A Hybrid Data Model to Share Medical Images

D. Revina Rebecca
Research Scholar
Avinashilingam University
Coimbatore, India

I. Elizabeth Shanthi, PhD
Associate Profeesor
Avinashilingam University
Coimbatore, India

ABSTRACT
The challenges involved in effectively storing, retrieving and sharing medical images have led the researchers to look into various means and methods of doing the same. It is the need of the hour for a hybrid data model which will solve all the challenges involved in it. In the previous work the suitability of using NoSQL databases in storing and retrieval of medical images was analyzed. It was found the MongoDB, A NoSQL database suitable to handle medical images. It is also necessary to look for a better way to transfer medical images. Since medical images are huge, it is a challenge to share it with minimal latency. A Model based on a distributed strategy using the sharding environment is proposed. It may be considered to be a hybrid data model using MongoDB to share and handle medical images. This data model is based on storing and retrieving using parallel processing and distributing the data across many machines. The aim of this paper is to study the effectiveness of the sharding or distributed processing concepts available in the NoSQL databases and how it helps us to enhance the bandwidth in sharing of huge medical images.

General Terms
Health Informatics, Distributed Databases, Sharding, Cloud Computing.

Keywords
DICOM, Cloud Computing, MongoDB , Chunked Storage, ,sharding,parallel processing, Medical Images.

1. INTRODUCTION
Today it is possible to share any information instantly, with the but instant sharing of huge Medical images has few challenges. The Cloud can aid to the instant storing of medical images, but the literature lacks in directing the means and methods of doing it. The failure of the Relational Databases to work with the cloud has led to a few cloud databases also referred as NoSQL Databases. As these NoSQL databases allow flexible data modeling, it is necessary to recommend a suitable Data Model which can work well with the Cloud technology and also with the Medical Images. It is desirable to have a Data model which suits NoSQL Databases and Medical Images which enhances the movement to the Cloud Technology. So a hybrid data model is proposed for handling Medical Images with high through put and minimal network latency. This paper is structured as follows: Section II, related work in the area of handling medical Images is discussed. In section III the various medical imaging methods and the need for a better way to transfer medical images using distributed methodology, sharding environment is discussed. Further in section IV the implementation of sharding in Mongodb is discussed. In Section V conclusions and future work is presented.

2. BACKGROUND RESEARCH &CHALLENGES

2.1 DICOM and NoSQL
The digital imaging and communications in medicine (DICOM) protocol is the one default standard for image data management in healthcare. The DICOM file contains two parts stored as a single object i)A header that stores Meta data ii)Image data stored as pixels. The medical images in DICOM format is acquired from different types of modern modalities like CT-Scanner,MR, X-ray, etc. these images are huge in size and the challenege lies in the image data transmission or sharing of these images required in telemedicine or teleradiology. Few attempts have been made to improve the data transmission time between medical imaging systems.

Rascovsky et al[8] developed a CouchDB based solution to Store medical images. The author argues the disadvantages of RDBMS to store and access DICOM metadata. DICOM objects are heterogeneous and it is unable to represent using an RDBMS. A DICOM object can be loaded into RDBMS, if and only if most of the metadata is stripped out.

The author also concluded the suitability of Document based databases in storing medical images. The document-based databases do not have the limitation of RDBMS databases. Document-based databases are much suitable than RDBMS for storing and retrieving DICOM objects, as they are schema-less. DICOM objects are freely structured and it is not possible to force them to fit into a predefined schema in RDBMS.

Luís A et al [9] developed a PACS archives based on MongoDB and CouchDB. The authors concluded the inability of both NoSQL databases in handling huge files. The authors reiterated the need for a better solution for storing huge files and there was performance degradation as the file size increased. The conclusion of the study was to find a better replication schema to handle bigger files.

A poster paper by Luan Henrique Santos , et al [10]suggest a work based on MONGODB. In a previous work [11] two NoSQL databases, the performances of Cassandra and MongoDB were compared. It was proved experimentally that the performance of MongoDB was better in huge files. So we conclude MongoDB, a Document based model to be suitable to store medical images. The literature clearly indicates the difficulty in handling huge medical images. It is also required to look for NoSQL databases for moving these images to the Cloud environment.

