International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 10, October 2017
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Comparative Analysis of Column Family NoSQL Cloud Data Stores for Managing Big Data

Kiranjit Kaur[1], Dr. Vijay Laxmi[2]

Research Scholar, UCCA, Guru Kashi University, Talwandi Sabo, Punjab,(India)

Dean and Professor, UCCA, Guru Kashi University, Talwandi Sabo, Punjab, (India)

ABSTRACT

With the advancement in technology, bulk of data is generated. To store this data traditional database management systems were used but they were incapable to handle unstructured data. For managing this data NoSQL cloud data stores came into existence. This paper provides comparative analysis of column family NoSQL cloud data stores named HBase, Cassandra and SimpleDB. Comparative analysis is performed on the basis of different important parameters named Query language, partitioning, replication, consistency and concurrency.

Keywords: *Consistency, Concurrency, Partitioning, Query Language, Replication*

I. INTRODUCTION

With the growth of multimedia, social media and Internet of things (IoT) large amount of data is generated. In IT industry this data is referred as big data. This data is of three types: Structured Data-Relational data, Semi Structured data-XML data and Un-structured data: PDF, Word, Text and Media logs. Big data is a data which has three characteristics known as V^3: Volume, Variety and Velocity. Volume refers to the amount of data generated from different sources. Variety refers to the type of data. Velocity refers to the speed of data transfer [1].

Initially this data was stored in traditional databases. As the amount of data become larger and larger these traditional databases are unable to handle it. They make the whole process of data retrieving and storing slow. In technical terms we can say that the throughput decreases and response time increases. For managing this big data new cloud data stores are developed called NoSQL cloud data stores. These data stores are simple to design and finer control over availability. They are sometimes referred as BASE systems. Its full form is Basically Available, Soft State, and Eventually Consistent. Basically Available means it is available all the time whenever it is accessed. Soft state means it can tolerate inconsistency for a certain time period. Eventually consistent means the data store come to consistence state after certain time period [2].

II. COLUMN FAMILY NoSQL CLOUD DATA STORES

In column family NoSQL cloud data stores, data is stored in column oriented way. The dataset consist of several rows, each of which is identified by a unique row key known as primary key. Each row is composed of set of

column families and different row can have different column families. HBase, Cassandra and SimpleDB are some of the examples of column family cloud data stores. They provide more powerful indexing and querying than key-value stores because they are based on column families and columns in addition to row keys. They offer huge flexibility in storing any type of data [3].

In this paper the focus is on column family NoSQL cloud data stores. These data stores have much higher efficiency, flexibility, powerful indexing and querying than other cloud data stores. Comparative Analysis is performed on three data stores named Cassandra, HBase and SimpleDB.

2.1 Cassandra

Cassandra is written in java and it's built after Google's BigTable. This data store is designed to handle large amount of data across different servers with higher availability means there is no single point of failure. In other words it implements a Dynamo-style replication model with no single point of failure, but adds a more powerful "column family" data model. Cassandra is being used by companies such as Facebook, Twitter, Cisco, eBay and more [4]. It is highly scalable data store. It allows adding more hardware to accommodate more data and customers. It can handle all type of data e.g. structured, semi-structured and unstructured. It supports ACID properties and provide high throughput and reduce response time.

2.2 HBase

It is a distributed and scalable big data store which provides random and real time read/write access to data. It is an open source and non relational database modelled after Google's Bigtable developed by Google for the management of large volume of structured data. HBase emulates most of the functionalities provided by BigTable [5].

2.3 SimpleDB

It is part of the Amazon's proprietary cloud computing. As the name suggest it model is simple. It supports more than one group in one database so it is considered as column family cloud data store. It does not partition data over servers automatically, for achieving better scalability partition can be done manually. It provides asynchronous replication [6].

III. COMPARATIVE ANALYSIS OF COLUMN FAMILY DATA STORES

In this paper analysis on three column family data stores is performed, on the bases of different important parameters as in TABLE 1.

Table 1: Comparison Table

Parameters/Data Stores	HBase	Cassandra	SimpleDB
Query Language	No	CQL	Amazon Proprietary
Partitioning Technique	Range	Consistent Hashing	Manually
Replication	Yes	Yes	Yes
Consistency	Strongly Consistent	Configurable	Configurable
Concurrency	MVCC	Lock Based	No

3.1 Query Language

HBase does not have its own query language but it can be used with Hive. Cassandra data provide its own Cassandra query language (CQL). SimpleDB data store belongs to Amazon so it having Amazon proprietary query language for retrieving and storing data. Value 1- No query language, 2- Proprietary/limited and 3-full query language support is shown in Fig 1.

