I J C T A, 9(11) 2016, pp. 5199-5206
© International Science Press

CASSANDRA VS MYSQL: MODELLING AND QUERYING FORMAT

Neeru[*] and Baljit Kaur[**]

Abstract: Nowadays with the advent in technology, there is massive increase in digital data. Due to this increased in data, Relational Databases are losing their magnitude and proved inept to process this data because of their rigid schema. To overcome these issues of Relational Databases, NoSQL Databases came into race which are non-relational, open-source and schema free. NoSQL Databases can process, manage and store Big Data efficiently. NoSQL Databases can be classified into four types: Key-Value Store, Document-Based, Graph Based and Column-Oriented Databases. This paper mainly focused on Column-Oriented Databases. Under Column-Oriented Database, Cassandra has been reviewed. Finally, Comparative study as well as query format of MySQL and Cassandra has been reviewed that will clearly define how queries in CQL are different from MySQL.

Key Words: Big Data, NoSQL, Cassandra, Query Format

1. INTRODUCTION

Big Data is defined as Prodigious bulky data that is very complex to operate, capture, analyze and manage by conventional techniques in reasonable time frame[1]. Big Data can be stored both structured, un structured and semi structured data. This data is very difficult to store and analyze for further future process[1]. Big Data and its analytics are at the center of contemporary data science and commerce. Now a days, data is growing at a huge speed. This will make it difficult to handling such fat data[2]. In 2003, people created 5 exabytes of data. Now a days, this much of data is fashioned even in two days. In 2014, this amount of data is increases to 3.72 zettabytes. It is expected to twice over each three years, and will reach about 10 zettabytes of data by 2017. So, To store this huge amount of data is big issue. To overcome this issues and to store Big Data, NoSQL Database are developed. NoSQL databases are generally described as open source non Relational databases. It does not offer SQL interface. Unlike Relational databases, NoSQL databases are flexible in nature. They don't require any fixed or static schema.

The paper is reviewed as follows: Section II represents Big Data characteristics, In Section III NoSQL Database are discussed, Section IV presents Column-Oriented NoSQL Databases and Section V represents the Cassandra and its features. In Section VI comparative study between Cassandra and MySQL database are reviewed. Section VII presents query format of Cassandra Query Language(CQL) and MySQL. Finally Section VIII concludes the work.

2. BIG DATA

To characterize Big Data, Now Seven Vs are defined that are Volume, Velocity, Variety, Veracity, Value, Variability, Visualization. Figure 1 illustrates the characteristics of Big Data are described below:

[*] M.Tech. Scholar, Department of CSE, Lovely Professional University, Phagwara, India neeru.mehra07@gmail.com

[**] Assistant Professor, Department of CSE, Lovely Professional University, Phagwara, India baljit.18610@lpu.co.in

2.1 Big Data Characteristics

- **Volume:** Now a days , Enterprisers are awashed with data. Volume of data increasing in such a pace that rather than in tera or pera bytes, Data is now available in exabytes or zettabytes. In today's world, Each millisecond large amount of data generated. At present the data existing is in petabytes and it is assumed that it increases to zettabytes in future. If we obtain all the data generated in the world between the beginning of the time and 2010, the same amount of data will be soon generated every second [9]. This increase of data makes difficult to store and analyze data using relational databases.

