ISSN: 0975-8585

Research Journal of Pharmaceutical, Biological and Chemical Sciences

Effective Handling of Patient Health Records Using NoSQL: MONGODB.

Subham Kumar Goyal*, Souvik Das, and Lakshmi M.

Department of Computer Science and Engineering, Sathyabama University, Chennai, Tamil Nadu, India.

ABSTRACT

Big data are grooming in reality. SQL does not have capacity to handle a very huge amount of data. All applications are now working in view to a vast volume of data[1]. Data is increasing massively in majority of stream of jobs , let it be employee details or health records. So the applications being used to handle these type of data should be modified too. Not only the applications but the databases and warehouses where we store these data have to be modifies too. SQL can store data in different tables and databases but later it is very difficult task to retrieve the same as that will include loads of join operation and very multifaceted transactions. So in this paper we propose to build an application for hospital management and to handle patient health records . Our application uses a NoSQL database(i.e here we use mongodb) for storing and retrieving the data. We are implementing Mongo lab in our application deployment. Each record and its associated data will be stored in a sole document thus simplifies the data access. Here, unlike SQL databases, the documents stored are schema free and similar to each other, this is a big advantage of NoSQL and helps in modelling unstructured data. We also use the tokenization concept to ensure security. We convert the user credentials like name, password, phone no, email id etc. into ASCII values and store it in separate mongo db. The patients' medical history, lab reports, medicine prescription etc. will be stored in a separate Mongo db.

Keywords: Big Data, patient health records, No SQL, databases, Hadoop

Corresponding author

ISSN: 0975-8585

INTRODUCTION

Data is everywhere. You chat with your friends, it's a data , you pay your bills-that's data, you travel-that's data, you go to a doctor -that's a data. In today's era where you are surrounded by data, Storing data has become a key issue.

There are many diverse kinds of databases available which helps in storing your data in best and effective way. SQL databases have ruled this stream for a long period of time. It helps us store data, link data from different databases, edit and update data, and retrieve the needed data whenever we want. But there are some limitations with SQL because of its data modelling techniques. This is why the data scientists approached up with the perception of NoSQL databases. NoSQL(also known as NOT ONLY SQL) databases are rapidly growing and deploying in many internet companies and other enterprises.

Oracle databases, MySQL , Microsoft SQL servers and many other relational databases follow the same basic SQL architecture[1]. But as we all know that, any data has to be pre-processed into appropriate schema before storing it in SQL databases. This is one of the reasons why these databases are not very effective for very hefty amount of data. NoSQL databases allow us to process structured, semi-structured, as well as unstructured data.

Relational databases

The most conventionl methods used to deal with data is to use traditional data warehousing system which is based on standard RDMS(Relational database management system). In this case, data is extracted from various sources either internal or external [2-3], then these are selected and aggregated, and then stored in data warehouse. As the volume of records increases they are stored in separate machines, tables are partitioned and stored in different nodes. But even in doing so, the time required to administer the data and the time taken to analyse and access the data also increases. More complex statements needs to be carried out to access these data.

Solution to existing problem

We need a system that can process all the queries in parallel in all the nodes or machines in the cluster to access the data. Mapreduce and hadoop comes handy in such scenarios. Hadoop follows a distributed file system to store the data and process it. The data is partitioned using basic partitioning technique and then stored in different nodes. The data here is mapped and the query undergoes parallel processing on the data distributed among different nodes. The results from all the mapper processes are combined in the last to produce the desired final output. Thus even big complex queries can be partitioned and can be executed in very less amount of time.

Literature survey

1. Jagdev Bhogal,Imran Choksi in 2015 in 29th International Conference on Advanced Information Networking and Applications Workshops stated the concepts behind NO SQL, highlights the different No SQL database types , and provides advantages of using No SQL.A small prototype application is generated to illustrate the benefits of No SQL and the differences between the SQL and NoSQL. But they were unable to cope with the security issues. Security issues remain in storing and retrieving the data.
2. Asadulla Khan Zaki in 2014 in international Journal of Research in Engineering and Technology introduced the causes of migrating towards NoSQL databases, characteristics, classification of NoSQL databases. Conclusively the security issues in NoSQL Databases were discussed and the security enforcement mechanism was proposed. To overcome the issues, it had to compromise with scalability and performance features.
3. Manpreet Singh, Jatinder Singh Bhatia and Devansh Malhotra in 2014 in International Journal of Research in Engineering and Technology came up with the structures and techniques to efficiently make use of big data architecture. It defines the Big data analytics using MapReduce concepts and promotes the use of Big Data in different organizations. Security , privacy and Capacity are a major drawbacks which need to be taken care of in future.

