ISSN 2229-5518

A NoSQL Solution to efficient storage and retrieval of Medical Images

D.Revina Rebecca, I.Elizabeth Shanthi

Abstract— Healthcare sector is one of the largest service sectors in the world. Health care industry produces huge volumes of Health care related data and Medical images has become an integral part of Health care Information Systems (HIS). Medical images has to be archived for future references, Retrieved for current clinical decisions and sometimes it may be real time as well. Real-time clinical decision support where images have to be shared in real-time is a challenge. As the size of medical images is huge, the challenge lies in effective Medical Image Archiving, retrieving and sharing. This paper deals with the challenges involved in the existing Medical Image archiving, retrieving and sharing systems and also discuss a NoSQL solution for the same.

Index Terms— Cloud computing, Health care, Medical images, NoSQL, MongoDB, MYSQL, Medical Image Archiving , RDBMS.

1 Introduction

The IT industry has given solutions to various problems in different sectors in the society. Health care being the most important sector and the need for having collaboration with the IT has gone to the peak. The Health care Information systems (HIS) and radiological Information systems (RIS) are already adopted by the IT industry and there are many customized systems available in the market. The most challenging aspect in health care industry is to store and share images. Images are stored in PACS(Picture Archiving and Communication Systems) which is primarily based on RDBMS[1]. This paper is a study of the challenges faced in storing medical images in an Relational Database based PACS. We also try to give a solution to store and archive images and DICOMM (Digital Imaging and Communications in Medicine) Objects in a Non-relational data store. The performances of NoSQL database MongoDB and Relational database MySQL are compared and the results are shown.

2 MEDICAL IMAGES

Medical images are data that are collected using medical imaging devices. The device may be any one of the following device like X- ray device, MRI device, Ultrasound device, Positron Emission Tomography (PET) device or a CT devices. The image interpretation process called "reading" or "diagnostic reading." The image interpretation report is generated from the medical image data. The images accompanied by the image interpretation report will be sent to the physicians who request that information. These image interpretation reports are usually digitized, stored, managed and distributed in plain text are in stored in Radiology Information System (RIS).The accompanying images are stored in a Picture Archiving Communication System (PACS).

2.1 Medical Image Archiving

The technology behind collecting and storing the medical images has relied on the development of the Information Technology. The storing of images changed over time. To start with, the images were stored in separate files outside the databases and the databases stored only the link or paths to the Image files. But this scenario changed as a new type of data object called BLOB (Binary Large Object) was introduced in databases [10]. All these storage evolution happened in the Relational databases, which has many limitations in storing medical Images.

2.2 Need for Medical Image Archiving

Medical imaging is a big slice of the digital data. More than 400 million procedures a year in the U.S. and this is the same in the Indian scenario. Each procedure now involves at least one medical image. Very huge image data files are produced, and the modalities used to create them are constantly evolving. With medical image archive volumes increasing exponentially, many healthcare organizations are wondering how they will manage and pay for potentially huge amounts of future data storage. These images are to be archived for many years.

3 CHALLENGES IN MEDICAL IMAGE ARCHIVING AND RETRIEVING

The Healthcare organizations today use the picture archiving and communication system (PACS) and the format being DICOM. The existing data storage capabilities are not able to satisfy the needs of this massive amount of medical imaging data. This is a huge challenge for healthcare organizations where it is a big struggle to share, manage and access this data in less cost

- *D. Revina Rebecca , is cuurently working as an Associate professor in REVA University, Bangalore and pursuing her PhD programme from Avinashilingam University for women, Coimbatore, Tamil Nadu,India. E-mail: drrevina1@gmail.com*
- *Dr. I.Elizabeth Shanthi, is currently working as Associate Professor in Avinashilingam University for women, Coimbatore, TamilNadu,India. E-mail: shanthianto@gmail.com*