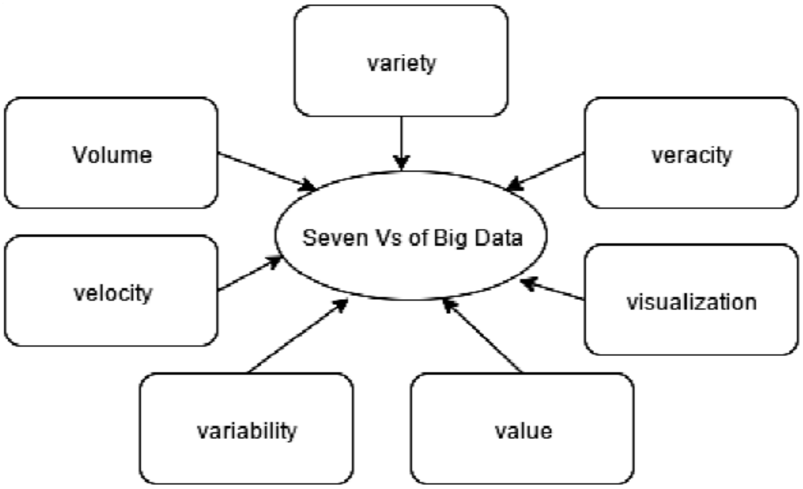


Figure 1: Seven Vs of Big Data

- **Velocity:** Velocity in Big Data illustrate the pace at which the data is retrieved from a diversity of sources and stored. This characteristic velocity is not even defines the speed of the data is incoming. It can also defines the speed of the data flows outside. Using Big data technology, now we can analyze the data while being generated, before putting to the databases.
- **Veracity:** Veracity describes the biases, noise and complexity of data. This mostly compact with quality and derivation of acknowledged data. Data can be categorized as good, bad, ambiguous or inconsistent[3]. For example Facebook post with hash tags and abbreviations. Now Big data allow to work with these types of data.
- **Variety:** Big Data comes from diversity of sources, therefore data is not available in particular structure or schema. Mainly, data is categorized as Un structured, semi structured and structured. Structured data is always in the form of tables or in rigid schema. Semi structured data preserve in structure form or may not be in structure form. Unstructured data is audio, blob, ASCII and videos.
- **Value:** Value describes the ability to turn data into value. It is very helpful in business to understand the customer needs and make the good customer relationship. It adds value to the business.
- **Variability:** Variability describes that data in Big Data which changes their meaning after several intervals of time.
- **Visualization:** As contradictory data can be stored in Big Data. To present that data is also a big task. This feature describes how to present data that is in the form of charts and other forms[5]

3. NOSQL DATABASES

As Big Data is difficult to handle, store and manage via traditional Relational data stores. Therefore, in order to process vast amount of real time heterogeneous data, a new solution is developed called NoSQL. It is referred to as "Not Only SQL". NoSQL databases are generally described as open source non Relational databases. It does not offer SQL interface. Unlike Relational databases, NoSQL databases are flexible in nature. They don't require any fixed or static schema[8].

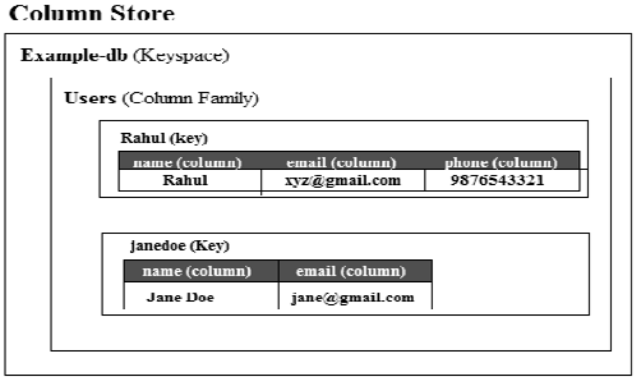
Classification of NoSQL Databases are as below:

NoSQL databases are broadly grouping into four classes:

- **Key-Value Store:** A Key-Value Store is a data storage model especially designed for retrieving, storing and managing data. Usually data is in the form of hash or dictionary. In Key-Value databases data can be store and retrieve easily because these databases use unique key that uniquely identifies the record and is used to quickly find the data within the databases.
- **Document-Oriented NoSQL Database:** A Document-Oriented database is designed to process, retrieve, store and manage document oriented data. All the data is pile up in the document type. The format of stored documents can XML, YAML and JSON[7]. Documented-oriented databases are a subclass of the Key-Value store. The main difference between the way in which data is processed.
- **Graph Based NoSQL Database:** Graph based databases are the databases that uses structures in the form of graph for semantic inquiry with edges, properties and nodes to represent data. Graph based databases are the combination of nodes and relationship between these nodes. There are further properties of nodes. In this database, nodes represent as entities. Graph based databases are mainly Based on Graph theory.
- **Column-Oriented NoSQL Database:** Column-oriented database is the database in which data storage occurs in the form Columns rather than rows. This paper mainly focused on Column-Oriented Databases which are reviewed in section IV.

