NOSQL OVER RDBMS IN IMAGE STORING USING MONGODB

Deepashree Karanjkar,
Student,
NCRD's Sterling Institute of Management Studies,
Nerul, Navi Mumbai

deepashreerk@gmail.com

Kanchan Barve
Student,
NCRD's Sterling Institute of Management Studies,
Nerul, Navi Mumbai

barvekanchan1@gmail.com

Prof. Mrunali Metri
Asst. Professor,
NCRD's Sterling Institute of Management Studies
Nerul, Navi Mumbai

mrunalimetri@gmail.com

ABSTRACT

In the modern world internet users are increasing incredibly day by day due to this more and more unstructured data is produced and consumed over the network. In the world of enterprise computing, we have seen many changes in languages, processes, platforms and architectures. But throughout the entire time one thing has remained unchanged i.e. relational databases. Relational database is broadly used in most of the application to store and retrieve data. They are best suitable to handle a limited amount of data. Handling a large volume of data like internet was inefficient in RDBMS. To overcome this problem "NO SQL" came into existence.

This paper introduces the concept of NoSQL databases, storing and retrieving of images as well as its advantages and disadvantages compared to traditional relational database management systems (RDBMS). The term "NoSQL" applies to a number of recent non-relational databases such as Cassandra, MongoDB, Neo4j, and Azure Table storage. As one of the leading NoSQL DBMSs MongoDB is selected for detailed analysis.

Keywords: *NoSQL , MongoDB, RDBMS, Big Data ,unstructured data.*

1. INTRODUCTION

In the last few years, by speedy advancement of web applications and social networks, the need for rapid development of applications for large number of users appeared. Existing relational DBMSs have proved to be too complicated because it failed to support the rapidly growing number of users and rapid application development.

Today, large volume of data is generated every day. Data has evolved greatly in recent years, in type, volume, and velocity with its rapid evolution attributed to the widespread digitization of business processes globally. There is simply much more semi-structured, unstructured data than structured. Unstructured data makes up 80% and more of enterprise data, and is growing at the rate of 55% and 65% per year [1].

Different technologies are changing and gaining acceptance to support the handling of large amount of data, which are produced and consumed from various sectors like Social media ,health care services , for Improving traffic safety etc. For example in facebook the current growth rate of images generated is 220 million new photos per week ,at the peak there are 5,50,000 images served per second[2]. Similarly, in Health care services it is estimated that 30% of world's storage will be related to health informatics, and mainly the medical images. Research forecasts that the market for medical imaging systems will grow to $49 billion in 2020.[3]. The increasing need for storing such semi-structured and unstructured data has led to the rise of databases called NoSQL databases. NoSQL databases are very much suitable for storage and retrieval of schema less data. NoSQL is open source and is capable of handling large amount of variety of data. In this paper we focus on one of the NoSQL technologies, namely MongoDB for storing images over RDBMS.

2. LITERATURE REVIEW

The review through literature surveys on large data generated in health care sector, data generated by passport authority, scanned documents in banking sector, social media as well as various real time application which are to be stored for an electronic future, where more and more users check in via electronic media like tablets, cell phones, etc. Users request their information in an easily transportable format, resultant constitutes the 3V's of big data.

Big Data is nothing but **data** with a **huge size**. Big data is a term used to describe a collection of data that is huge in size and yet growing exponentially with time [4].

3v's of Big data:

Volume - related to a size
Velocity - speed of generation of data
Variety - heterogeneous sources and the nature of data, both structured and unstructured.