2.1 Data Modeling to handle huge Medical Images.
As medical image sizes vary from 10 GB to 300 GB, these images are categorized as Big Data. A recent study predicts that there is a potential growth for the medical imaging [13]. A hybrid data model which can possibly handle huge sized

medical images is the need of the hour. The healthcare industry is moving to the cloud and this adaption is essential to handle the huge storage required for storage of medical images. Also in [6] the author has discussed the advantages of NoSQL databases over RDBMS. As the literature shows the suitability of NoSQL databases in handling medical images, MongoDB is considered in this work. A Hybrid data model is essential which will effectively handle huge medical images. This paper aims proposing a hybrid data model for sharing Medical Images in the cloud, using distributed sharding environment. Medical images can be shared using the Cloud and it is a necessity to have a Non-Relational based storage to handle medical images in the Cloud environment. MongoDB is a Document Database is much suitable to store information in the cloud. The concept of sharding supported by MongoDB allows partitioning the huge medical image into chunks and move to a distributed environment.[2,3] is the objective study of this paper.

2.2 Robustness of NoSQL Databases

The requirement of a NoSQL database to handle Medical Images in DICOM format is inevitable. The salient features NoSQL Databases is the way the NoSQL database differ from a traditional RDBMS. NoSQL databases are better in handling unstructured data. They differ in Data model, Architecture, data distribution and also in performance.

- Data model – A NoSQL database has a flexible schema whereas the Data Model of RDBMS follows a rigid Schema and it can handle only structured data. A NoSQL Database is capable of handling all types of data, structured,semi-structured and Unstructured.
- Architecture – A NoSQL system can operate in a distributed, scale-out design whereas RDBMS's are architected in a centralized way.
- Data distribution model – A NoSQL database allows data to be distributed evenly to all nodes making up a database cluster and enables both reads and writes on all machines whereas it is difficult to distribute data to the clients as it in works in a a centralized fashion. In NoSQL it is the distributed model enables to parallel process huge data.
- Scaling and Performance model – A NoSQL database scales horizontally based on the load by adding extra nodes that deliver increased performance in a linear manner wheras an RDBMS typically scales vertically by adding extra CPU, RAM, etc., to a centralized machine.[2]

3. HYBRID DATA MODEL FOR MEDICAL IMAGE SHARING

As mentioned before, medical imaging plays a vital role in both decision making and treatment support. Sharing of Medical images with shorter latency time to access the images is needed to have the best quality health care service.

3.1 Medical Images storage structures

The DICOM standard is capable of integrating almost every all modern imaging equipments, networking servers, accessories and picture archiving and communication systems (PACS) from different manufacturers[1]. A DICOM image file is Digital Imaging and Communications in Medicine standard. To be more specific, image files that are compliant with part 10 of the DICOM standard are referred as “DICOM format files” or simply “DICOM files” and have the extension “.dcm.”[16]. Due to this ease of integration this communication standard has become a nearly universal level of acceptance among vendors of radiological equipment.

3.2 Parts of a DICOM file.

The Digital Imaging and Communications in Medicine (DICOM) standard adopts files as individual, self-contained repositories for the storage of a mixed of alphanumerical and binary content regarding radiological images.

The digital imaging and communications in medicine (DICOM) protocol is the one default standard for image data management in healthcare. The DICOM file contains two parts stored as a single object i) A header that stores Meta data ii)Image data stored as pixels. The header stores details about the patient, acquisition parameters for the imaging study. It also stores image dimensions, matrix size, color space, and a host of additional non intensity information required by the computer to correctly display the image. The header is followed by the image data stored as a long series of 0s and 1s, which can be reconstructed as the image by using the information from the header. Fig 3.1 shows a sample DICOM file.