[14].According to HIMMS analytics Medical images will require 30% of worlds total storage in the coming years. [2] Healthcare applications will be dealing with large binary files which are results of various tests done on the patient from various medical imaging devices. It can be radiology test results, x-ray and MRI images, CT scans. This may also include medical records created by scanning paper documents. It is not uncommon for a large hospital or a medical insurance provider to have 50-100 TB of patient data, out of which 90% of the information are unstructured binary data. The healthcare providers look for new methodologies and paradigms to store these huge volumes of unstructured and semi structured data. Handling medical images poses the following challenges [5]

(a) Handling Different types of Data and image file formats
(b) Handling huge sized images

3.1 Handling different types of image formats

The medical images are from different modalities which may be of different formats [6,7,8].The medical image files sourced from different modalities may be either in DICOM (Digital Imaging and Communications in Medicine) format or in raw format. In general it is unstructured or semi-structured. Picture Archiving and Communication Systems (PACS) is used to store Medical images which use a Relational Database Management System (RDBMS) in the background.[11,12]. The medical images are so far managed using RDBMS. Medical images are semi structured. The traditional Relational data model (RDBMS) handles structured information very effectively. RDBMS cannot handle semi-structured data efficiently [13]. So there is a need to look for an alternate solution to handle the medical images. In [13] the author argues SQL is not the answer to handle semi-structured data. So the need of the hour is to use a non-relational database, which may be the possible solution to this challenge. NoSQL databases come to rescue, as NoSQL databases can effectively handle unstructured or semi-structured information.

3.2 Handling Huge Sized Images

Medical images are very huge in size. Handling huge images poses a challenge in archiving, retrieving and sharing as well. The use of RDBMS in storing huge images is a challenge, as the size is limited in RDBMS. NoSQL databases can easily handle huge sized images at ease. Also there is a hype going around moving the health care information to the cloud. As RDBMS is a worst fit for the cloud applies to say storing medical images aswell. So these considerations are to be placed before going for a NoSQL database.

4 A NOSQL BASED SOLUTION

The schemaless databases which are efficient to support data distribution in cloud computing are called NoSQL databases. They are also referred as cloud databases NoSQL stands for Not only SQL, it is a common term used to denote the Non-relational databases. “NoSQL” is often used as an umbrella category for all non-relational databases. PACS being based on RDBMS(Relational Database Management Systems) has many disadvantages when the medical images are moved to the cloud or to a distributed environment. Some of the challenges faced in RDBMS when medical imges are stored in are [5]

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.

4.1 MongoDB and Medical Images

Digital medical images represent a particular challenge from a data management point of view. Each year millions of medical images are generated from the millions of medical procedures that come from different modalities. The challenge of handling medical images lies in handling different data formats including DICOM and raw image data. 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 cannot store, by default huge sized images, However, to overcome this issue, GridFS API is provided by MongoDB. GridFS allows storing files in several chunks (256 kb max per chunks). GridFS uses one collection to store the file chunks and another one to store file metadata. This work uses MongoDB GridFS.

4.2 MongoDB and its features

MongoDB has few features that can handle the challenges we face in handling medical images. MongoDB can easily handle different image file formats also it can handle huge images.

a) BSON: MongoDB uses a an open data format called BSON which is short for Binary- JSON(JavaScript Object Notation), a great way to exchange data . It’s also a nice way to store data. MongoDB stores everything together in a single document. BSON is schemaless and documents can be updated individually or changed independently of any other documents. As an added bonus, JSON keeps all related data in one place thus providing an excellent. This provides faster retrieval capabilities.

BSON has additional feature which has the the ability to add types for handling binary data. So MongoDB can easily handle any type of medical image format including binary data format.[9].

b) GridFS(Grid File System): This is a powerful feature in MongoDB which can handle large binary files. Binary files including videos, images, PDFs, etc. can be stored using GridFS. The supporting information of medical images can be fit into into JSON documents. As we in the case of medical images it is necessary to store binary artifacts as well as all the supporting information about those artifacts [9]. MongoDB manages large fles using GridFS and their associated metadata

ISSN 2229-5518

as a group of small files. When a query is made for a large file, GridFS automatically reassembles the smaller files into the original large file. There is a need to store all other related patient details apart from the medical image like information about the patient (name, address, insurance provider, etc.) and his associated medical records (office visits, blood tests and labs, medical procedures, etc.). MongoDB stores all these information as JSON documents. MongoDB makes it easy to accommodate variability in content and structure.

