Data Migration from Relational Database to MongoDB

Ajit Singh[1]

[1] Patna Womens College

Abstract

MongoDB is a document-oriented database which helps us group data more logically. This paper demonstrates the conversion of data from a native tabular form to unstructured documents. The document and collections within it needs not to be well defined prior to the creation of unstructured data in MongoDB. The MongoDB has lots of extensive built-in-features and is highly compatible with other software systems, with extensive and flexible ways of accessing data beyond JSON query, its highly compatible Business Intelligence Connector is highly compatible which makes it compatible with existing databases. High scalability is making it remarkable and popular in the World and hence made me think about writing a paper demonstrating the data conversion. This conversion has helped me in making the most of modern data to be compatible with MongoDB. Data is stored on the cloud as cloud-based storage is an excellent and most cost-effective solution. My solution is highly scalable as the built-in shading solution for data handling makes it one of the best big data handling tool. The data that i have used, is location based in MongoDB that can directly yeild document ACID transactions to maintain data integrity.

Index terms— data migration, relational database, MongoDB, XAMPP, NoSQL, ACID transactions.

1 Introduction

Database is usually defined as collection of data and the system that handles data, transactions, problems and issues of the database is known as Database Management System (DBMS). A Database was incepted in 1960 in order to satisfy the need of storing and finding data. It began with navigational databases which were based on linked-lists, moved on to relational databases with joins, and then afterwards object-oriented and then without joins in the with NoSQL (MongoDB, Cassendra, HBase/Hadoop, CouchDB, Hypertable etc.), emerged and has become a popular trend [1] in the late 2000s. On one hand, there is relational database management system (RDBMS) that offers more consistency as well as powerful query capabilities and a lot of knowledge and expertise over the years [2]. On the other hand, NoSQL approach, which offers higher scalability i.e. it can run faster and support bigger loads.

The NoSQL in general doesn't support complicated queries and doesn't have a structured schema. It recommends de-normalization and is designed to be distributed (cloud-scale). Because of the distributed model, any server can answer any query. Servers communicate amongst themselves at their own pace. The server that answers our query might not have the latest data. This research paper is based on data conversion using Mongo DB. I have demonstrated data conversion from traditional database to MongoDB document-based database. Section II covers the related work done in this field. In section III, the performance of the proposed concept has been verified with the aid of XAMPP and MongoDB management system. Results are followed up in Section IV presents the conclusion.

2 NoSQL characteristics

3 II.

4 Related Work

This research work is based on MongoDB in which the relations within the applications are well defined by two main tools which are references and embedded documents. The references store the relationships between data by including links or references from one document to another. The applications can resolve these references to access the related data. Having the URL link within the field definitely adds to its range of accessing data. The embedded documents capture the relationships between data by storing related data in a single document structure. The MongoDB documents make it possible to embed document structures in a field or array within a document along with a new feature.

The MongoDB is derived from the word humongous as the database has a characteristic of exceptional scale-up capability. So, big data are huge massive exponentially changing and growing data sets [3]. The traditional framework was not able to handle and analyze large amount of data. Also the internet of things (IoT) is creating exponential growth in data. The Big data and linked data are two faces of the same coin both representing the integral and concentric part of web based world. Each data element is connected via a URLs and are identifiable, locatable and accessible. Stronger the bond between the two, higher will be the unsustainability and big data utility. Semantically well structured, interconnected and linkied data will add intelligence to the data. The main goal is to make data more interactive, participatory and innovative. Making sense and extracting information from the data for organizational benefits. So, semantic technologies can help us extracting the main value from the data. Test analytics aids in the conversion of unstructured text to structured meaningful text by deriving and extracting patterns.

Database scaling is a highly difficult task. The modern applications require high scalability and data curating capacity. I have used XAMPP which is apache distribution web server solution for creating a local web server, after setting the environment for the database was tested. The JSON enables the transferring data between the master server and the web application in human readable form. This improved the overall efficiency.

5 III.

6 Proposed Work

The key benefits of NoSQL are speed, scalability, price, flexibility and simplicity. The main characteristics is its Non-adherence to relational database concept. As an example, Grolinger et al [4] identified the difficulties in handling information which can be the huge Map Reduce. Naheman and Wei [5] studied and compaired various BigData tools like HBase and other NoSQL databases eg Bigtable, Cassandra, CouchDB, and MongoDB etc. Laurence [6] worked on a virtualization system which does allow us to enquire and join information making use of a sql query and the result in API that is underlying to MongoDB.

