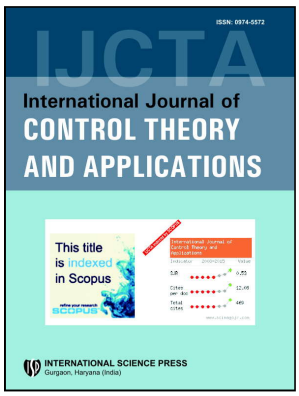

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 19 • 2017

A survey on approach of data migration process from relational database to column oriented NoSQL database

Krina Shah[1] and Rohit Srivastava[1]

[1] *Department of Computer Engineering, Parul University, Vadodara, Gujarat, India, Emails: krinashah1511@gmail.com,* rts080185@gmail.com

Abstract: In recent times, we are seeing a huge surge in generation and flow of data around us. The conventional ways of handling and dealing with data may no longer be practical and convenient for handling such huge amounts of data. In this generation of Big Data, data migration has come up as a possible solution to many of the problems caused due to use of legacy systems. To embrace this massive and large scale proliferation and growth of data, we need to convert the relational databases into NoSQL database and achieve higher flexibility and scalability. Hence, in this work a framework is proposed for data migration from relational database into NoSQL column oriented database.

Keywords: Big Data, NoSQL database, Cassandra,Data migration, Column oriented database, No-SQL database, distributed database.

1. INTRODUCTION

In current scenario, Relational databases are an integral part of data storage and analysis. But with the advent of Big data and burgeoning data growth, NoSQL databases have come to fore and presented itself as a solution to overcome the shortcomings of relational databases. These have the capability of dealing with both, the structured as well as the unstructured data.

NoSQL, for their ability to access these huge amounts of data at lightening speeds , flexible schema support, scalability and use of distributed databases, they have become the go-to option for big data execution and implementation. Hence the Not-Only SQL databases are now extremely popular for usage in handling terabytes and petabytes of data.

NoSQL does not replace the conventional SQL but it rather adds to the existing features and functions of SQL. At many places the relational databases are bolstered by NoSQL databases for additional application support. One cannot give a comprehensive definition for NoSQL, but majority of NoSQL databases share certain useful properties like flexible schema, not using relational models and better performance on clusters. Also, majority of these databases are freely available and can be modified and redistributed.

The multifaceted and flexible behaviour of NoSQL databases make them extremely desirable for applications of big data. The NoSQL databases rely and operate on the idea of Denormalization in which information gets copied and duplicated with an aim of easy retrieval of information from existing table instead of fetching it all from different tables thereby minimize the query processing time and increasing flexibility and adaptability

There are 4 main domains into which we can segregate the NoSQL databases. These are: Key value data store, column oriented, Document oriented and Graph Databases. Each part aims at a distinct type of data. Now we will be focusing, column family databases. These ever increasing information aggregation and capacities have questioned the abilities of present day databases and have augmented the need of shift towards the NoSQL databases(For e.gCassandra) to increase database flexibility and adaptability. So there is a shifting of schemas from sql to nosql formats which requires the tables to be denormalized.[1]

First category of NoSQL database , key-value data stores The Key value data stores model is straightforward and simple and yet is quiet potent and efficient. These store data in a way that are schema free. The technique is closely matched to maps or dictionaries where non identical keys are used to address data and as the values are uninterpreted byte arrays which the system cannot see, the keys is the only option left to recover or fetch the stored data. There are two partitions in the data. First is the string which is a representation of the key and second is the actual data which is to be considered as value thereby creating a key-value set. These stores closely resemble the hash tables in which keys are used for indexing which makes it faster than the existing RDBMS. The modern stores now value higher scalability over consistency

The second is column oriented database. The data is put in a column-family having row-key with the columns as row. In spite of the fact that at a logical level the database may show up as putting away data in tables just like the case with the SQL yet at the physical level the data is put away on a for each column family [3].It stores data in distributed database and uses the concept of column-by-column storage style. Each key is linked with one or more columns .Providing higher scalability for data accessing and storing for large databases.

Third is Document oriented database and it stores the data in form of documents .It contains key-value pair in the document and keys are mandatory to be unique and Every document contains special key "ID", Documents inside a document-oriented database are somewhat similar to records in relational databases, but they are much more flexible since they are schema less. Data is stored in JSON,XML,etc format and they are very flexible and schema-free. The Document Oriented database should not be used to store data if the database having lots of relationship among data and if they are much more normalized.