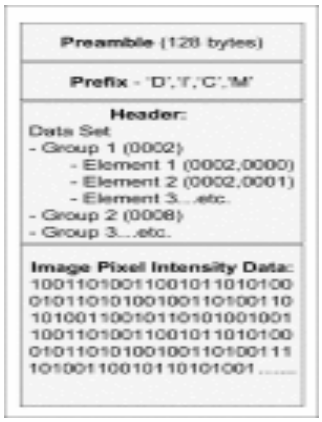

Fig 1. Structure of a DICOM image file

The process of medical diagnosis relies on the technological capabilities of medical imaging and image analysis. The diagnosis by the physicians relies on the accuracy and conclusions drawn from the medical images. The final medical prescription depends on the various capabilities of the medical imaging in computer systems that aids in medical diagnosis [7].

3.3 Medical Image Sharing

Today the Medical diagnosis happens by sharing the medical images. The most important challenge in implementing a sharing system for medical diagnosis using medical imaging is to consider and to choose the right data storage technology. The development of information technology has given different solutions in handling images. The methods have changed from time to time. The various methods are discussed below.

3.3.1 File Systems

Initially, images were stored in files outside databases and inside databases only their paths were collected. This was referred as file systems. Usually, groups of DICOM files are hierarchically organized in studies and series, physically disposed into file system directory trees. Despite its simplicity in storing content, ordinary file systems do not provide index

capabilities allowing searches by content – restricting access by directory names and file names.

3.3.2 RDBMS

To surpass above limitation, Picture Archiving and Communication Systems (PACSs) often adopt Relational Database Management Systems (RDBMSs) as metadata repositories, benefiting from its general-purposed index structures

Then BLOB (Binary Large Object)- a new type of data was developed and introduced which allowed the possibility of image storage in RDBMS. Even though, Relational Databases are the most popular technology for data storage, the accepted fact being that the BLOB is not the best solution for binary data storage. SQL is highly incapable of handling binary content. It is not possible to access binary content from the SQL.

3.3.3 NoSQL

The NoSQL(Not Only SQL) databases which are non-relational in nature can handle the multimedia content with ease. As there is a substantial growth of multimedia data in the form of binary, it is essential to look into other non-relational solutions to handle images and medical images in specific. So the use of NoSQL in handling Medical images for the many reasons discussed in Section II is to be considered. The reason primarily is the i) the ability to handle binary content as the native data format of NoSQL databases have JSON as their storage format. ii) They are capable of handling huge data files as they are scalable. This approach has gained general name of NoSQL approach

3.4 Hybrid Data model

Medical images in DICOM format has to be stored and also be shared. Sharing of medical images leads to transfer of large amounts of data shared across the network. This may lead network bottlenecks and congestion. Due to this there is an increase in the latency time. This can be avoided and it is possible to have better bandwidth utilization. A data model using MongoDB is presented here. This transfers medical images using a distributed –sharding environment. It is possible to distribute and parallelly process the huge data through sharding.

3.4.1 Methodology

A Single machine cannot hold huge data, whereas a cluster of inexpensive hardware can be leveraged to hold huge amounts data. The data can be stored and processed effectively and efficiently. Three key goals emerged to achieve this:

Data needs to be stored in a networked file system that can be stored in multiple machines, rather than a centralized system as in RDBMS. Huge files can be chunked and stored in multiple nodes.

Data needs to be stored in a schema free structure or it should possible to change schemas without much alteration.

Data needs to be processed in a way that computations on it can be allowed to be performed as isolated subsets and then combine to generate the desired output.[15]

Data needs to be stored in a networked file system that can be stored in multiple machines, rather than a centralized system as in RDBMS. Huge files can be chunked and stored in multiple nodes.

Data needs to be stored in a schema free structure or it should possible to change schemas without much alteration.

Data needs to be processed in a way that computations on it can be allowed to be performed as isolated subsets and then combine to generate the desired output.[15]

MongoDB has the ability to shard and distribute, parallel process it.

3.4.2 Sharding

The process of splitting data up and storing the different portions of the data on different machines is called sharding; we can also use the term partitioning to describe this concept. It is possible to handle more loads without using powerful servers by just splitting of data and storing it up across many machines. It is possible to handle huge files without requiring large or powerful machines.