Hadoop -the main frame technology being used is a Java based framework that supports the processing and storage of tremendously large data sets in a distributed computing environment. Even though, as we all we know that Hadoop is written in Java programming language, programs for Hadoop can be written in other languages like Python. Mostly, Python code is translated to Java jar files using Jython. The Hadoop has two major areas of concern. The first one is Hadoop which is built using Java and hence the application developers should know Java to develop the framework and to develop the map reduce technologies. The Hadoop provides a framework by which Map Reduce applications can be built using python. The second is the co-sister technologies which work on top of this framework is one of the example being Cassandra, Cassandra being a NoSQL database technology which is ideal for high-speed online transactional data, while as Hadoop being a big data analytic system.

Apache Spark once a component of the Hadoop ecosystem is now fetching big data platform of choice for enterprises. Surveys reveal that many data analyst and data scientist preferred spark over map reduce, which is batch-oriented and it does not offer itself for interactive applications and real time stream processing. As we advance towards the Internet of things, we towards the era of sensor based things too, the sensors that are intended to send the data back to the mother ship repository. The data that we deal with is mostly very complex and is deployed across various relational and non-relational systems. However, the demand for analytical tools is increasing. Such tools helps us in extracting and utilizing data stored anywhere. Even for the sensor input data, the input data is tremendous. To cater the efficiency, the Metadata catalogues help us to relate and understand data. The Machine learning is definitely automating the task of finding data in Hadoop. Some of the emerging tracks in Big Data are in the fields of sensing and Internet of things Services, Smart City Data, Big Data Networking.

7 a) NoSQL comparisons

The NoSQL system of designing the database is a relation less and hence schema less, they are now constructed on a single model but and layer out to adopt different ones. There are almost a handful of different operational models for NoSQL. Key/value, e.g. in MemchacheDB, Redis, etc.

Column, e.g. HBase Document, e.g. MongoDB, Couchbase, etc. Graphs, e.g. Neo4j, OrientDB. I have used MongoDB which is document based. This technology uses deeper nesting and handles much more complex structures.

8 b) Defining data model

I have created three collections in my paper. They are movies, ratings and users. The data model used in Movies is as { "_id" : ObjectId("58ab3f58ce38be146435cc76"), "id" :"35", "title" : "20", "release_date": "F", "IMDBURL" : "homemaker", "genre ":42459" } The data model used in ratings is as { "_id" :ObjectId("58ab3f58ce38be146435cc76"), "user" : "35", "movie" : "20", "rating" : "F", "timestamp" : "homemaker

9 Global Journal of Computer Science and Technology

Volume XIX Issue II Version I18 Year 2 019 ( ) C }

The data model for users is { "_id" :ObjectId("58ab3f58ce38be146435cc76"), "id" : "35", "age" : "20", "gender" : "F", "occupation" : "homemaker", "zip_code" : "42459" }

Process to be followed for data conversion INPUT: The input for this proposed paper is a movie.normalised_2015-11-30.sql file, which i have compressed to zip file by using WinRAR file compressor and later imported the same into XAMPP phpMyAdmin.

10 Conversion Methodology

The first tool we used was XAMPP. XAMPP Stands for Cross platform (X), Apache (A), MariaDB (M), PHP (P), and Perl (P).

Using XAMPP, i have taken the compressed input file movie and then normalized and converted it to JSON file. But before it, i have created a view of genres and movies. The view is named as moviefile. The attached snapshot covers the view creation with only the required attributes needed. The original database is with five tables where i merged two tables into one view and named it as moviefile. In Moviefile view, i have taken id, title, release date, IMDBURL. I dropped video attribute since the column was empty.

The code we used to create a view is as follows: CREATE VIEW Moviefile AS SELECT id, title AS Name_OftheMovie, release_date AS DateofRelease, IMDBURL ,Genres FROM movies, genres WHERE movies.id = genres.id ;

But before i exported the file, i supposed to convert the view created above with the SQL files users and ratings to JSON file which is later imported into NoSQL manager for MongoDB. The NoSQL Manager GUI which has an embedded smart shell to work on, act as a platform to work upon to maintain, update, embed, delete, interact and querying the database repository in a protocol set environment. The performance rate of NoSQL manager for MongoDB is a grade efficiency, especially the enterprise version. its easy to use document view and easier management utility features make it excellent to use and implement. It's SSH tunneling capacity for the connection, enables us to work on a highly secure system. The Map-Reduce operations inbuilt editor feature is remarkable in itself. It saves all map reduce parameters on the current session in an XML format. Another point to add is its GridFS which will divide the file onto chunks of size 255KB, with an exception of the fast chunk with as large as needed. Therefore the technological feature uses two collections; one for the file chunks and the other for file metadata.