Graph Databases are used to represent entities and relationship between entities where entities are represented as nodes in the database and the relationships between the entities are represented as directed edges. Graph databases are gaining importance as they are now being deployed in organizations for managing data within applications like social networking.[1]

Table 1
Examples of different kind of NoSQL Databases[2]

Type of Database	*Example*
Key-Value Store	Azure Table Storage, AmazonDynamoDB, Redis, Berkely DB, etc
Column-Family Database	Cassandra, Hbase, Hypertable, Amazon Simple DB etc.
Document-Oriented Database	MongoDB,CouchDB, OrientDB, etc
Graph Databases	Neo4J, Infinite Graph, etc.

2. NOSQL AND CAP THEOREM

Google Inc , in 2006 published a paper on Big table and then came the paper on the Dynamo by Amazon in 2007. The idea of data getting stored in column families instead of traditional tabular style was brought about by Big

Table. Column families were referred to as collection of correlated data which was on many occasions accessed together.

As the modern day companies are producing huge volumes of data , new alternatives are getting developed to address the performance and scalability requirements of such massive data. All these alternatives are grouped under NoSQL databases as several of them may not support SQL for querying the data.

In distributed database system the CAP theorem applies as follows:

Consistency: All the system servers will contain or have access to the same data and hence anyone who uses the system will be given the same copy irrespective of which of the available server answers the request.

Availability: The system will always be available to reply to an incoming request be it a message of system not working properly.

Partition Tolerance: In case of failure of individual servers, the system still continues to function as a whole.

Above we discussed three properties- consistency, availability and partition tolerance and the CAP theorem says that NoSQL databases can hold only two of the three properties.

Having said that, we can optimize the two properties namely, consistency and availability by properly handling partitions and as a result achieve the best trade off amongst the 3 properties.

3. OVERVIEW OF CASSANDRA AND HBASE

Here there is an overview of certain characteristics of column oriented databases i.e Cassandra and HBase

3.1. Cassandra

Cassandra is an open source NoSQL database management system and it's a project of Apache Software Foundation. And it handles large amount of structured, semi-structured and unstructured data and language in which it is writtenis java.

Cassandra also works across multiple datacenters and its fault tolerance and it supports enterprise applications with constant availability and improving the performance rate of Cassandra and writes operation of Cassandra is much faster than other databases and provide higher scalability and higher availability. Cassandra does not support master-slave architecture as it's designed as master-less and peer-to-peer architecture. Cassandra is maintain industry-leading read and write performance for supporting the flexibility and scalability of zettabytes and petabytes of data

3.2. HBase

HBase is also a Column oriented NoSQL database developed in Java to that of Google's Big Table). Hbase database that run on top of Hadoop HDFS(Hadoop Distributed File System). A database that gives ongoing read/write access to those substantial datasets. HBase also handles large amount of data and performs read and write operations for these datasets and access speed is substantially higher but not compared to Cassandra database. HBase is also a apache software project and it is working on variety of schemas and different structures. HBase is working on master-slave architecture.

HBase has a feature of adding a random searches and generating indexes on row or column of the column oriented database .HBase work on Hadoop and Hadoop has the functionality to perform batch processing

HBase, has single HMaster server and many HRegion Servers. The HMaster server in Hbase has the responsibility of performing administration managing and monitoring the cluster and assigning region to region server and controlling the load balancing and failover.

In the event that different HMaster servers are utilized, endless supply of the HMaster that adjusts the heap among the area servers, another HMaster consequently accepts this accountability. In this HBase gives a programmed failure bolster. Be that as it may, a solitary purpose of disappointment happens in situations where just a single HMaster server is utilized. HBase is utilized as a part of situations where the application manages irregular read/write operations to huge data.

Cassandra‘s performance is the most fastest among the databases and gave blasting quick write speeds. Quick writes into the database is a one of a kind component of Cassandra. HBase, then again, gave write speeds that were almost twice as quick as the conventional connection databases.

3.3. Cassandra supplements Hadoop?

Similarly as with legacy SQL database applications, there is commonly need in present day Web application, social media, mobile and IOT applications to have a database gave to online operations and a batch-oriented data warehouse environment that supports the processing of colder data for logical purposes.

