Analysing the suitability of storing Medical Images in NoSQL Databases

D.Revina Rebecca, Dr.I.Elizabeth Shanthi

Abstract— The need for storing semi-structured and unstructured data has led to the rise of new kind of databases called NoSQL databases. This is due to the need of the storage needs of todays data which is schema less. NoSQL databases are very much suiting the needs of different aspects of storage and retrieval. The IT industry is going through a paradigm shift, where the entire scenario of storing and retrieval of information is moving towards NoSQL Databases. The medical industry has no exception, where the health care records mainly the medical images need a better data model for storage and retrieval. The need of the hour is to find a better suiting NoSQL `Database. This paper aims in studying the different NoSQL databases in the light of medical images.

Index Terms— MongoDB, Cassandra, Chunked Storage, Cloud Computing, Medical images, NoSQL databases.

1 INTRODUCTION

Storage and retrieval of medical images is very crucial in medical informatics. There is a dramatic change in the methodology of data storage in the past decade in every area. Healthcare informatics is no exception. Medical images are part and parcel of health informatics where each therapeutic procedure has at least one medical image involved. Medical imaging technologies are very important and it plays a vital role in medical diagnostics and therapeutics. So due to these experts predict a substantial increase in the volume of Medical images stored in the near future. It is estimated that in the future, 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]. Medical imaging systems can be classified as either basic modalities such as general X-ray, mammographic X-ray, and ultrasound, or advanced modalities such as computed tomography, magnetic resonance (MR), and molecular imaging.it is expected to reduce the complexity and cost of storage of medical images.[4] Having this in mind, it is necessary to find a better solution which will be helpful in effective archiving of the medical images. The volume of medical images stored has exceeded 1 exabyte mark today , which takes medical imaging into Big Data territory.[5]"

Different technologies are evolving and gaining acceptance to support the handling of large amount of data. This change, where medical images are slowly coming under the big data category.[6,7]. NoSQL (Not only SQL) is one such where the change in the storage needs led to the rise of NoSQL (Not only SQL) Databases. These are open source databases. databases, such as Cassandra,HBase, MongoDB, Redis. These open source databases are capable of handling large amount of unstructured data and semi structured data. It also provides high performance. These databases are being rapidly adopted by industry and research fraternity. Medical Images which comes under unstructured data, can easily be stored using NoSQL databases. In this paper we compare the performances of two NoSQL databases, MongoDB and Cassandra with respect to storing of medical images.

This paper is structured as follows: In Section II the related work in the area of storage and retrieval of images using NoSQL databases is discussed. In section III we present the need for NoSQL databases and the advantages of storage of images using NoSQL. We also discuss the two types of NoSQL databases and their various functionalities. Further in section IV we present the details of our implementation in MongoDB and Cassandra. We present performance graphs. Finally section V presents our conclusions and future work.

2 Literature Survey

NoSQL is becoming adopted by the industry. It has been compared with its RDBMS counterparts under various scientific and research contexts [9,13] .

A CouchDB based medical archiving system was developed Rascovsky et al. The authors suggest that document databases are highly suitable to store, retrieve and query DICOM files. Also the metadata of DICOMM images can be better stored in Document databases.[10,13]

In [13] Luís A. Bastião Silva et al, compared MongoDB and CouchDB in storing and retrieving medical images, where the performances of MongoDB was better than CouchDB.

In our previous work we had compared the storage and retrieval of Medical images in MYSQL and MONGODB[11]. It was proved in [11] that the performance of MongoDB was better than MySQL . Also NoSQL is better and well suited for storing unstructured . Medical images are stored using PACS which is a RDBMS based solution. In [10,11] the various disadvantages of storing medical images in a RDBMS based archive is discussed. Also in our previous work, MongoDB, a NoSQL based Document database was proved better. So we decided to compare the various NoSQL databases in the context of medical images.