Though for analyzing the data, MongoDB can work very well, but still the complexity of the database may increase while leading to have a platform to hold it. This is where Hadoop can provide a powerful framework for complex analytics and measurements. During the encounter of such a scenario, data can be pulled from MongoDB and processed and cared by Hadoop via the tool of map reduce. Nevertheless, Hadoop is stronger enough so that it can act as data warehouse and can hold the central control.

Data business analytics now has an option of holding Map Reduce [4] and Pig Map Reduce procedure is used to extract, transform and load data repository from one location to another. It pulls data from a place, transforms the data by using new methods and after transforming it to the desired shape it will load the data to another repository. Using this approach, i can handle huge data very easily. MongoDB is implemented in C++ [7][8] **??**9] Select the JSON file type and enter the file name along with path, is the first foremost step. Figure 1 represents it. IV.

11 Results and Discussions

The MongoDB guarantees data integrity with multi-documents transaction which makes MongoDB remarkable. This makes it well versed for developers to address the complete set of queries on the database. Interacting with the database constructed in MongoDB, supports multi-documents due to which a global consistent view of data is maintained. This new outlook of MongoDB made it possible to transact through multiple databases with the power of multi-documents ACID transaction.

I was successfully able to convert the JSON files to MongoDB format and consequently. Now my database is ready for interactions. Once the database is ready, i can extract information from the MongoDB multisite database by verifying multi-document ACID transaction.

MongoDB Queries 1. db.movies.find ({title:{$regex:/^A/I}}): This extract titles starting with letter A.

2. The Second query is db.ratings.count ({ rating:

1 })

I have movies as the collection name and ratings as one of the attribute and count is as the member function which will count the number of instances of ratings.

V.