GridFS allows large binary files to be chunked by breaking the files into smaller files called "chunks" and saving them in MongoDB and handle each chunk independently.

GridFS uses two collections to save a file to a database: fs.files and fs.chunks. The fs.chunks collection contains the binary file broken up into 255k chunks. The fs.filescollection contains the metadata for the document. Refer Fig.1 and Fig.2.

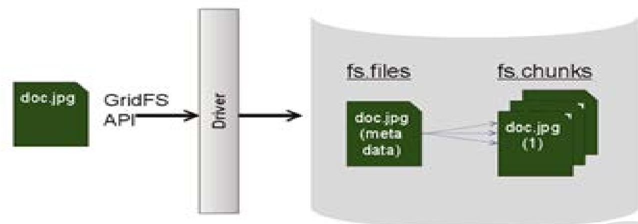

Fig.1 GridFS Structure

Fig.2 MongoDB-Collections showing chunks files

5 MYSQL AND MEDICAL IMAGES

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 files. 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. This enabled accessing to images as part of a single transaction in the same manner as to other types of stored data.[6]. As medical images are huge in size we will use LBLOB(Long BLOB) which can hold upto 4 GB of file size.[10] . The Relational support to medical images is to be questioned, as the size of a medical 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 medical images.

6 COMPARITIVE STUDY OF MONGODB AND MYSQL

Considering all the above factors, RDBMS would be a worst fit to store medical images. NoSQL databases may be a better solution. The solutions given to store medical images are basically based on RDBMS [11]. In the search for a better alternative to store medical images a comparative study of the performances with respect to storage and retrieval was done for both MYSQL and MONGODB. The time complexity was studied in i3 and i5 processors. 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 and the time was recorded on single system.The experiment was conducted on Intel i3 processors and Intel i5 processors. The sizes of images in the experiment ranged from 5 to 50 MB. The time taken was recorded for storing and retrieving the images to and from MongoDB and MySQL respectively.

6.1 Program to store and retrieve medical images in MongoDB

For storing and retrieving images from MongoDB, we are using 2 functions saveImageIntoMongoDB() and saveToFileSystem() respectively.

Storing images into MongoDB

```
void saveImageIntoMongoDB(DB db, String source) throws IOException {
File imageFile = new File(source);
GridFS gfsPhoto = new GridFS(db, "photo");
GridFSInputFile gfsFile = gfsPhoto.createFile(imageFile);
        gfsFile.setFilename(imageFile.getName());
        gfsFile.save();
}
```

Retrieving images from MongoDB

```
void saveToFileSystem(DB db, String source, String dest) throws IOException {
GridFS gfsPhoto = new GridFS(db, "photo");
GridFSDBFile imageForOutput = gfsPhoto.findOne(source);
imageForOutput.writeTo(dest + source);
}
```

The Image will be chunked and stored in MongoDB.
MongoDB keeps track of the chunks and retrieves the same.
The user need not keep track of the details. MongoDB can thus handle large medical images .