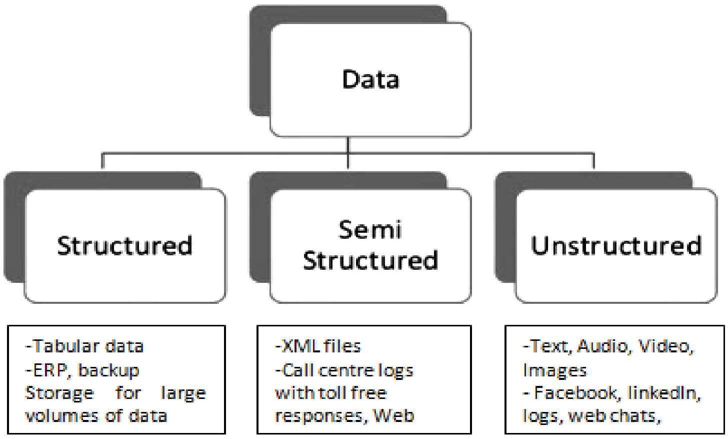
Data
Structured
Semi Structured
Unstructured
-Tabular data
-ERP, backup Storage for large volumes of data
-XML files
-Call centre logs with toll free responses, Web
-Text, Audio, Video, Images
- Facebook, linkedIn, logs, web chats,

Fig. 1: Categories of data

1. Structured data: Organized into specific fields as part of a scheme, with each field having a defined purpose. Data can be structured within each field(s) through data validation by enforcing the use of a standardized data format or allowing only a specific range of values entered in a field. Fields store length- defined data phone numbers, identification numbers, or ZIP codes. Even text strings of variable length are included like usernames in records. data may be human/machine generated as long as the data is created within an RDBMS structure. MySQL allows queries on this type of structured data within relational databases.

2.Unstructured data : Information that typically requires a human touch to read, capture and interpret properly. Data that cannot be easily organized using pre-defined structures. It may be textual or non-textual, and human generated or machine generated. It may also be stored within a non-relational database like NoSQL. Example: human generated data includes Text files, Email, Social media, website, media, business application and machine generated data includes **Satellite imagery, Digital surveillance(digital photos and video), sensory data(traffic data).**

3. Semi-structured data: Includes both the forms of data structured and unstructured. Semi-structured data can be viewed as a structured in form but it is actually not defined with e.g. a table definition in relational DBMS. Example of semi-structured data is a data represented in an XML file.

Tremendous amount of data is being generated every second due to advancement in technology this data is mostly unstructured. The traditional relational databases handles structured data efficiently and cannot process this large volume of unstructured data efficiently[5] .So there is a need of a technology which can efficiently handle such large volume of unstructured data. This inefficiency is handled by "Not Only SQL" called NoSQL.

NoSQL is an approach to databases that represents a shift away from traditional relational database management systems (RDBMS)[6]. NoSQL is a non-relational database management system, that does not require a fixed schema, and is easy to scale. NoSQL is used for storing large amount of unstructured data, which is growing rapidly than structured data and doesn't fit in relational schema of rdbms. NoSQL database is used for distributed data stores with huge data storage needs. For example like Facebook, Google, Twitter that collect terabytes of user data every single day.

NoSQL provides Several different varieties of databases that have been created to support specific needs and use case. These fall into four main categories. Figure 2 shows the categories of NoSQL.

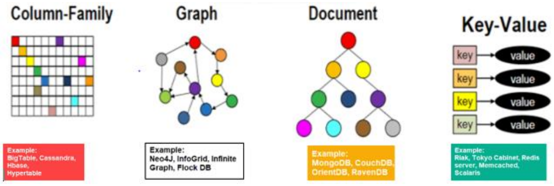

Fig.2: Categories of NoSQL

1. **Column-Family:** Column-Family NoSQL database stores data in table with rows and columns similar to RDBMS, but names and formats of columns can differ from row to row across the table. Column-Family database group columns of related data together and to group similar column names together column family is added along with column name. They can scale to manage large volume of data. Column stores use row and column identifiers as general purposes keys for data lookup.
2. **Graph:** A graph type database uses graph structure to store entities and relationship among those entities. The node in graph represents entity and edge represents relationship between nodes. Every node and edge has a unique identifier. Graph database are mostly used for social networks, logistics.
3. **Key-Value:** Key-value stores are most easy types of NoSQL databases. The key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBjec) etc[7].A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys[7]. For example a key-value pair might consist of a key like "Name" that is associated with a value like "Divij".
4. **Document:** Document NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats.