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

The NoSQL databases are very much suitable for storing images of larger size. The time has come where the medical image providers are moving to the cloud for their storage needs[15]. Cloud based medical image processing systems are picking up. So far the whole scenario of medical imaging is based on RDBMS[10,11] which would be a bad fit for the cloud. [16,17]. It is needed to find a better NoSQL database which will effectively store Medical Images. So in this paper we compare the performance of two NoSQL databases.

3 NoSQL Databases to store Medical images

NoSQL databases are non relational databases which do not follow a strict schema. They follow a very different data model. NoSQL databases have several advantages where it has a more flexible data model. Unlike RDBMS where rigid schema is required, NoSQL databases can store any type of data that includes data that can be either structured, semi-structured, and unstructured data.

NoSQL Databases are advantageous over RDBMS i) The data is available with redundancy across one or more locations. ii) It can run over multiple data centers and its cloud enabled. iii) it has very good write speed and low latency query speed iv) Supports scale-out architecture where it is possible to add more processing power and storage capacity can be increased. It is highly scalable.

It is difficult for a Single machine to hold huge data, whereas a cluster of inexpensive hardware can be leveraged to hold huge amounts data. This data has to be stored and processed effectively and efficiently. Three key goals emerged to achieve this:

* Data need to be stored in a networked file system which can be moved to many machines, rather than a centralized system as in RDBMS. Huge files can be chunked and stored in multiple nodes.
    * Data should be stored in a schema free structure or it should possible to change schemas without much alteration.
    * Data need to be processed in a way that computations on it can be performed as isolated subsets and then combine to generate the desired output.
    By moving Computation to data it is possible to stop the movement of huge amounts of data being transferred across the network and leads to better bandwidth utilization.

To achieve this several NoSQL databases evolved. There are different categories of NoSQL databases viz. i)Key-Value Databases ii) Columnar databases iii) Document databases. iv) Graph Databases. There are many popular NoSQL databases in each type. In these four types, the existing related work is based on Column databases and Document databases. So we have compared the efficiency of storage and retrieval of medical images with one column database and Document Database.

3.1 Column Databases

Column databases have existed in many forms, but they made a re-entry to the IT world by Google. Google came up with Big Table. Google needed a schema-less Database to store unstructured and semi-structured data in large scale that helps in Google's search engine, other big data efforts by Google. All Google's products deal with huge data and Column family stores are versatile in storing huge data.

Column-oriented databases are very much widely used among the non-relational databases. It started with Google and many social networking companies like Facebook, LinkedIn, and Twitter developed their own version of it which started the NoSQL revolution .

Column Databases in storing Medical Images

Schema definitions are not very strict in a Column-oriented database where it can accommodate new columns as needed. In a column-oriented store, a column-family is a set of columns grouped together and Columns in a column-family are logically related to each other. A column-family is pre-defined, but not the columns. Relevant columns are added when it enters the system. A Column-family members will be physically stored together which aids in faster retrieval when we need to read data with similar characteristics in the same family.

In RDBMS we need to define the type of data which the Columns can store. It is not so in Column-families, no limitation; where a column database can contain any number of columns, which can store any type of data. The data should be persisted as an array of bytes. Null values are not stored at all.

Column databases can be thought of RDBMS tables with special properties, but with a difference, a column database can easily scale out .[3]

Cassandra – A Column Database

Cassandra is a distributed storage system for developed by Facebook. Cassandra is used by the largest social networking platform Facebook, that serves hundreds of millions users at peak times using tens of thousands of servers located in many data centers around the world. The database system should be reliable, efficient and support continuous growth. It is needed for the platform needs to be highly scalable.

Cassandra meets the reliability and scalability needs described. [4]

Cassandra system was designed to run on cheap hardware and handle high write throughput while not sacrificing read efficiency.

Managing very large amounts of structured data spread out across many commodity servers, and to provide highly available service, is done by Cassandra.[18]

There are many key characteristics of Cassandra that makes it Ideal for many Modern Online Applications [3]

• Massively scalable architecture - Cassandra has a master less peer to peer design where every node is the same as the other, which provides operational simplicity and easy scale out capabilities.