ISSN: 0975-8585

4. Balla Wade Diack1 , Samba Ndiaye1 and Yahya Slimani2 in year 2013 in International Journal of Advanced Science and Technology discussed about the CAP theorem, its evolution and its influence on distributed systems. It also talks about the misunderstandings and problems aroused by this theorem. Finally, they gave the updates on CAP brought by some researchers. But they failed to cope with the Real world problems like scalability and the use of this theorem in wireless network of computers.

Proposed work

In this paper, we are trying to develop a more efficient methodology to manage the patient health records in hospitals. The present hospital management system uses an SQL database which uses the local memory to store the data. The data is present in the hospital system and is only accessible by the hospital. In our proposed system we are using a NoSQL(MongoDb) database which is a cloud based database and provides database as a service[1]. Mongo Db being a cloud based database allows the access to the database from anywhere and anytime.

- If a patient is out of town and he falls sick. If he needs immediate medical attention but the doctors requires the patient's medical history.
- If a patient is out of medicine and has lost the prescription.

In these cases patient can easily access all its details through the online portal.

Implementation

For implementing the health record system, we are developing a web based application which is used to take input as patient details and prescription and is going to store it in mongo db using Mongo lab. Mongo db uses a document database design. Table schema is required for SQL databases but the document database has a much more flexible schema[2]. Here all the records are taken through the jsp pages and is passed to the hadoop analysis program where the data will be splitted. User profile data is saved separately in a separate database and its medical records are stored in a separate database which can be seen in the architecture diagrams.

Architecture

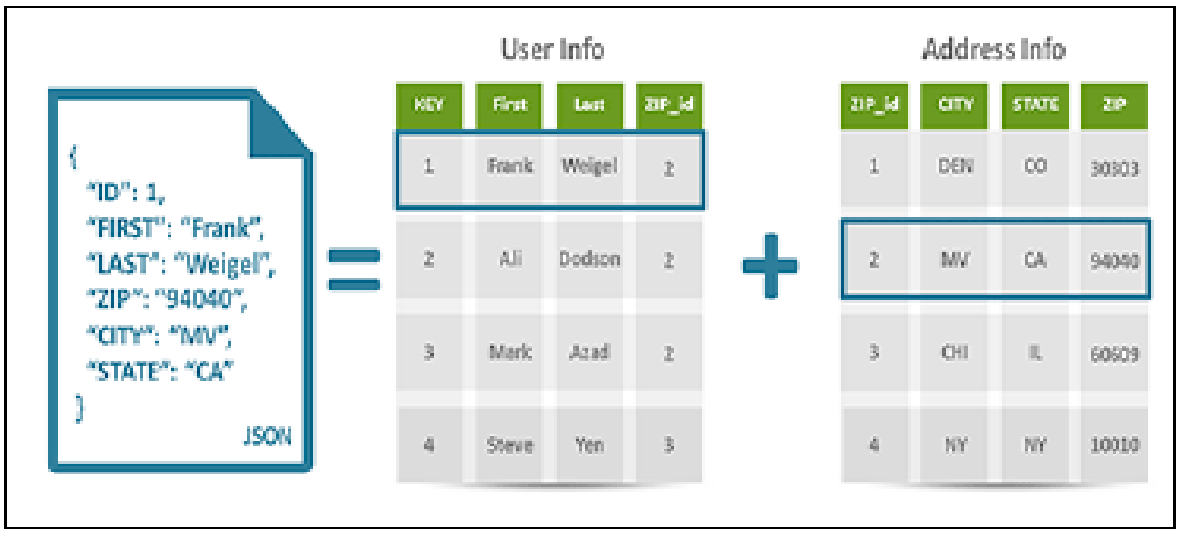

Fig. 1 Data structure diagram

For the user data, first an index is made for each and every user and its rest of the profile is stored separately related to the index table. As seen in the above diagram, the user info table stores a key pair for each user and the address info table stores the rest details provided by the user. The two tables i.e. the user

ISSN: 0975-8585

info table and the address info table is related using the zip id created for each user. Also when the user sends a request for data, it is fetched directly from the database available in the cloud. Mongo db supports availability and partitioning in the cap theorem. So, it is always available and it partitions the data into key value pair to make an effective searching of data and retrieving the same.

