In recent years, after realizing the importance of NOSQL databases, significant work is done in this area. R. Arora and R.R. Aggarwal [1] proposed an algorithm to transform relational databases (MySQL) to NOSQL databases (MongoDB). The work can be extended for other NOSQL databases in future. The proposed algorithm was implemented in NetBeans Java IDE. B. R. M. Rao, A. Govardhan [2] proposed the algorithm to improve the performance of online aggregation. It was named as sharding parallel mapreduce for online aggregation. It obtained approximate result in less time as compared to the traditional mapreduce paradigm. A method was proposed to integrate two types of databases, namely MySQL and MongoDB, by adding a middleware between application layer and database layer [3]. The middleware consists of metadata which consists of different types of packages. Three popular NoSQL databases were compared, namely Cassandra, MongoDb, CouchDB [5]. And it was concluded that each solution was developed for different applications and had their own pros and cons. Prudence Kadebu, Innocent Mapanga[6] made following recommendations for security issues of NOSQL databases: making use of third party open source tools for audit and logging, built-in authentication, input validation, access control, Segregation of duties and encryption so that it can satisfy Firesmith's security requirements [7].

III. CLASSIFICATION OF NOSQL

NOSQL solutions are divided into four categories [8]. Key-value data stores, Column- oriented data stores, document data stores and graph data stores. Each category has its various tools to work with.

A. Key-Value data store

It consists of rows and column like relational database, but has got only two columns: key and value. Each key is unique and is used to retrieve the values associated with it. The stores are schema less which needs no fixed data model. These data stores are designed to handle massive data and the query speed is faster than the relational databases. The popular key value stores are Raik, Dynamo, Voldemort, Redis, Tokyo Cabinet.

B. Column Oriented data store

They are also similar to key value data stores except the data is stored in columns. The column of related data is stored in same file which is called column family. The column family contains the row key which consists of super columns. Super column is a column which contains other columns but not the super column. And a column is pair of name and value. Cassandra, Google's Big Table, Hyper Table and HBase are famous column oriented data stores.

C. Document data stores

In document database the data is stored in the form of document with the format of JSON, BSON or XML. The document is stored in the collections. It is schema less so any number of fields can be added to the database without keeping empty spaces for other document in the collection. There is no need of JOINS as in RDBMS due to embedded documents and arrays. The popular document databases are MongoDB, CouchDB, Terrastore, ThruDb.

D. Graph data store

Graph data bases use graph structures like nodes, edges to store and represent the data. It is index free adjacent store system. Every node is connected to the adjacent node and no look up is necessary. They are more suitable to applications with many inter connections like social networks. OrientDB, HyperGraphDB, Neo4j, AllegroGraph are popular databases of this category.

IV. MONGODB

MongoDB is an open source NOSQL database which falls under the classification of document database. It was initiated by 10gen Company. It was written in C++. MongoDB documents are stored in binary form of JSON called BSON format. BSON supports string, integer, date, Boolean, float and binary types. MongoDB is schema less so it gives the freedom to user for inserting new fields to the document or updating the previous structure of document. It does not use joins like relational databases as it has embedded documents which can be accessed quickly. MongoDB also support Master Slave replication where slave nodes contains the replicas of master nodes and are used for backups and reads.

MongoDB cluster is built up of three components.

Shard node: it contains one or more shards. Each master shard has its one or more slave shards. The slave shards contains the

copies of actual data which can be used at the time of failures. The read/ write operation is done by using the data present at the master node.

Configuration Server: A group of servers in MongoDB are called the configuration servers. The role of configuration server is to store the routing information and send the request to the right shard.

Mongos: Mongos are stateless and can be run in parallel [8]. It is responsible for serving the task requests from the clients. The request can need multiple shards. So mongos collects the data from different shards and merge them nicely before handing it over to the client.

Some features of MongoDB are [9]:

- **Flexibility**: MongoDB stores data in document format using JSON. It is a schema less document and maps to native programming language types.
- **Rich query language**: It gives the feature of RDBMS what we are used to with additional features of its own. Dynamic queries, sorting, secondary indexes, rich updates, easy aggregation, upsert (update if document exists and insert if it does not) are few RDBMS features and flexibility and scalability are the additional ones.
- **Sharding**: Autosharding allow us to scale our cluster linearly by adding more machines. It is possible to increase the efficiency which is very important on the web when load can increase suddenly and bring down the website.
- **Ease of use**: It is very easy to install, use, maintain and configure.
- **High performance**: It provides high performance data persistence. It reduces I/O activity on database system by supporting embedded documents. Use of indexing supports faster queries.
- **High availability**: MongoDB support replication facility called, replica set. Replica set is a group of servers that maintains same dataset. It provides automatic failover, redundancy and increased data availability.
- **Support for multiple storage engines**: It supports multiple storage engines such as WiredTiger storage engine, MMAPv1 storage engine. It also supports pluggable storage engine API that allows third party to develop storage engine for MongoDB.

