International Journal of Advance Research in Science and Engineering
Vol. No.4, Special Issue (01), August 2015
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A COMPARATIVE STUDY OF NOSQL DATA STORAGE MODELS FOR BIG DATA

Ompal Singh

Assistant Professor, Computer Science & Engineering, Sharda University, (India)

ABSTRACT

In the new era of distributed system where the data is distributed, too big (big data), diverse in nature, unstructured and has big user (global users 24 hours a day, 365 days a year), the organizations are migrated towards Non-Relational databases known as NOSQL (Not only SQL). Not only SQL is a database used to store huge amounts of data. NOSQL databases are distributed, non-relational and open source. Today NOSQL is mainly due to its Scalability and Performance characteristics .This paper surveying concept of NOSQL, compared with traditional relational databases, fundamentals theorem like ACID, CAP and BASE, different types of model used in NOSQL data stores with their examples.

Keywords: ACID, BASE, CAP, KEY VALUE, NOSQL.

I. INTRODUCTION

In today's world computer, internet and businesses generate huge volume of data and many centralized data have been changed into distributed data which has grown too big to managed and analysed by traditional databases like Oracle, DB2, SQL server, and MySQL.

However, after size of data and files has become bigger, e.g. VDO containing many Giga bytes, and DBMS' functioning that needs to support the clustering, these relational databases work slower [1] [2], due to they need to work on the Cartesian operation between the relations in order to join tables and views altogether before selecting the suitable rows for the SQL command. According to these problems, new kinds of database has been introduced and it is called NoSQL database [4] which function more quickly than did the relational databases though it works in the Big Data environment and distributed environment. More importantly, many companies i.e. Facebook, Twitter, Amazon and Google [5] that store the data of 100 – 1,000 million members also use NoSQL databases.

NoSQL as term was first used in 1998 by Carlo Strozzi as name of file-based database he was developing, since that time it has being used for the relational databases that omit the use of Structured Query Language (SQL). However, it was not before 2009 that it became a serious competitor to the term RDB. In present Eric Evans an employee in Rackspace Company described the ambition of the NoSQL movement, as "the whole point of seeking alternatives is that you need to solve a problem that relational databases are a bad fit for" [4]. The wildly usage of these NoSQL products encouraged other companies to make their own solutions and led to emerge of generic NoSQL database systems, now there is more than 150 NoSQL product [5]. These products come with issues like suitability to some areas of application, security and reliability.

Fig 1: Symbolic Representation of NoSQL

II. BACKGROUND

There are some Fundamentals must be aware of, ACID used to refer to the four properties of transactions (atomicity, consistency, isolation, durability).

1. Atomicity: means that you can guarantee that all of a transaction happens, or none of it does.
2. Consistency: means that database is stable at a valid state before or after any transaction occurs.
3. Isolation: means that one transaction cannot read data from another transaction that is not yet completed. Thus, requiring the concurrent transactions to be serialized.
4. Durability means that once a transaction is complete, it is guaranteed that all of the changes have been recorded to a durable medium (such as a hard disk), and the fact that the transaction has been completed is likewise recorded. [6]

III. CONCEPTS FOR NOSQL

NoSQL databases replace ACID properties with *BASE* properties (Basically available, Soft state, Eventual consistency) It is intended that the consistency after a transaction is not a solid state anymore (soft state). It shall be reached not right after finishing the transaction, but rather in some time during the operation (eventually consistent). The focus of BASE is the permanent availability. BASE is the opposite of ACID. NoSQL databases are classified in-between the spectrum from ACID to BASE. In the case of a bank, the eventual consistency is not what you want, thinking about two different balances on different servers! The balance must be equal just in time in every database involved in a money transaction session. In the case of an online book trade, the "just-in-time consistency" becomes less important. It does not matter if a book's price on one replication differs from another during a short time like a few hours. [7]

In addition, the *CAP* theorem must be mentioned, it first appearance was in year 2000; Eric Brewer introduced the idea that there is a fundamental trade-off between consistency, availability, and partition tolerance. These terms explained below:

Consistency - This means that the data in the database remains consistent after the execution of an operation. For example, after an update operation, all clients see the same data.