4. COLUMN-ORIENTED NOSQL DATABASE

Column-oriented database is the database in which data is stored in Columns rather than rows. In Column oriented databases, Column families are exits so that columns are club in these column families. Column families can contain many numbers of columns that can be created at runtime. Read and write operation is done simultaneously with the help of columns rather than rows. Following figure 2 represents basic representation of Column-Oriented data stores.



name (column)	email (column)	phone (column)
Rahul	xyz@gmail.com	9876543321

name (column)	email (column)
Jane Doe	jane@gmail.com

Figure 2: Column-Oriented NoSQL Database

Column-Oriented databases also provides versioning feature. It allows storage of multiple versions corresponding to each column. Column-Oriented databases are well suited for OLAP (Online Analytical Processing) layouts. Column-oriented database is the database in which data is stored in Columns rather than rows. In Column oriented databases, Column families are exits so that columns are club in these column families. Column families can contain many numbers of columns that can be created at runtime. Read and write operation is done simultaneously with the help of columns rather than rows.

- **Column Family:** Column family is a structure in which group of column takes place. It is also called table of CQL (Cassandra Query Language).
- **Row-Key:** It is primary key in CQL that is Cassandra query language. Row of columns is uniquely identify by this Row key
- **Key Space:** Key space is similar to the database in MySQL. It contains full data of applications.
- **Cassandra Query Language:** The CQL is the query language of Cassandra. It is used to interface with Cassandra.

This paper mainly focused on Cassandra Column-Oriented NoSQL Databases that is described below.

5. CASSANDRA

Cassandra is a open source database management system that is easily scalable, fault tolerant, highly available, eventually consistent, schema free and build to handle infinite amount of data across different multiple servers. Apache Cassandra is a top level project of Apache that is born at facebook and built on Big Table of Google. Cassandra's architecture is responsible for its ability to perform, scale and offer incessant uptime. Cassandra has a master less ring design that is elegant, east to setup and easy to maintain. In Cassandra, all nodes play an equal role[4]. There is no master node. So, that all nodes communicating with each other have equal importance or all are equal. Architecture of Cassandra is conscientious for its capability to perform, extent and offer incessant uptime. Cassandra has a master less architecture that is like a ring. In Cassandra's architecture all nodes in the ring plays equal or identical role. There is no master node in the Cassandra cluster. Cluster of Cassandra is easy in arrangement and maintenance. It is built for extent architecture means that it is being able to leverage huge quantity of data and thousands of parallel users[10].

5.1 Cassandra Query Language

As we know that Cassandra is Column-Oriented NoSQL Database which is non- relational and open-source database. To Communicate with Cassandra database we requires some language. So, Cassandra NoSQL Database gives the language called Cassandra Query Language. It is nearly similar to SQL but it user CQL shell or cqlsh to interface with Cassandra[11]. In Previous versions of Cassandra, Thrift is used as interface which creates the objects of database and data manipulation.

The data types of CQL are also similar to SQL but some of them are different like for un-structured data blob, for numericals int, decimal etc, for characters varchar and ascii etc. Cqlsh command line is used for CQL and many other tools that are graphical are used to interact with Cassandra. Datastax community offers the Datastax Dev Center to communicate with Cassandra and their clusters.

Main Features of Cassandra

In this section, The Features of Column-Oriented NoSQL Databases are given below:

- **Decentralized:-** Data is distributed across the cluster. There is no master node. Every node plays an important and identical role. So, that there is no failure even in single point.