All Cassandra nodes are active everywhere, Irrespective of their Location the nodes may be written to and read. from no matter where they are located.

• Cassandra has a performance which is scales Linear. Additional nodes increases performance that is an addition one node doubles the performance. If one node can do 100K transactions/sec, Two nodes will deliver 200K transactions/sec, and eight nodes, 800K transactions/sec.[21]

• Cassandra offers Continuous availability, where both func-

tions and data are replicated and there is no single point of failure.
• Failed nodes can be detected and recovered easily using the peer to peer mechanism.
• Cassandra supports a very flexible and dynamic data model. It can store any type of data from structured to unstructured data.
• Transactions are eventually consistent.
• It provides good cross and Multi-data center replication – It also excels in Multi-cloud availability for writes/reads.
• Data is compressed up to 80% without much performance overhead and thereby storage costs are reduced..
Since Cassandra has high throughput and can handle images, we consider Cassandra to store our medical images.

3.2 Document Databases[4]

Document databases store data as documents. These documents are stored as an object and uses a structure similar to JSON(JavaScript Object Notation).Each JSON document is an Object. A set of records can be aggregated into a single document/object in a document database. This helps in building a flexible data model. It also aids in efficiently distributing the aggregated documents with high read and write performance. Even Document databases do not need strict schema definitions. This is helpful in modeling unstructured and polymorphic data.
Most document stores group documents together in collections. [7]. Document collections can be used in many ways to manage large document stores. They can serve as ways to navigate document hierarchies, logically group similar documents, and store business rules such as permissions, indexes, and triggers. Collections can contain other collections. Flexibility comes with using a document store: allowing collections to have collections.

MongoDB- A Document Database

MongoDB is a document Database; The documents are grouped together as collections. Collections are similar to relational tables.
MongoDB uses a BSON format to store documents. BSON is a binary way of JSON-type representation. Here the structure is similar to a nested set of key/value pairs. BSON supports more data types like regular expression, binary data, and date. BSON is helps in storing and exchanging data; Also BSON helps in describing the contents in a given document. Due to this it is not needed to specify the structure of the document in advance. JSON can be regarded schema less as the documents can be updated individually or changed independently of any other documents. The performance of MongoDB is enhanced due to BSON. It even makes the processing and searching faster. BSON stores data in Binary and all objects in BSON is a set of key/value pairs. Each document in MongoDB is identified using a unique identifier called the _id key.
Since MongoDB can handle Binary data and its effectiveness in storing images[11], we have considered MongoDB for storage of medical images.

4 STORAGE OF MEDICAL IMAGES IN CASSANDRA AND MONGODB

Digital Medical Images are obtained from different modalities and devices like Color flow Doppler, Computed radiography, Computed tomography, Digital Subtraction Angiography, Digital Radiography, Mammography, Magnetic Resonance, MRI, Nuclear Medicine, Positron Emission Tomography – PET, Ultrasound, X-Ray, etc. They are stored in DICOM (Digital Imaging and Communications in Medicine) format. DICOM is a software integration standard used in Medical Imaging. All modern medical imaging systems (AKA Imaging Modalities) Equipment like X-Rays, Ultrasounds, CT (Computed Tomography), and MRI (Magnetic Resonance Imaging) support DICOM and use it extensively.[19,20] These images are very huge. The size of an MRI with high-resolution is more than 100MB. These DICOM images also store demographic data of the images. [20] The storage of these images and retrieval is becoming a challenge. Also the health care informatics are slowly moving to the cloud environment, where the storage and maintenance are taken care by the cloud providers.[14,15] As discussed in [1,11], RDBMS based storage of medical images will be a bad fit. To overcome the disadvantages of RDBMS, Two NoSQL databases, MongoDB and Cassandra are used to store the Medical images.