Availability - This means that the system is always on (Service guarantee availability), no downtime.

Partition Tolerance - This means that the system continues to function even if the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

The theorem says that only two of these aspects can be guaranteed at the same time in a distributed system. You have to "pick" two of them. [6] [7]

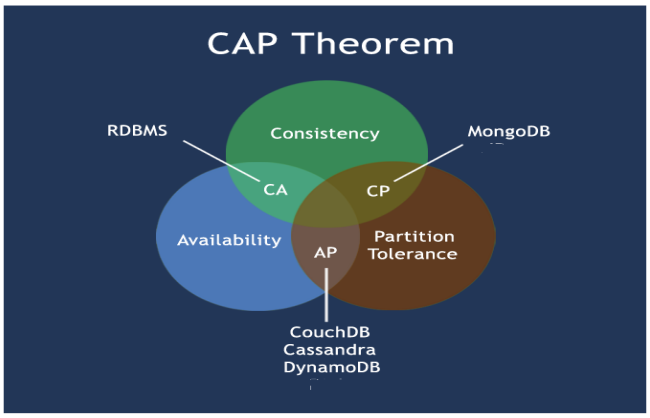

Fig 2: CAP Theorem

Here is the brief description of three combinations CA, CP, AP:

CA - Single site cluster, therefore all nodes are always in contact. When a partition occurs, the system blocks.

CP - Some data may not be accessible, but the rest is still consistent/accurate.

AP - System is still available under partitioning, but some of the data returned may be inaccurate. [7][8]

IV. NOSQL

The key advantage of NoSQL non relational database that it resolves the problem of big data as alternate database technology. In NoSQL databases data stores may not require fixed-table schemas, and usually avoid join operations, typically scale horizontally and easy replication support, simple API, eventually consistent / BASE (not ACID), a huge amount of data and more.[9] "Non-relational" may be more accurate term than "NoSQL", as some NoSQL DBs do support a subset of SQL.

4.1. Characteristics of NoSQL

- It support simple and flexible non-relational data models
- It stores large volume of data and having more flexible structure
- Use NoSQL without any inconsistency, in distributed environment so provide high availability
- No discontinuation of any work, if any faults or failures exist in any machine
- It does not have predefined schema.
- It does not support ACID transactions as provided by RDBMS
- Ability to scale horizontally leading to high performance over many commodity servers.

4.2 Why NoSQL

- NoSQL provides horizontal scalability better than vertical
- NoSQL support hardware getting cheaper and processing power increasing
- NoSQL support less operational complexity as against RDBMS solutions
- NoSQL provides, in most of the solutions you get automatic sharding etc. as default

V. DIFFERENT MODELS OF NOSQL

On the basis of CAP theorem NoSQL databases are divided into number of databases. There are four new different types of data stores in NoSQL [4].

5.1 Key Value Databases: Key value databases are combination of two things one is key and another is value. It is one of the traditional database systems. Key Value (KV) databases are mother of all the databases of NoSQL. Access data (values) by strings called keys.Data has no required format – data may have any format.
Key value databases stored data as hash tables. In this type of databases key is associated with every data in the database and it represents an attribute names and its value together. Key value databases provide faster execution of query and high concurrency compared to other non relational databases. For higher availability of data stores data objects are replicated. This is as illustrated in figure below. For example, let's take an example of car database as shown in figure.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Fig 3: Example of Key Value Database

5.1.1 Characteristics of Key value databases

- Number of keys can have a dynamic set of attributes in the key value databases during storage of data.
- Data stored in the database is stored in the alphabetical order.
- All the activities can be performed on the data i.e. CRUD (Create, Read, and Update and Delete).
- All the relationships to the data are stored in the application code (not explicitly spread).

5.1.2 Key Value (KV) databases uses

- It is one of the simple data model among all (we will discuss later) as it uses only key and a value.
- It handles huge data load.
- It scales to large volume of data.
- Replication of data is done using database in the form of ring. The replicated data is stored in the form of ring as well as in the alphabetical order.

Redis is Key-value Store database.