6.2 Program to Store and retrieve Medical images in MYSQL

i) Storing images into MySQL

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/"+"test","root", "root");
Fileimgfile=new File("C:\\Users\\foto1.jpg");
FileInputStream fin = new FileInputStream(imgfile);
PreparedStatement pre = con.prepareStatement("insert into
```

ISSN 2229-5518

```
myData values(?,?,?)");
pre.setString(1," foto1.jpg");
pre.setInt(2,1);
pre.setBinaryStream(3,(InputStream)fin,(int)imgfile.length());p
re.executeUpdate();
```

ii) Retrieving images from MySQL

```
Connection con = DriverManag-
er.getConnection("jdbc:mysql://localhost:3306/"+"test","root",
"root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select binData from myData
where name=\" foto1.jpg\"");
int i = 0;
while (rs.next()) {
        InputStream in = rs.getBinaryStream(1);
        OutputStream f =
new FileOutputStream(new File("foto1"+i+".jpg"));
        i++;
int c = 0;
        while ((c = in.read()) > -1)
                f.write(c);
}
```

7 RESULTS

The time complexity for storing and retrieving medical images in both MongoDB and MySQL was recorded and the results are shown below.

7.1 Time Complexity for the i3 processor

Figures 3 and 4 shows the graphical representation of the time taken to store and retrieve images using an i3 processor respectively.

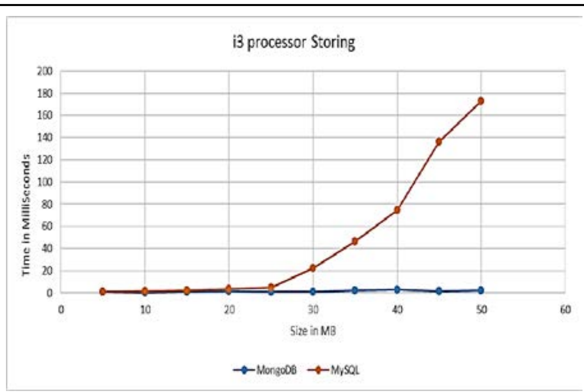

Fig.3- i3 Processor Storing

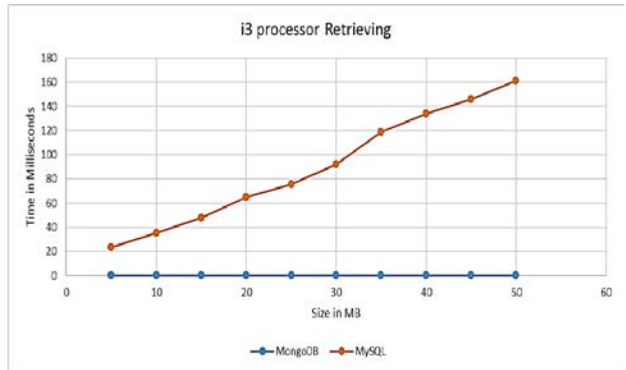

Fig.4- i3 Processor Retrieving

7.2 Time complexity for i5 processor

Figures 5 and 6 shows the graphical representation of the time taken to store and retrieve images using an i3 processor respectively.

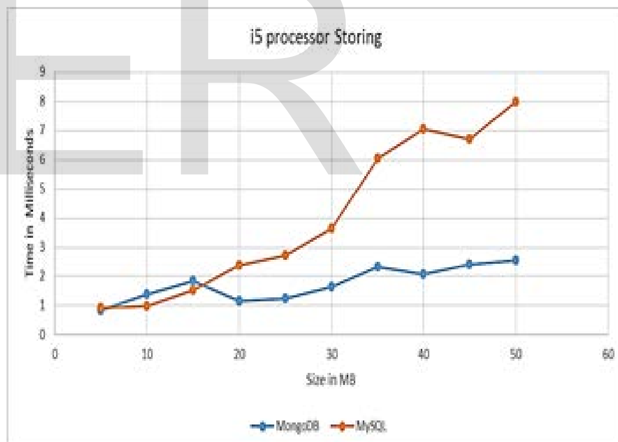

Fig.5 i5 - Processor Storing

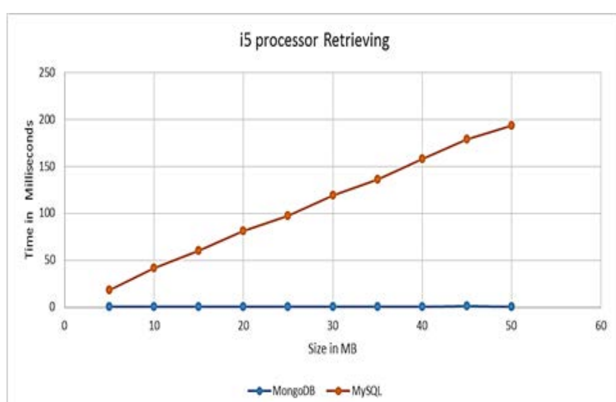

Fig.6- i5 Processor Retrieving

8. Conclusion

The experimental result of this study clearly indicates the performance of the NoSQL database - MongoDB is better than MySQL. The time for storing and retrieval in MONGODB was consistently lesser, even when the size of the images increased. This indicates that MongoDB is a better option to store medical images. The challenges faced in RDBMS can be overcome using the NoSQL database MongoDB.

The need for a better storage alternative for handling medical images may be done through MongoDB in the future. Also this takes us to the next level work, where we can combine this NoSQL database with cloud environment for effective storing and retrieval of medical images. So far archiving and sharing of medical images in the cloud was done only through relational databases, which has lot of drawbacks as discussed. Our future work will be based on moving these medical images to the cloud with a better performance using MongoDB. This work may be of great help in the field of telemedicine and Tele-Radiology.