- **Scalability:-** It can horizontally scale the data. It provides random read and write operations. It can increase as new machines or users are increased and there is always uptime and no place for downtime or disturbance to the applications.
- **Fault-Tolerant:-** In Column-Oriented Databases, Data is replicated to multiple nodes so that it can be used for fault tolerance. If nodes are failed due to interruption or many other reasons then failed nodes can be replaced with zero downtime.
- **Replication:-** In this, we can replicate data across multiple centers. Data is replicated towards the multiple nodes so that there is no failure even in single point.
- **Flexible Data Model:-** Column-Oriented databases supports modern data types with fast reads and writes[12].

6. COMPARITIVE STUDY BETWEEN RELATIONAL DATABASES(MYSQL) AND COLUMN-ORIENTED NOSQL DATABASE(CASSANDRA)

The comparative study between Relational Databases(MySQL) and Column-Oriented NoSQL Databases are discussed below:

	Relational Database(MySQL)	*Column-Oriented NoSQL Database(Cassandra)*
Definition	MySQL is open source, relational database that is broadly used as Relational Database Management System(RDBMS).	Cassandra is open source, non-relational database that is broadly used as Distributed Database Management system. It was born at Facebook and built on Google's Big Table.
Developer	MySQL was developed by Oracle Corporation in 1995.	Cassandra was developed by Apache Software Foundation in 2008.
Implementation	MySQL was implemented using C/ C++ language.	Cassandra was implemented using Java Language.
Peculiar Features	• Master/Slave Architecture • SQL Compatibility • Transactions • Portability • Views	• High Availability • Fault Tolerance • Map Reduce • Horizontal Scalability • Replication
Competitive Advantages	MySQL lacks because of high availability and Cloud deployment	Cassandra gains more popularity because it provides 100% availability and ensures users that there is no single point of failure in the system. It provides operational ease for lowest entire cost of possession
Storage	It uses SQL for storage purpose	It uses SST(String Stored Tables) for storage purpose
Terminologies used	• Database • Table • Primary Key • Column-Name • Column-Value	• KeySpace • Column Families • Row Key • Column-Name/Key • Column-Value
Query Language	Structured Query Language (SQL)	Cassandra Query Language(CQL)
Prominent Users	NetQos, iStock, Italtel, NASA, Yahoo! Finance	Facebook, Netflix, OpenX, IBM,Digg

7. MYSQL VS CASSANDRA QUERY LANGUAGE (CQL) : QUERY FORMAT

Comparison of query format between Relational Database(MYSQL) and Column-Oriented NoSQL Database(CASSANDRA) has been exemplified with the help of Student record for a university.

Query 1. To Create a Database named 'University'

MYSQL	*CQL*
CREATE DATABASE University;	CREATE KEYSPACE University WITH REPLICATION = { 'class' : 'UniversityStrategy', 'replication_factor' : 2 };

Query 2: To Use Database named 'University'

MYSQL	*CQL*
USE DATABASE University;	USE KEYSPACE University;

Query 3: To Create Table name 'Student' and their corresponding attributes.

MYSQL	*CQL*
CREATE Table Student (Roll_no. integer primary key, Stu_Lastname varchar (20), Stu_Firstname varchar(20));	CREATE COLUMNFAMILY Student (Roll_no integer primary key, Stu_Lastname text, Stu_Firstname text);

Query 4: To Insert values into Table name 'Student'

MYSQL	*CQL*
INSERT INTO Student (Roll_no. integer primary key, Stu_Lastname, Stu_Firstname) VALUES (1, 'Singh', 'Ranbir');	INSERT INTO Student (Roll_no. integer primary key, Stu_Lastname, Stu_Firstname) VALUES (1, 'Singh', 'Ranbir');

Query 5: To Select/Retrieve Data from Table 'Student'