Document store is a tree like structure having single or multiple root elements below which sequence of branches, sub-branches and values are present. Each document can have same or different structure.

NoSQL encloses a wide variety of different database technologies but generally all NoSQL databases have a few features in common.

1. Non-relational NoSQL databases never follow the relational model. Never provide tables with flat fixed-column records.Work with self-contained aggregates or BLOBs. Doesn't require object-relational mapping and data normalization. No complex features like query languages, query planners, referential integrity joins, ACID.
2. Schema-free In NoSQL databases data can be inserted without a predefined schema. That makes it easy to make significant application changes in real-time, without worrying about service interruptions – which means development is faster, code integration is more reliable, and less database administrator time is needed.
3. Distributed Nosql is designed to distribute data globally i.e it can use multiple locations for write and read operations. Relational databases, in contrast, use a centralized application that is location-dependent, especially for write operations. NoSQL databases also support automatic replication, means you get high availability and disaster recovery without involving separate applications to manage these tasks. The storage environment is essentially virtualized from the developer's perspective.
4. **Scaling** provides improved performance, allowing for continuous data availability and very high read/write speeds.

MongoDB is an open source document-oriented NoSQL database released in 2009 which stores data in the form of JSON-like objects. It has emerged as one of the leading databases due to its dynamic schema, high scalability, optimal query performance, faster indexing and an active user community[8]. MongoDB can manage small data as well as large data efficiently. It holds a set of collections; a collection holds a set of documents. A document further is a set of key-value pairs. These documents have a dynamic schema which means that documents in the same collection do not need to have the same set of fields or structure[9].

Here is an example of a JSON document:

```
 {
course: "java",
details: {
        duration: "6 months",
          Trainer: "Sonoo jaiswal"
          },
Batch: [ { size: "Small", qty: 15 }, { size: "Medium", qty: 25 } ],
category: "Programming language"
}
```

3. PROBLEM DEFINITION

There are two main database management systems, RDBMS and NoSQL. Both are suitable for storing structured and semi-structured data but not structured (BLOB) data could cause a problem when using it with relational databases. The technology behind collecting and storing the images has relied on the development of the Information Technology.

The normal way is to store the file-data in other parts of file system, images were stored in separate files outside the databases and the databases stored only the link or paths to the Image files. This comes a problem: keeping the consistency between the file system and the records in the tables [9].

The existing data storage capabilities are not able to satisfy the needs of this massive amount of unstructured image data. This is a huge challenge for various organizations where it is a big struggle to share, manage and access this data in less cost. So organizations look for new methodologies to store these huge volumes of unstructured and semi structured data. Handling images poses the following challenges.

(1) Handling Different types of Data and image file formats

(2) Handling huge sized images

Traditional relational databases were designed in a different hardware and software era and are facing difficulty in meeting the performance and scale requirements of Big Data. NoSQL present themselves as alternatives that can handle huge volume of data.

4. OBJECTIVE

The main objective is to store and retrieve the large Blob image in mongoDB database efficiently and provide better performance and storage over challenges faced in RDBMS. Here NoSQL comes into the picture to handle unstructured Big data in an efficient way to provide maximum business value and customer satisfaction. That can be achieved using MongoDB.

5. RESEARCH METHODOLOGY

NoSql provides different types of database such as key-value , Graph Stores, Column-oriented and Document-Oriented. It provides various framework to support different types of database such as Cassandra, couchDB, HBase, MongoDB, etc. In this paper we focus on Document-oriented NoSQL technology, namely MongoDB.