A single server's capacity is challenged while handling large data sets or huge sized Medical images. These applications that handle huge medical image data can be categorized into Big Data needs demands high throughput. Huge data which is larger than the system's RAM stress the I/O capacity of disk drives. A Shard is a computer connected to a cluster of machines used in the Sharding process.

There are two types of Sharding methods i) Manual sharding and ii) Automatic Sharding

3.4.2.1 Manual Sharding

Manual sharding is when the application connects to different independent servers. The sharding process is taken care of using the application code which manages the sharding process of storing the data in different servers and getting it back by querying against the appropriate server. This approach becomes difficult to maintain when nodes are added or removed from the database cluster in maintain the load patterns.

3.4.2.2 Auto Sharding

Auto sharding is the process where the data gets evenly distributed across the shards or the computers connected to the sharding environment. The data is chunked and sent across. The balancer put approximately same number of chunks into each shard/system connected to the sharding cluster.

4. MONGODB AND SHARDING

MongoDB supports autosharding, which helps in eliminating the administrative overhead involved in manual sharding. As mentioned earlier the sharding cluster manages the splitting up of data and rebalancing it automatically. MongoDB sharding can be used to support applications with very huge data sets which needs to have high throughput operations with minimum latency.[3]

4.1 Autosharding in MongoDB

MongoDB performs autosharding by breaking up the data stored in collections into smaller chunks. A cluster of computers can be connected to the sharding environment and the broken up chunks can be distributed across shards evenly. Each shard contains/stores in a subset of the total data set. A routing process called mongos stores detail about where all of the data is located, to keep things anonymous to the application. The applications connect to the router and gets information regarding the meta Data. The router, knows what data is on which shard, is able to forward the requests to the appropriate shard(s).Fig 4.2 Shows the sharding process where 3 Shards are connected to the router/Mongos. When the client needs to send data across, the data is evenly distributed

and sent. Here the sharding process is abstracted from the application. Sharding can be used only when there is a need to handle large objects with ease, which improves performance.

Fig 4.1 shows a Non-sharded MongoDB setup; where a client connects to a mongod process. Here there is no cluster of machines wherein huge files cannot be handled with ease.

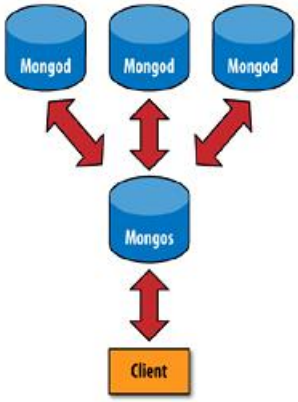

Fig 2

A Non-sharded MongoDB setup is shown in Fig 3; where a client connects to a mongod process. Here there is no cluster of machines wherein huge files cannot be handled with ease. The latency time increases as the sharing is directly uploaded to the network for sharing. This method fails in handling huge medical images.

Fig 3

4.1 Setting up a Sharding environment

Sharding basically involves three different components working together:

shard
A shard is a container that holds a subset of a collection's data. Thus, even if there are many servers in a shard, there is only one master, and all of the servers contain the same data.

mongos
This is the router process and comes with all MongoDB distributions. It basically just routes requests and aggregates responses. It doesn't store any data or configuration information.

Config Server
Config servers store the configuration of the cluster: which data is on which shard.Because mongos doesn't store anything permanently, it needs somewhere to get the shard configuration. It syncs this data from the config servers.

5. EXPERIMENTAL SETUP

A Sharding environment was setup using 7 systems with Ubutu Operating System and MongoDB 3.03. The machines had the configuration,, 6th Generation Intel(R) Core(TM) i5-6200U Processor (3M Cache, up to 2.80 GHz). The set up is as given below.

We set one config server, One Query Router and 3 shards. One system was a Client and other one was a server.

We studied the time complexity of sharing or storing DICOM image files from a client machine to a server.

First the Config Server was set and then the Query router. Then the shards were added one by one to the config server.