5.2 Document Stores Databases: These databases use records as documents. "Documents" are encoded in a standard data exchange format such as XML, JSON (JavaScript Object Notation) or BSON (Binary JSON).

These databases store unstructured or complex or semi-structured documents in hierarchal manner and helps in easy debugging and conceptualizing data.

Document stores databases are free from fixed schema and have dynamic nature so that data can be changed at any time. Document consists of a set of keys and values which are almost same as there in the Key Value databases. Each database residing in the document stores points to its fields using pointers as it uses the technique of hashing.

The figure depicts that it consists of number of databases in the document store such as databases 1,2,3,4 and is having its id A, B, C, D residing in it which is pointing to its database that is having some relation to it. Databases point to its value using some unique key residing in its database. This consists of an array of databases (that is in form of buckets).This will be clearer after taking an example discussed below.

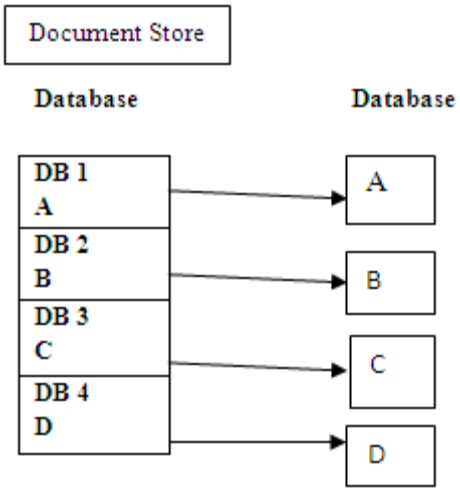

Fig 4: Example of Document Store Database

5.2.1 Characteristics of Document Stores Database

- Documents are addressed in the database using key (unique) that represents that document.
- There are number of varieties to organize data that is collections, tags, non-visible metadata and directory hierarchies.
- In this we can use a key-value lookup to retrieve a document.

An example record from Mongo, using JSON format, might look like

```
{
"_id" : ObjectId("4fccbf281168a6aa3c215443″),
"first_name" : "Thomas",
"last_name" : "Jefferson",
"address" : {
        "street" : "1600 Pennsylvania Ave NW",
        "city" : "Washington",
        "state" : "DC"
}
}
```

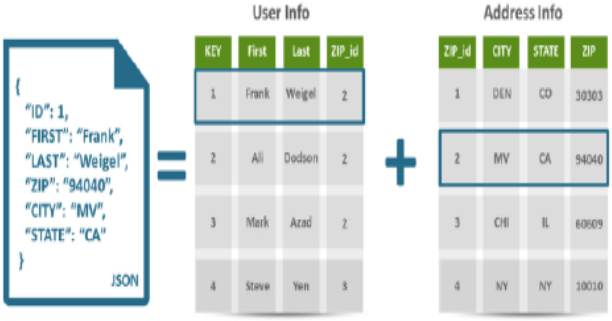

Fig 5: Example of Document Store Database using JSON

MongoDB and OrientDB are Document Store databases.

5.3 Columnar Databases

Columnar Databases are column-oriented databases also known as column family databases because data tables are stored as sections of columns of data, rather than as rows of data. There are two types of column oriented databases:

(1)Wide-column data stores

(2)Column oriented database

(1) Wide-Column data stores: Based on Google's Big Table store. Data tables are stored as sections of columns of data, rather than as rows of data.Wide Column data stores are those databases that are used for processing of web, streaming of data and documents.

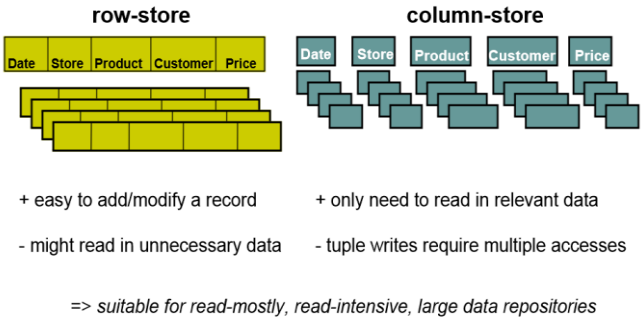

Fig 6: Column Based Database

The column is lowest/smallest instance of data. It is a tuple that contains a name, a value and a timestamp. The column is used as a store for the value, and has a timestamp that is used to differentiate the valid content from stale ones.