There is a need to handle huge sized images say 1GB or more. As MongoDB databases can easily handle binary data and also huge sized data as it is using JSON as the data exchange format. MongoDB by default cannot store, huge sized images However, to overcome this issue GridFS API is provided by MongoDB. This work uses MongoDB GridFS.

GridFS is a specification for storing and retrieving files that exceed the BSON-document size limit of 16 MB[10]. Instead of storing a file in a single document, GridFS divides the file into parts, or chunks , and stores each chunk as a separate document[11]. By default, GridFS uses a default chunk size of 255 kB; that is, GridFS divides a file into chunks of 255 kB with the exception of the last chunk[11].

GridFS uses two collections to save a file to database. One collection stores the file chunks i.e fs.chunks, and the other stores file metadata i.e fs.files. The fs.chunks collection contains the binary file broken up into 255k chunks. The fs.files collection contains the metadata for the document[12].

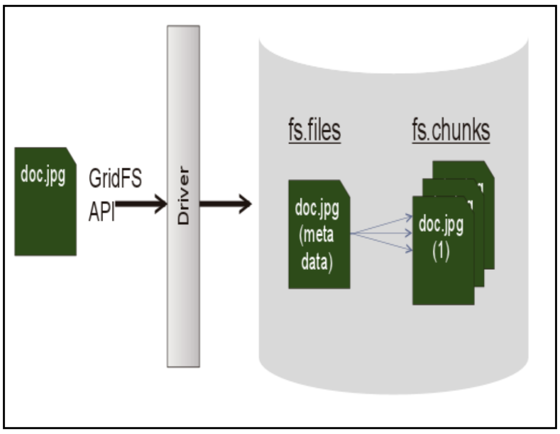

Fig. 3: Structure of GridFS

6. ANALYSIS AND FINDING

Initially images were not an integral part of RDBMS and the image files were stored separately. The RDBMS would store only the link to the image file. This is not a good practice as it is difficult to keep track of the image files link or path. Then in RDBMS a new type of data type or storage method called BLOB (Binary Large Object) was developed and this led to the possibility for storing images in the database. As images are generated huge in size we will use LBLOB(Long BLOB) which can hold upto 4 GB of file size. The Relational support to images is to be questioned, as the size of a image can exceed 4 GB as well. In such a case the user is expected to write coding to chunk it to handle it, this poses an extra overhead for the user and the application developer. This is highly complicated as any RDBMS solution fails in handling large images.

Problems faced in RDBMS while storing images are:

i) RDBMSs don't scale out.

ii) Chunking the image data and Sharding over many servers will be operationally inefficient.

iii) Integrated search functions are not available in RDBMS.

iv) It is difficult to store unstructured/semi-structured data in tables of any RDBMS.

NoSQL databases can solve the challenges described. A NoSQL Database has many advantages like

i) Horizontal scaling can be automatic.

ii) Supports chunking of huge images which help in autosharding.

iii) Integrated search functions are available which provide better search results.

iv) It's easy to store unstructured/semi-structured data.

NoSQL databases may be a better solution. The solutions given to store images are basically based on RDBMS. In the search for a better alternative to store images a comparative study of the performances with respect to storage and retrieval was done for both MYSQL and MONGODB. The result showed that the performance of MongoDB was better than MySQL. The images was stored and retrieved in both MySQL and MongoDB. The application program was written in JAVA.

Code to store images in MongoDB using GridFS

1. Save image

```
String newFileName = "mkyong-java-image";
File imageFile = new File("c:\\JavaWebHosting.png");
GridFS  gfsPhoto = new GridFS(db, "photo");
GridFSInputFile  gfsFile = gfsPhoto.createFile(imageFile);
gfsFile.setFilename(newFileName);
gfsFile.save();
```

2. Get image

```
    String newFileName = "mkyong-java-image";
    GridFS gfsPhoto = new GridFS(db, "photo");
    GridFSDBFile imageForOutput = gfsPhoto.findOne(newFileName);
  System.out.println(imageForOutput);
```