MYSQL	*CQL*
Select * from Student;	Select * from Student;

8. CONCLUSION

As the volume of digital data is blasting in this advanced world, Organizations need databases that are adaptable and sufficiently versatile to handle Semi-Structured and Unstructured Data. Though, Relational databases demonstrate their epoch, when it comes to the turn of admittance of real time data. NoSQL databases serves as best solution for giving availability, flexibility and scalability for heterogeneous data. This prompts movement of Organization's Relational databases to NoSQL databases. NoSQL Database can be classified as Key-Value, Document Based, Graph Based and Column -Oriented. This paper mainly focused on Column-Oriented database. Under the Column-Oriented Databases, Cassandra has been discussed. It has many features like replication, horizontal scalability and fault-tolerance. This paper motivates to provide an autonomous perceptive of the potency of Column-Oriented Databases and Cassandra. Then, Comparative study between Cassandra and MySQL has been discussed. Finally, With the examples, corresponding to MySQL queries, Cassandra Query Language(CQL) queries has been demonstrated. The Concept of Joins in Cassandra is not implemented yet, we will be to reinforce research in this area in future.

References

[1] Sagiroglu, Seref, and Duygu Sinanc. "Big data: A review." In Collaboration Technologies and Systems (CTS), 2013 International Conference on, pp. 42-47. IEEE, 2013.

[2] Katal, Avita, Mohammad Wazid, and R. H. Goudar. "Big data: issues, challenges, tools and good practices." In Contemporary Computing (IC3), 2013 Sixth International Conference on, pp. 404-409. IEEE, 2013.

[3] Tekiner, Firat, and John A. Keane. "Big data framework." In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, pp. 1494-1499. IEEE, 2013.

[4] Zhang, Du. "Inconsistencies in big data." In Cognitive Informatics & Cognitive Computing (ICCI* CC), 2013 12th IEEE International Conference on, pp. 61-67. IEEE, 2013.

[5] Schell, Roger. "Security—A big question for big data." In Big Data, 2013 IEEE International Conference on, pp. 5-5. IEEE, 2013.

[6] Zheng, Zibin, Jieming Zhu, and Michael R. Lyu. "Service-generated big data and big data-as-a-service: an overview." In Big Data (BigData Congress), 2013 IEEE International Congress on, pp. 403-410. IEEE, 2013.

[7] Han, J., Haihong, E., Le, G., & Du, J. Survey on NoSQL database. In Pervasive computing and applications (ICPCA), 6th international conference on (pp. 363-366). IEEE, 2011.

[8] Li, Yishan, and Sathiamoorthy Manoharan. "A performance comparison of SQL and NoSQL databases." In Communications, Computers and Signal Processing (PACRIM), 2013 IEEE Pacific Rim Conference on, pp. 15-19. IEEE, 2013.

[9] Turk, Ata, R. Oguz Selvitopi, Hakan Ferhatosmanoglu, and Cevdet Aykanat. "Temporal workload-aware replicated partitioning for social networks."Knowledge and Data Engineering, IEEE Transactions on 26, no. 11 (2014): 2832-2845.

[10] Huang, Xiangdong, Jianmin Wang, Jian Bai, Guiguang Ding, and Mingsheng Long. "Inherent Replica Inconsistency in Cassandra." In Big Data (BigData Congress), 2014 IEEE International Congress on, pp. 740-747. IEEE, 2014.

[11] Chebotko, Artem, Andrey Kashlev, and Shiyong Lu. "A Big Data Modeling Methodology for Apache Cassandra." In Big Data (BigData Congress), 2015 IEEE International Congress on, pp. 238-245. IEEE, 2015.

[12] Dede, Elif, Bedri Sendir, Pinar Kuzlu, Jessica Hartog, and Madhusudhan Govindaraju. "An evaluation of cassandra for hadoop." In 2013 IEEE Sixth International Conference on Cloud Computing, pp. 494-501. IEEE, 2013.