Apache Cassandra is an impeccable database decision for online Web and versatile applications, while Hadoop focuses on the preparing of colder information in information lakes, data warehouses, and so forth. This permits an IT association to successfully bolster the distinctive diagnostic “beats” expected to fulfil client prerequisites and maintain the business

3.4. How does Cassandra contrast to HBase?

HBase is utilizing more time for an online application in light of the fact that a current Hadoop usage exists at a website and not on the grounds that it is an ideal choice for the system. Hadoop(HBASE) is ordinarily not a decent decision for growing dependably on real time data and is almost 3-4 years behind Cassandra in numerous specialized regards.

3.5. In comparison to HBase, Cassandra:

I. Cassandra’s performance is higher compared to Hbase

II. Availability and fault tolerance are the major feature of cassandra as it supports peer to peer architecture and no single point of failure

III. Here CQL is just like SQL but differ in technical aspects and using CQL development is much simple and easier.

IV. Cassandra is very powerful and supports multi-datacenters as its having master less architecture and it is simple to setup with less requirements

4. FRAMEWORK FOR MIGRATION

Here, a framework is proposed for data migration from relational database to a column oriented database (Cassandra

The steps of our schema migration are as follows.

1) Establish a connection between RDBMS and column oriented database i.e Cassandra.

2) After successful connection check that is there any table is created in cassandra? if not then make table with same name as in relational database and map the data type between SQL and Cassandra and then table is created.