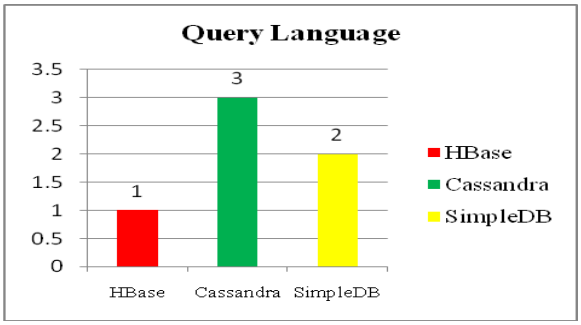

Fig 1: Comparision Chart of Query Language

3.2 Partitioning

HBase and Cassandra support partitioning whereas in SimpleDB partition is done manaully in DBdesign Model. HBase use range partitioning to store data on different machines whereas Cassandra use consistent hashing partition. It uses hash function to partition the data. Fig 2 is drawn on the bases of partitioning technique mannual or not. Value 0- mannual 1-Not mannual.

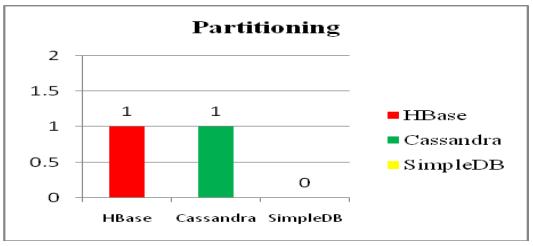

Fig 2: Comparision Chart of Partitioning

3.3 Replication

Asynchronous replication is supported in all of these data stores whether it's HBase, Cassandra or SimpleDB. This property prevents failure. Fig 3 is drawn on the bases data stores support replication or not. Value 1-yes and 0-No.

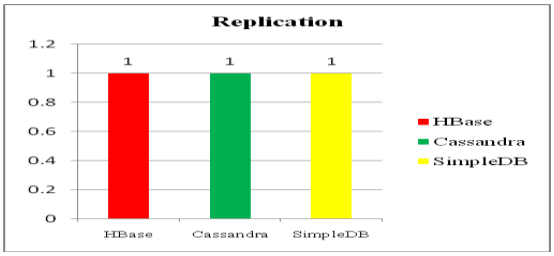

Fig 3: Comparision Chart of Replication

3.4 Consistency

HBase is a strongly consistent data store where Cassandra and SimpleDB are configurable data store means they are not always consistent but they are configurable which lead them to consistent state. Fig 3 is drawn where Value 3-Strongly consistent, 2- Configurable where as 1- Eventually consistent.

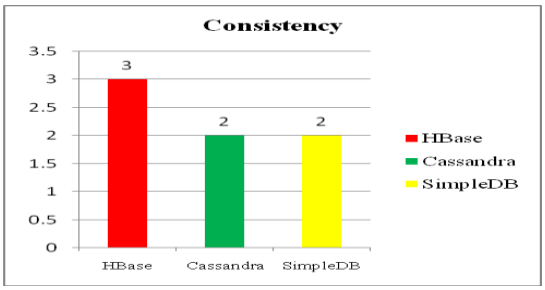

Fig 4: Comparision Chart for Consistency

3.5 Concurrency

HBase and Cassandra provide support for concurrency control with Multi Version Concurrency Control and Time stamp based concurrency control respectively. In SimpleDB optimistic concurrency control is there means no technique is applied to control concurrency in SimpleDB data store. 1-Concurrent approach available, 0-No Approach as shown in Fig 4.

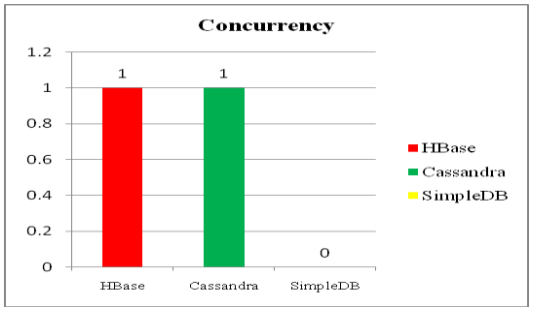

Fig 5: Comparision Chart for Concurrency

IV. CONCLUSION

After comparing these three data stores we reach to the conclusion that HBase is the suitable choice where strong consistency, replication management and concurrency control is required. Cassandra is used where full query language and replication is required. Consistency is configurable in this data stores. SimpleDB is easy to use and provide replication.

REFERENCES

[1] V. S. Kancharla, Survey paper on big data and scope of existing frameworks, *International journal of pharmacy and technology, 8(4), 25153-25157, 2016.*

[2] R. Cattell, Scalable SQL and NoSQL data Stores, *SIGMOD Record, 39(4), 2010.*

[3] K. Grolinger et al., Data Management in cloud environments: NoSQL and NewSQL Data Stores, *Journal of Cloud Computing: Advances, Systems and Applications, 2013.*

[4] J. Pokorny, NoSQL Databases: a step to database scalability in web environment, *International Journal of Web Information Systems, 2011.*

[5] P. Vijay and B. Keshwani, Emergence of Big Data with Hadoop: A Review, *IOSR Journal of Engineering, 6(3), 50-54, 2016.*

[6] I. Abaker, The rise of big data on cloud computing: Review and open research issues, *Information Systems, 47, 98-115, 2015.*