Acknowledgment

We would like to thank Goutham Doddamani, G.R.V.Manikanda, Mahendra Reddy and Suman Cheedala, MCA students, REVA University, Bangalore for their contribution to this work.

References

[1] S. J. Rascovsky, J. A. Delgado Luís A. Bastião Silva, Louis Beroud, Carlos Costa and José Luis Oliveira, Medical imaging archiving: a comparison between several NoSQL solutions, Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference Year: 2014, , DOI: 10.1109/BHI.2014.6864305.

[2] http://www.himssanalytics.org

[3] Tse-Chuan Hsu , Dong-Meau Chang , Hsin-Jan Lee, The study of application and evaluation with NoSQL databases in Cloud Computing, 2014 International Conference on Trustworthy Systems and their Applications

[4] Liliana BYCZKOWSKA-LIPIŃSKA, Agnieszka WOSIAK, Multimedia NoSQL database solutions in the medical imaging data analysis, PRZEGLĄD ELEKTROTECHNICZNY, ISSN 0033-2097, R. 89 NR 12/2013

[5] Rick Cattell, Scalable SQL and NoSQL Data Stores , SIGMOD Record, December 2010 (Vol. 39, No. 4)

[6] Michele Larobina & Loredana Murino, Medical Image File Formats, Published online: 13 December 2013 # Society for Imaging Informatics in Medicine 2013,J Digit Imaging (2014) ,27:200–206, DOI 10.1007/s10278-013-9657-9

[7] http://www.nf.mpg.de/vinci3/doc/image-formats.html

[8] Michele Larobina & Loredana Murino, Medical Image File Formats, Published online: 13 December 2013 # Society for Imaging Informatics in Medicine 2013,J Digit Imaging (2014) ,27:200–206, DOI 10.1007/s10278-013-9657-9

[9] https://www.mongodb.com/blog/post/building-mongodb-applications-binary-files-using-gridfs-part-1

[10] MySQL Reference manual

[11] US Patent WO 2013123085 A1: Cloud-based medical image processing system with anonymous data upload and download.1109/TKDE.2007.190746.(PrePrint)

[12] S. J. Rascovsky, J. A. Delgado, A. Sanz, V. D. Calvo, and G. Castrillón, "Informatics in Radiology: Use of CouchDB for Document-based Storage of DICOM Objects," Radiographics, vol. 32, pp. 913-927, 2012

[13]] 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.

[14] D.Revina Rebecca et al, Impact of adapting Cloud Computing in health care industry for storing medical Images,National conference on emerging innovative technologies, (2014)-ISBN:978-81-923796-5-4.