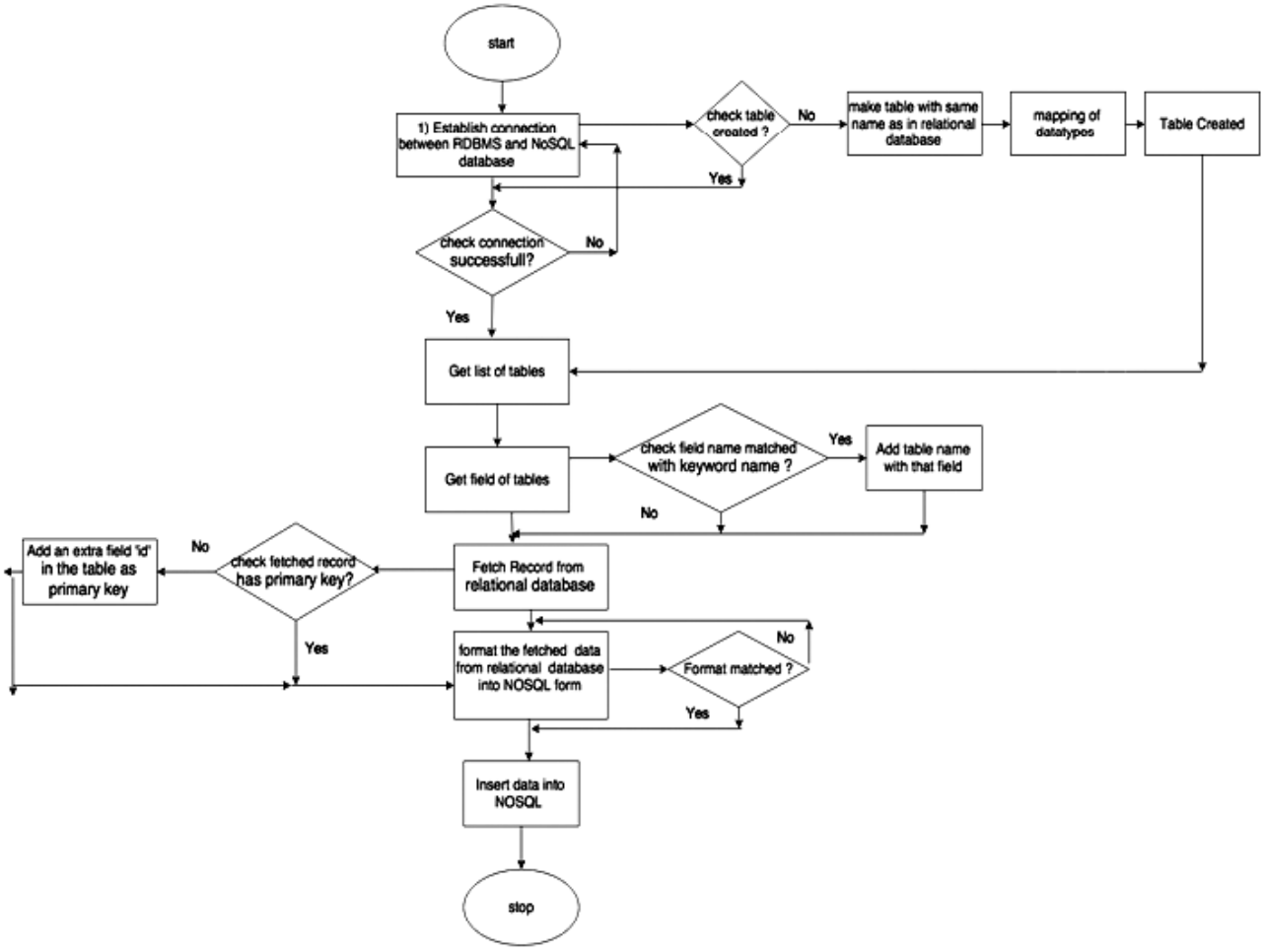

Figure 1: Flowchart of proposed data migration

3) Get list of table from relational database, after table created in cassandra

4) Also get list of fields from the list of tables and check if the field's name match with the keyword of the database then add table name with that perticular field name.

5) Fetch the record from the relational database and check if the record has primary key .If the record doesnot have primary key than add the key named 'id' in the table as primary key and formate the data into nosql form.

6) After the matched format of cassandra Insert the data into NoSQL database.

7) After successfully migration of a record into cassandra, repeate the process same as for the record having primary key in relational database.

5. CONCLUSION

As we discuss about the framework used for performance improvement of relational databases into column oriented databases based on their storage style, it is understood that after the migration of Relational Databases into NoSQL the scalability and performance of existing system has enhanced to a larger extent. The storage capacity of database is highly increased in comparison to Relational Database.

REFERENCES

[1] Vishal Dilipbhai Jogi and Ashay Sinha ,"Perfromance Evaluation of MySQL,Cassandra and Hbase for Heavy Write Operation,"3rd Int'l Conf. on Recent Advances in Information Technology RAIT -2016

[2] chao-Hsien Lee and Yu-Lin Zheng"SQL-to-NoSQL Schema Demoralization and Migration: A Study on Content Management Systems," 2015 IEEE International Conference on System,Man and Cybernetics

[3] koshy George and Tessy Mathew"Big Database Stores A review on various big data datastores",2015 IEEE

[4] Leilei Chen2, Yang Zheng2 "A Federated Approach on Heterogeneous NoSQL Data Stores "IEEE,2013 .

[5] Li-Yung Ho, Meng-Ju Hsieh, jan-jan Wu,Pangfeng Liu "Data Partition Optimization for Column-Family NoSQL databases,"2015 IEEE International Conference on Smart city/Social/Com/SustainCom together with Datacom 2015 and SC2 2015.

[6] Dharavath Ramesh, Ashay Sinha, Suraj Singh,"Data Modelling for Discrete Time Series Data using Cassandra and MongoDB","3rd Int'l Conf. on Recent Advances in Information Technology RAIT -2016

[7] Michael J. Mior, Kenneth Salem,Ashraf Aboulnaga, Rui Liu," NoSE: Schema Design for NoSQL Applications", ICDE 2016 Conference

[8] Satoshi FUKUDA, Ryota KAWASHIMA Shoichi SAITO and Hiroshi MATSUO," Improving Response Time for Cassandra with Query Scheduling, 2013 First International Symposium on Computing and Networking

[9] Jen-Chun Hsu, Ching-Hsien Hsu, Shih-Chang Chen, Yeh-Ching Chung," Correlation Aware Technique for SQL to NoSQL Transformation", 2014 7th International Conference on Ubi-Media Computing and Workshops

[10] David Bermbach, Steffen M¨uller†, Jacob Eberhardt and Stefan Tai," Informed Schema Design for Column Store-based Database Services", 2015 IEEE 8th International Conference on Service-Oriented Computing and Applications

[11] https://hbase.apache.org/poweredbyhbase.html

[12] http://www.datastax.com