12 Conclusion

The efforts to ensure the smooth and flexible functioning of a business, depends on the software capabilities of the Big Data Management Framework, The Data curation, scalability and data security are the key features. Through this paper it is clear that interactions have become easier on Big Data and even huge real time data management is possible with high er level of understanding and accuracy. The sensor data and many more can be translated to a MongoDB format which is composed of fields and value pairs. The document in MongoDB is being similar to JSON. In this paper, JSON enabled the transferring of data between the master server and the web application, in human readable form. It is concluded from the observed results that MongoDB provides high performance on data persistence, support for embedded data models, reduces I/O activity on database and high speed interactions can include keys from embedded documents and arrays. The resultant data repository has been in document format. The JSON document that i have, can be very easily parsed with the help of JavaScript. [1] [2]
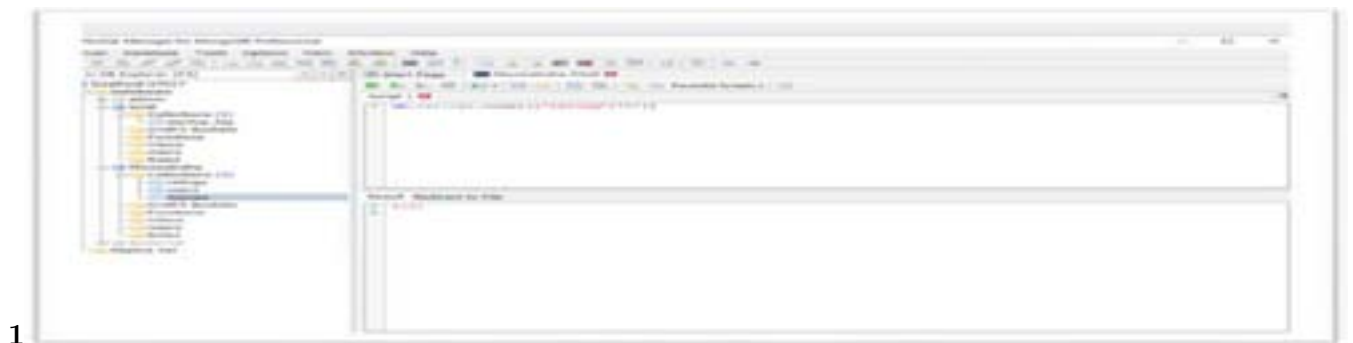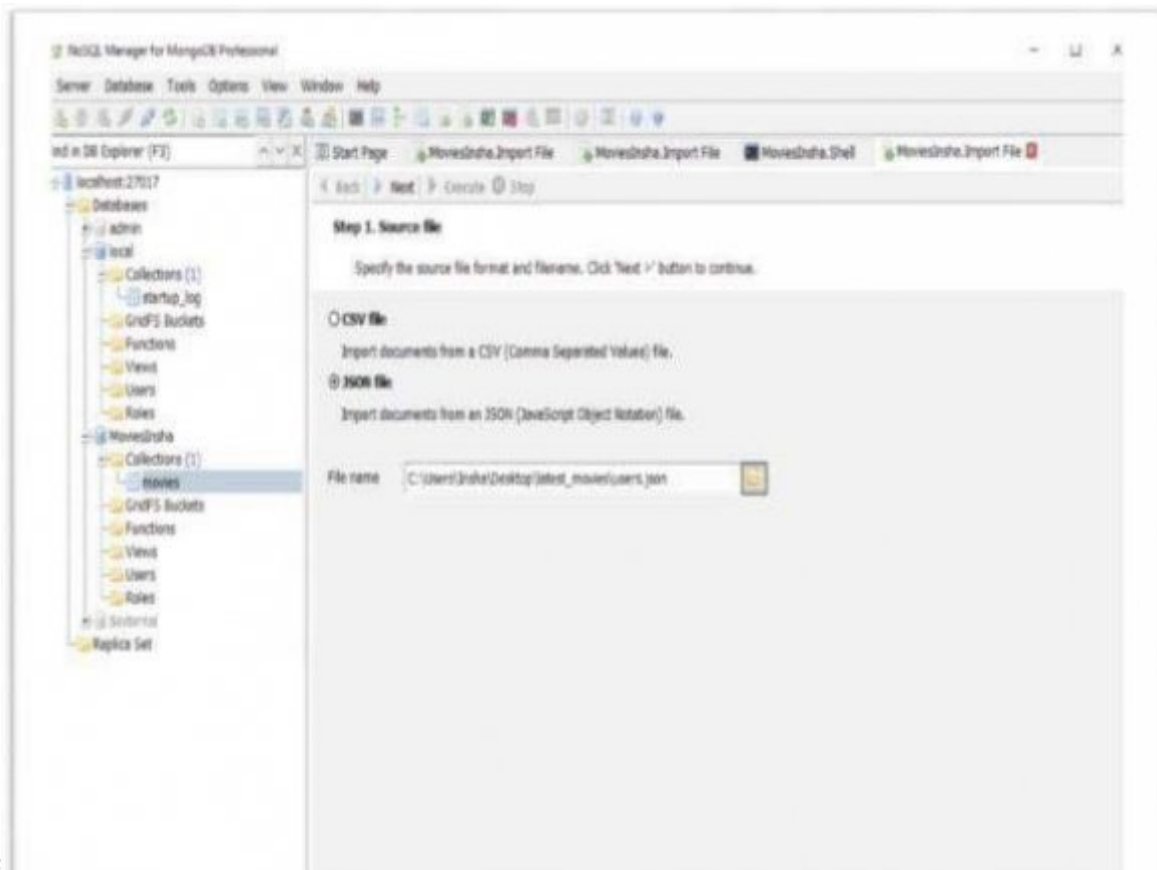

Figure 1: Figure 1 :

2

Figure 2: Figure 2 :

Figure 3:

12 CONCLUSION

174 [Carme et al.] , Julien Carme , Atos R&d Engineer , Francisco Worldline , José .

175 [Jimenez] , Ruiz Jimenez . *Atos Spain, Open Source* (Technical Architect Manager)

176 [Hall (2014)] , S Hall . `http://www.scriptrock.com/articles/mysql-vs-mongodb,acceseednov`
177   2014. Jul) MySQL vs MongoDB, jul, 2014. 2018.

178 [Berg et al. ()] 'History of Databases'. K Berg , T Seymour , R Coel . *International Journal of Management and*
179   *Information Services* 2013. 17.

180 [Lawrence (2014)] 'Integration and Virtualization of Relational SQL and NoSQL Systems Including MySQL
181   and MongoDB'. R Lawrence . *Proceedings of International Conference on Computational Science and*
182   *Computational Intelligence (CSCI)*, (International Conference on Computational Science and Computational
183   Intelligence (CSCI)) Mar.2014. p. .

184 [Solutions for Big Data Management] *Solutions for Big Data Management*, Atos.

185 [Stonebraker (2013)] Michael Stonebraker . *SQL databases v. NoSQL databases*, Dec.2013. MEC. p. .

186 [Khan (2013)] 'Vanita Mane SQL Support over MongoDB using Metadata'. Sanobar Khan , Prof . *IJSRP* October
187   2013. 3 (10) .