According to the CAP theorem, distributed data stores cannot guarantee consistency - availability is a more important issue. Therefore, the data store or the application will use the timestamp to find out which of the stored values in the backup nodes are up-to-date.

The structure of wide Column data store is as depicted is given below:

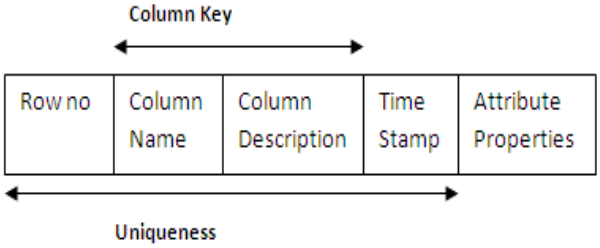

Fig 7: Structure of Column Based Database

Table 1: Structure Description of Wide Column Data Store

ATTRIBUTE	**MEANING**
Row No	It is a key that is unique in nature. It may be a string or a number.
Column Name	Data stored on the basis of column family.
Column Description	It describes the stored data item.
Time stamp	It tells the complete time of particular instance.
Data value	Value or attributes related to that corresponding key.

(2) Column Oriented Databases:

To understand column oriented databases let's take an example of bank database given in Table 2 whose attribute fields are EmpID, Salary and designation and values corresponding to it are as depicted in database.

Table2: Example of Bank Database

Emp ID	Salary	Designation
100	10,000	Clerk
200	20,000	Assistant Manager
300	30,000	Manager
400	40,000	Zonal Head

Representation of Row oriented databases and column oriented databases:

- Row oriented databases are those databases in which all the rows are put together one by one.
- Column oriented databases those databases in which all the values containing columns are put together.

With the help of the database given above we will represent row and column oriented databases which is as shown in below fig 8:

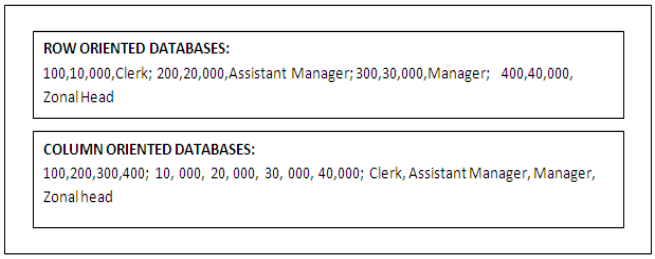

Fig 8: Example of Column Based Database

5.3.1 Characteristics of Columnar Databases

(1) Columnar databases are faster than row based databases while querying.

(2) In columnar databases, assignment of storage unit is done to each and every column.

(3) In the columnar DBMS only the required columns are read, so reading is faster in this case.

Cassandra and HBase are column family databases.

5.4 Graph Databases: A graph database is based on graph structure usually consists of nodes, edges and properties to store data. Graph use nodes to represent entities, edges to represent relationships and properties to represent attributes. [10].

By definition, a graph database is any storage system that provides index-free adjacency.

This means that every element contains a direct pointer to its adjacent element and no index lookups are necessary.

The structure of graph database is as shown below:

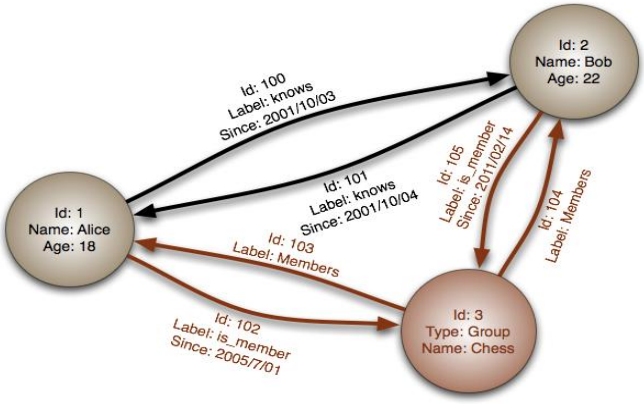

Fig 9: Graph Database

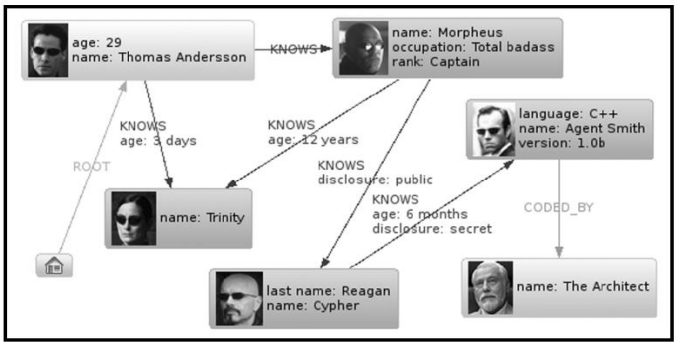

Fig 10: Example of Graph Based Database

5.5 Characteristics of Graph Databases

- Graph traversals are executed with constant speed independent of total size of the graph. There are no set operations involved that decrease performance as seen with join operations in RDBMS.
- Graph databases are having high performance in context to their deep traversals.
- These are used for shortest path calculations.
- These are scalable. But its complexity increases.

  Neo4J graph is graph based database.

VI. CONCULSION

This paper generally focuses on introduction of NoSQL databases with their functionality and characteristics. It also gives an overview of basic concepts like BASE and CAP theorem because NoSQL databases do not use ACID Property for data consistency. So, the concepts of BASE and CAP theorem introduce new type of NoSQL databases that are Key-Value databases, Document Store Databases, Columnar based databases and Graph databases with their functionality and characteristics. Finally NoSQL is big revolution in the future because most of current web applications and social networking websites are tend to depending on web also size of data is huge need to store in continues increasing rapidly. The future work is on new NOSQL.

REFERENCES

[1] Preecha Noiumkar, and Tawatchai Chomsir "A Comparison the Level of Security on Top 5 Open Source NoSQL Databases".

[2] James Manyika, "Big Data: The Next Frontier for Innovation, Competition, and Productivity," Executive Summary, McKinsey Global Institute, May 2011.

[3] Zikopoulos, P., Eaton, C., deRoos, D., Deutsch, T., & Lapis, G. (2012). Understanding big data: Analytics for enterprise class hadoop and streaming data. New York, NY: McGraw-Hill.

[4] R. Cattell, "Scalable SQL and NoSQL Data Stores," ACM SIGMOD Record, vol. 39, December 2010.

[5] "No SQL: An Overview of "No SQL Databases", Tim Perdue ,http://newtech.about.com/od/databasemanagement/a/Nosql.htm, 2014.

[6] Mohamed A. Mohamed ,Obay G. Altrafi ,Mohammed O. Ismail "Relational vs NoSQL databases: A survey", International Journal of Computer and Information Technology (ISSN: 2279 – 0764) Volume 03 – Issue 03, May 2014.

[7] S. Weber, "NoSQL Databases," University of Applied Sciences HTW Chur, Switzerland, 2010.

[8] N. A. L. Seth Gilbert, "Perspectives on the CAP Theorem," Singapore, 2012.

[9] Rakesh Kumar, Bhanu Bhushan Parashar, Sakshi Gupta, Yougeshwary Sharma, Neha Gupta"Apache Hadoop, NoSQL and NewSQL Solutions of Big Data", International Journal of Advance Foundation and Research in Science & Engineering (IJAFRSE) Volume 1, Issue 6, October 2014.

[10] An Oracle White Paper,"OracleNoSQLDatabase", September 2011,http://www.oracle.com/techetwork/database/nosqldb/ Leammore/nosql-database-498041.pdf.

[11] Luis Ferreira Universidade do Minho, "Bridging the gap between SQL and NoSQL", httpsikhote.files.wordpress.com201105artigo-mi-star1.pdf.

[12] DAMA - Philadelphia / Delaware Valley, the "Role of Data Architecture in NOSQL", Wednesday January 11th, 2012, http://www.damaphila.org/HaugheyNOSQL.pdf.