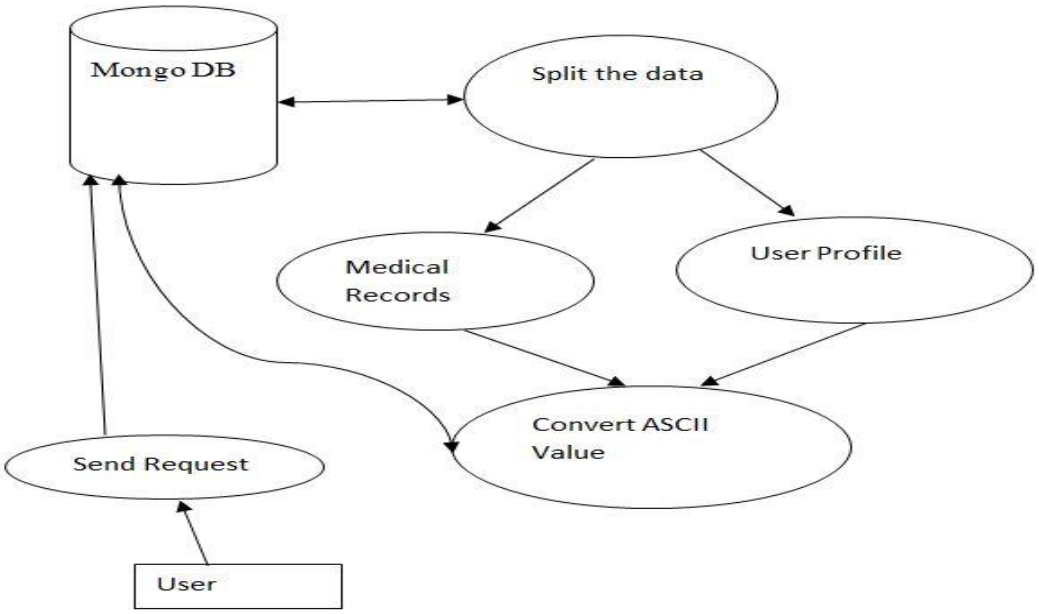

Fig. 2 Data flow diagram

Security

Security can be ensured by saving the data in ASCII value format. Before the data is sent to the cloud server it is converted into its ASCII value using the following code:

```
public String getASCIIVal(String obj) {
    byte[] b = obj.getBytes(StandardCharsets.US_ASCII);
    String asciiString = Arrays.toString(b);
    return asciiString;
}
```

Thus it is difficult to track what data is sent and to change the data values. Also when we try to retrieve the values from the database, the values will be converted back to its original form and then it will be exposed to the user [4]. The following function is called to convert the ASCII values back to string format:

```
public String getStringFromAscii(String obj) {
    String orgString = null;
    String temp = obj.substring(1, obj.length() - 1);
    System.out.println("Trimmed " + temp);
    String[] split = temp.split(",");
    byte[] getBytes = new byte[split.length];
    for (int i = 0; i < split.length; i++) {
        split[i] = split[i].replaceAll(" ", "");
        getBytes[i] = Byte.parseByte(split[i]);
    }
    System.out.println("Orginal String=>" + new String(getBytes));
    orgString = new String(getBytes);
    return orgString;
}
```

CONCLUSION

Thus we get a new System that can handle the patient health record in a more effective fashion. It lets users to register their details, store their information, medical records, lab reports, and prescribed medicines, and retrieve data according to their privileges. In this system, user can change only their personal information; they can view their medical history and reports but cannot change them. Doctors can view a patient's medical history and reports and can submit a new record according to the latest diagnosis. Since here the records are all stored in cloud, it is much secure, always available, and accessible from anywhere using the web application. As he server here runs on the same system as the application, it responds really quick. It uses hadoop to analyse and distinguish between sensitive data and insensitive data, and store them by converting them to its respective ASCII values.

REFERENCES

[1] Jagdev Bhogal, Imran Choksi. 2015, "Handling Big Data using NoSQL", 29th International Conference on Advanced Information Networking and Applications Workshops;1-6.
[2] Asadulla Khan Zaki. International Journal of Research in Engineering and Technology 2014:403-409.
[3] Manpreet Singh, Jatinder Singh Bhatia, Devansh Malhotra. International Journal of Research in Engineering and Technology 2014; 16:130-136.
[4] Balla Wade Diack, Samba Ndiaye and Yahya Slimani. International Journal of Advanced Science and Technology 2013;56:1-12