Output, the image is saved as following JSON format.

```
{
        "_id" :
        {
                "$oid" : "4dc9511a14a7d017fee35746"
        } ,
        "chunkSize" : 262144 ,
        "length" : 22672 ,
        "md5" : "1462a6cfa27669af1d8d21c2d7dd1f8b" ,
        "filename" : "mkyong-java-image" ,
        "contentType" :  null  ,
        "uploadDate" :
        {
                "$date" : "2011-05-10T14:52:10Z"
        } ,
        "aliases" :  null
}
```

7. LIMITATION AND FUTURE SCOPE

MongoDB is becoming even more popular than it is now. More people want to learn about it. As far as the future of data is concerned, is very sure that data will go increasing data by day due to advancement in the technology, every handheld device will generate a huge amount of data. Nearly 80% of data is unstructured, no one other than NOSQL can handle this data. MySQL will continue as the usual tool for data analysis and NoSQL, which is evolving, will emerge as the free tool for data analysis. There are few short comes in NoSQL such as security, memory management, use of joints etc, these all disadvantages will be resolved as the time passes. Earlier, MongoDB was a simple database that aimed at rapid application by building-from-ground. Over time, it took JSON document format as an approachable extension that could efficiently handle complex data types. You can use MongoDB for your everyday needs without any problems. With continuous development cycle, new features are getting added almost every other month. So you can think MongoDB as non-SQL only in nature. It has almost all the features as of an RDBMS.

8. CONCLUSION

Relational database management systems (RDBMSs) are traditional storage systems designed for structured data. RDBMSs are facing challenges in handling Big Data and providing horizontal scalability, availability and performance required for Big Data applications. Nosql provides efficient solution for handling those challenges. MongoDB is a advancement in the big data and with its ability to offer multiple data type support and better efficiency, it will only spread its grasp across the data industry. RDBMS are still definitely needed but the storage requirement for the new generation of applications are huge different from legacy applications. We can choose MongoDB instead of MySQL because of two factors, ease of use and performance. MongoDB GridFS is a high quality specification used for storing images and large files in MongoDB. It ensures that the file is divided into chunks and stored into a database. The main advantage of this approach is that only a portion of the file can be read without loading the entire file into the memory. Posing a challenge to traditional Relational Datatabase System, MongoDB is definitely the future of data storage. In future there will be lots of sites running with MongoDB.

REFERENCES:

[1] https://www.datamation.com/big-data/structured-vs-unstructured-data.html

[2] https://code.fb.com/core-data/needle-in-a-haystack-efficient-storage-of-billions-of-photos/

[3] A DBA's Guide to NoSQL, Apache Cassandra, Datastax-2014.

[4] https://www.guru99.com/what-is-big-data.html

[5] M. Stonebraker, S. R. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland. The end of an architectural era (it's time for a complete rewrite). In VLDB, Vienna, Austria, 2007

[6] http://basho.com/resources/nosql-databases/

[7] https://www.w3resource.com/mongodb/nosql.php

[8] https://code.tutsplus.com/articles/mapping-relational-databases-and-sql-to-mongodb--net-35650

[9] https://www.researchgate.net/publication/289519291_Storing_of_Unstructured_data_into_MongoDB_using_Consistent_Hashing_Algorithm

[10] https://codingsans.com/blog/nosql-vs- relational-database

[11] https://docs.mongodb.com/manual/core/gridfs/

[12] https://www.mongodb.com/blog/post/building-mongodb-applications-binary-files-using-gridfs-part-2

[13] https://www.researchgate.net/publication/313400371_Big_Data_Analysis_and_Storage

[14] A-NoSQL-Solution-to-efficient-storage-and-retrieval-of-Medical-Images.pdf

[15] https://www.mkyong.com/mongodb/java-mongodb-save-image-example/

[16] https://www.thecrazyprogrammer.com/2016/01/save-and-retrieve-image-from-mysql-database-using-java.html