4.1 Experimental Setup

The test environment had following configuration: The machine was running on Windows – operating system version 10 Pro 64-bit. Processor: Intel Core i5-4670K CPU @ 3.40 GHz, 4 Cores, 4 Logical CPUs, Memory: 8 GB RAM. The MongoDB version was 3.2 and Cassandra version was 2.6. The study was done using JAVA. The images were stored in Cassandra and in MongoDB. In both Cassandra and MongoDB the image is divided into smaller parts or chunks and stored.The Storage and retrieval times of images of sizes ranging from 5 MB to 100 MB was taken for this study. The time complexity for storing and retrieving medical images in both MongoDB and Cassandra was recorded and the results are shown in Fig. 1 and Fig. 2.

4.2 Chunked storage in Cassandra

Storing large images in Cassandra with single set operation is difficult, as it creates heap pressure and hotspots. To store a file under a single key and column creates performance bottlenecks as the streaming capability is designed around smaller objects. To overcome this limitation, Cassandra allows large objects to be broken into small parts and storing the parts across multiple columns. The chunk size can be specified in bytes.This can be done by using the utility Astyanax which splits up large objects into multiple keys and can fetch them parallel. A basic cassandra chunked provider is provided with Astyanax. .[22,23] First a chunked provider is created

To create a chunked provider.

```
ChunkedStorageProvider provider
  = new CassandraChunkedStorageProvider(
       keyspace,
       "data_column_family_name");
```

To store an object

The ObjectWriter will break up the file into chunks and push them to cassandra from multiple threads

```
ObjectMetadata meta = ChunkedStorage.newWriter(provider, objName, someInputStream)
   .withChunkSize(0x1000)
   .withConcurrencyLevel(8)
   .withTtl(60)
   .call();
```

To Read an Object

The file is read directly into an OutputStream. The ObjectReader handles parallelizing and randomizing the requests in batches.

```
ObjectMetadata meta = ChunkedStorage.newInfoReader(provider, objName).call();
ByteArrayOutputStream os = new ByteArrayOutputStream(meta.getObjectSize().intValue());
meta = ChunkedStorage.newReader(provider, objName, os)
   .withBatchSize(11)
   .withConcurrencyLevel(2)
   .call();
```

4.3 Chunked storage in MongoDB

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 images. 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. GridFS can easily handle files that exceed 16 MB.GridFS allows large binary files to be chunked by breaking the files into smaller files called “chunks” and stores in MongoDB. Each chunk can be handled independently. GridFS uses two collections to save a file to a database. One collection stores the file chunks, and the other stores file metadata.

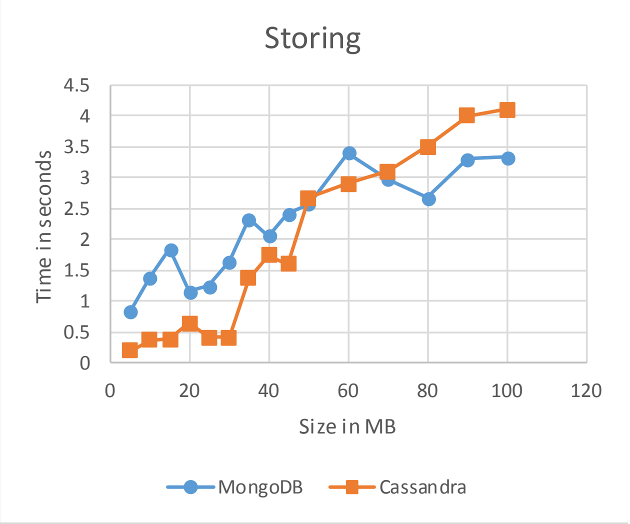

Fig-1

5 CONCLUSION AND FUTURE WORK

The result indicates that the time complexity of Cassandra is less when compared with MongoDB for smaller files. But as the file size increases time complexity of MongoDB remains constant comparatively, so for larger file MongoDB seems to be better candidate. In Cassandra the time increases proportionally with size of the file. We propose both MongoDB and Cassandra may be suitable to store Large Medical images. But MongoDB will be a better candidate.