Sharding was enabled in the database level and the time complexity was studied for huge DICOM files. The file sizes varied from 1 GB to 5 GB. The study was carried carried out for sharing data from a Client to a server machine through shards.

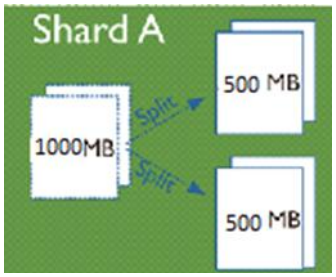

Fig 4 .Splitting/Chunking of Huge DICOM files

6. RESULT

6.1 Time Complexity with Sharding

We try to share the medical images in a sharded and non-sharded environment. The time was recorded in Sharding and a Non-sharding environment. The latency time in a non-sharded environment is much higher than the latency in sharded environment. The results of the Non-sharded environment are shown in Table 1.

The study was also carried out in a sharded environment. The results indicate that the latency time decreased as we increased the number of shards. The time taken to store was much higher with a two shards and was very less with 3 or more shards.

Table-1 Time in a Non-Sharded Environment

Size(MB)	Three shards(Mins)	N0 Shard(Mins)
1000	1.5	3.1
2000	2.14	13.3
3000	2.8	19.1
4000	3.1	24.4
5000	3.8	35.2

Table-2 Two and Three Shards comparison

Size(MB)	Three shards	Two Shards
1000	1.5	1.8
2000	2.14	5.21
3000	2.8	8.16
4000	3.1	12.23
5000	3.8	16.8

The Following is the Graphical representation of the same.

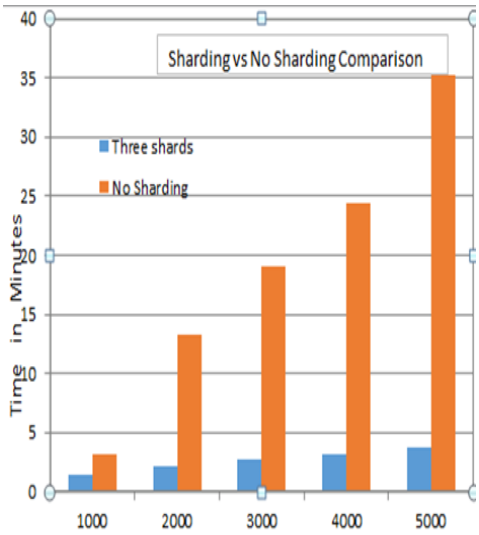

Fig 5. Sharding vs NoSharding

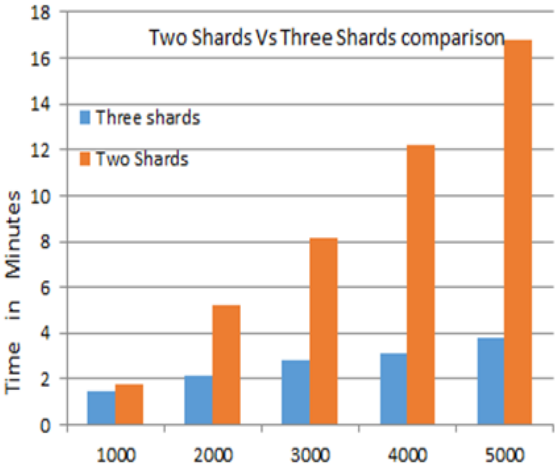

Fig 5 Three Shards vs Two Shards

7. CONCLUSION AND FUTURE WORK

7.1 Conclusion

The study shows the effect of parallel processing and there is a great reduction in time as the number of machines used to distribute the data is increased. The time taken to share the image reduces with Data distribution using Sharding. The time gets reduced with more number of Shards. More the data gets distributed, it takes lesser time. This clearly indicates that a huge data can be shared in a sharded environment with ease. The main challenge in medical images was in handling huge images, sharing, storing and retrieval. The degradation of the performance as the size increased can be easily overcome with this model. Health care departments using telemedicine and sharing of medical images can be highly benefited with this model. This can be extended to any radiological department which involves sharing of Medical Images.