The table Table1 below shows the terms and the concepts of two different types of database relational database such as MySQL and NOSQL database, here MongoDB.

TABLE I. TERMS AND CONCEPTS OF DIFFERENT DATABASES

Relational Database	MongoDB database
Fixed Schema	Schema less
Table	Collection
Rows, columns	Documents
Joins	Embedded documents
Vertical scaling	Vertical scaling, horizontal scaling

Table 2 compares the two different databases and shows few basic database queries which are used in manipulating the data stored in database.

TABLE II. BASIC QUERIES USED IN TWO DIFFERENT DATABASES

Query	Relational Database	MongoDB database
Create Command	CREATE TABLE table_name (column_name1 datatype, column_name2 datatype)	No need for defining schema
Insert Command	INSERT INTO table_name (column_name1, column_name2) VALUES (value1,value2)	db.collection_name. insert ({ name1: value1, name2: value2 })
Delete Command	DELETE FROM table_name WHERE (condition)	db.collection_name.remove ({condition})
Import command	BULK INSERT table_name FROM file_name WITH { FIELDTERMINATOR R=',', ROWTERMINATOR ='\\n' } GO	mongoimport --db database_name -- collection collection_name --type csv --file "file_name"
Select Command	SELECT column_name FROM table_name	db.collection_name({, {condition})

V. REPRESENTATION OF DATA IN JSON FORMAT

JSON stands for JavaScript Object Notation. It is light weight data interchange format. It is language independent. Moreover it is self descriptive and easy to use. JSON uses JavaScript syntax but JSON format is text only. That is why it is easy to send and receive from the server.

```
{
  "title": "product",
  "description": "A product from Archies",
  "type": "gift",
  "properties":
  {
    "name": {"description" : "name of the product",
    "type": "string"
    },
    "id": {"description" : "the unique identifier for the
    product",
    "type": "integer"
    }
  }
},
"required": ["id", "name"]
}
```

The JSON format represents data of a document as a name, value pair. All related data of the user is stored into one document. And many related documents are stored together into a collection. Each document may have its own one or more sub documents called embedded document. As in the above mentioned example, "properties" has got its embedded document, "id" and "name". Value can be string, integer, array etc.

VI. CONCLUSION

The growing needs of today's technical world has given rise to the need of NOSQL databases which can store structured, unstructured and semi structured data. Document data stores are easy to use and dynamic data can be easily stored into them. The documents are independent so it improves the performance and decreases the side effects of concurrency. MongoDB is an open source tool of this category of database. The paper high lights some features and categorization of NOSQL. MongoDB is studied in particular. The representation of data using JSON is shown and the queries in relational database and MongoDB are compared.

REFERENCES

- [1] Rupali Arora, Rinkle Rani Aggarwal, " An Algorithm for Transformation of Data from MySQL to NoSQL (MongoDB)" *IJASCSE*, Volume 2, Special Issue 1, 2013
- [2] B Rama Mohan Rao, A Govardhan, "Sharded Parallel Mapreduce in Mongoddb for Online Aggregation", *International Journal of Engineering and Innovative Technology (IJEIT)* Volume 3, Issue 4, 2013
- [3] Sanobar Khan, Prof.Vanita Mane, " SQL Support over MongoDB using Metadata" *International Journal of Scientific and Research Publications*, Volume 3, Issue 10, 2013
- [4] Cristina BĂZĂR, Cosmin Sebastian IOSIF, "The Transition from RDBMS to NoSQL. A Comparative Analysis of Three Popular Non-Relational Solutions: Cassandra, MongoDB and Couchbase" *Database Systems Journal* Volume 5, Issue. 2, 2014

-
- [5] Prudence Kadebu, Innocent Mapanga, “ A Security Requirements Perspective towards a Secured NOSQL Database Environment” *International Conference of Advance Research and Innovation (ICARI)*, 2014
- [6] Donald G Firesmith, “Engineering Security Requirements”, *Journal of Object Technology*, Volume 2, Issue 1, 2003
- [7] S. Edlich, A. Friedland, J. Hampe, B. Brauer, “NoSQL: Einstieg in dielt Nichtrelationaler Web 2.0 Datenbanken”, 2nd edition.: Hanser Fachbuchverlag, 2010
- [8] Rabi Prasad Padhy, Manas Ranjan Patra, Suresh Chandra Satapathy, “RDBMS to NOSQL:Reviewing some next generation Non Realtional databases”, *International Journal of Advanced Engineering Science and Technology*, Volume 11, Issue 1, 2011
- [9] <https://docs.mongodb.com/manual/introduction/>