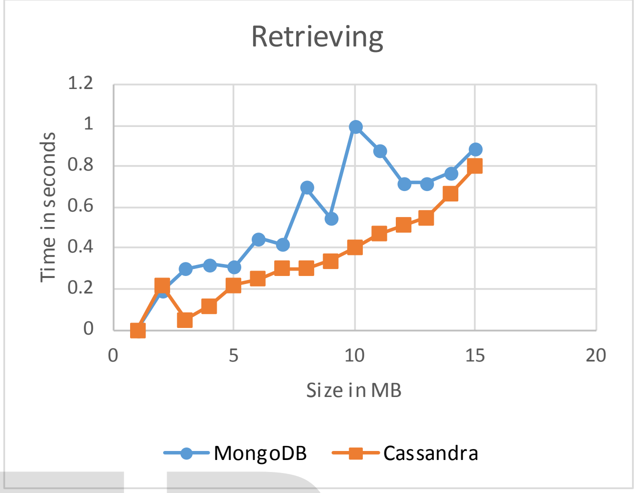

Fig-2

Since Medical images are moved the cloud and NoSQL bases are highly suitable for storing data in the cloud, In future we propose to study storing of Medical images using MongoDB database in the distributed cloud environment.

Acknowledgment

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

REFERENCES

[1] Steve G. Langer, Challenges for Data Storage in Medical Imaging Research, Volume 24, Issue 2, pp 203-207, April 2011, Journal of Digital Imaging.

[2] Luís A. Bastião Silva • Carlos Costa, José Luis Oliveira DICOM Relay over the cloud, Springer, 2013

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

[4] Shashank tiwari, professional Nosql, John Wiley & Sons, Inc. 2011.

[5] http://www.frost.com/prod/servlet/press-release.pag?docid=268728701, Frost & Sullivan: U.S. Medical Imaging Informatics Industry Reconnects with Growth in the Enterprise Image Archiving Market

[6] N. V. Chawla and D. A. Davis, "Bringing big data to personalized healthcare: A patient-centered framework," Journal of general internal medicine, *vol. 28, pp. 660-665, 2013*

[7] Dan McCreary,Ann Kelly, Making Sense of NoSQL Databases, Manning Publications,2014.

[8] N. Leavitt, "Will NoSQL databases live up to their promise?," Computer, vol. 43, pp. 12-14, 2010

[9] Rajat Aghi, Sumeet Mehta, Rahul Chauhan, Siddhant Chaudhary and Navdeep Bohra,A comprehensive comparison of SQL and MongoDB databases,International Journal of Scientific and Research Publications, Volume 5, Issue 2, February 2015,ISSN 2250-3153

[10] 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

[11] 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

[12] D.Revina Rebecca et al, Impact of Adapting Cloud Computing in Health care Inductry for storing Medical Images, NCEIT (2014)ISBN:978-81-923796-5-4

[13] 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

[14] CARE PROVIDERS ASCEND TO CLOUD FOR MEDICAL IMAGING NEEDS,2016,http://media.techtarget.com/digitalguide/images/Misc/EA-Marketing/Eguides/Healthcare_Cloud_Storage.pdf

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

[16] J.Antony John Prabu, Dr.S Britto Ramesh Kumar,Issues and Challenges of Data Transaction Management in Cloud Environment

[17] Katarina Grolinger1, Wilson A Higashino, Abhinav Tiwari and Miriam AM Capretz,Data management in cloud environments: NoSQL and NewSQL data stores,journal of cloud Computing, Springer Open journal, 2013

[18] Avinash Lakshman, Prashant Malik, Cassandra - A Decentralized Structured Storage System

[19] Luís A. Bastião Silva, Carlos Costa, Augusto Silva and José Luís Oliveira, A PACS Gateway to the Cloud

[20] http://www.dicomlibrary.com/dicom/study-structure/

[21] http://docs.datastax.com/en/cassandra/2.1/cassandra/gettingStartedCassandraIntro.html

[22] https://github.com/Netflix/astyanax/wiki/Chunked-Object-Store

[23] Edward Capriolo, Cassandra High Performance Cook Book, Copyright © 2011 Packt Publishing