7.2 Future Work

This Data model is suitable for medical Image processing in the Cloud environment. Health care Informatics data grows day by day. The amount of data is huge and the Health care providers are in a verge to move Health care information and medical images to the cloud. The movement of medical images to the cloud needs a specific data model where large amounts of data can be shared and processed without much network bottlenecks. There is also a huge need to process and analyze the huge volumes of data stored in the Cloud. This model will be highly suitable to share and analyze huge sized images and health informatics data stored in the Cloud. In future, it is possible to develop a model to move medical images to the cloud using this method.

8. ACKNOWLEDGMENTS

The authors wish to thank Gautham, Manikanda and Yogesh MCA students of REVA University, Bangalore, for their contribution to this work.

9. REFERENCES

[1] Oleg S. P ianykh,"Digital Imaging and Communications in Medicine (DlCOM), A Practical Introduction and Survival Guide ", book published by Springer-Verlag Berlin Heidelberg, pp 247-261, 2008 and 2012

[2] Yimeng Liu, Yizhi Wang,Yi Jin, Research on The Improvement of MongoDB Auto-Sharding in Cloud Environment, IEEE, 978-1-4673-0242-5-2012

[3] Kristina Chodrow, Michael Dirolf, Scaling MongoDB.

[4] Alexandre Savaris, Theo Härder, Aldo von Wangenheim, DCMDSM:A DICOM decomposed storage model, Journal of the American Medical Informatics Association · February 2014

[5] Alexandre Savaris, Gabriela Bussolo Colonetti, Rodrigo Rodrigues Pires de Mello, Aldo von Wangenheim Relational Databases versus Search Engines: A Performance Comparison for Storing and Querying DICOM Metadata

[6] D.Revina Rebecca, I.Elizabeth Shanthi,A NoSQL Solution to efficient storage and retrieval of Medical Images,International Journal of Scientific & Engineering Research, Volume 7, Issue 2, February-2016,ISSN 2229-5518

[7] Liliana BYCZKOWSKA-LIPIŃSKA, Agnieszka WOSIAK, Multimedia NoSQL database solutions in the medical imaging data analysis

[8] Simón J. Rascovsky, MD, MSc • Jorge A. Delgado, MD • Alexander Sanz, BS • Víctor D. Calvo, BS • Gabriel Castrillón, BS,Use of CouchDB for Document-based Storage of DICOM Objects

[9] Luís A. Bastião Silva, Louis Beroud, Carlos Costa and José Luis Oliveira,Medical imaging archiving: a comparison between several NoSQL,978-1-4799-2131-7/14/$31.00 ©2014 IEEE.

[10] Luan Henrique Santos Simões de Almeidaa, Marcelo Costa Oliveiraa, A Medical Image Backup Architecture Based on a NoSQL Database and Cloud Computing Services, MEDINFO 2015: eHealth-enabled Health,

doi:10.3233/978-1-61499-564-7-929.

[11] Marcosa E., Acuna C.J., Vela B., Caveroa J. M., Hermandez J.A.: A database for medical image management, Computer methods and programs in biomedicine, vol. 86, pp: 255-269, 2007 Elsevier Ireland Ltd

[12] D.Revina Rebecca, I.Elizabeth Shanthi , Analysing the suitability of storing Medical Images in NoSQL Databases, International Journal of Scientific & Engineering Research, Volume 7, Issue 6, June-2016,ISSN 2229-5518

[13] http://www.siemens.com/innovation/en/home/pictures-of-the-future/health-and-well-being/medical-imaging-facts-and-forecasts.html

[14] Yan Hu, Fangjie Lu, Israr Khan, Guohua Bai, A Cloud Computing Solution for Sharing Healthcare Information

[15] D.Revina Rebecca et al, Impact of adapting Cloud Computing in health care industry for storing medical Images.

[16] Dandu Ravi Varma, Managing DICOM images: Tips and tricks for the radiologist, Indian J Radiol Imaging. 2012 Jan-Mar; 22(1): 4–13, doi: 10.4103/0971-3026.95396.