

Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases

Clarence J M Tauro

Centre for Research,
Christ University,
Hosur Road, Bangalore, India

Aravindh S

Dept. of Computer Science,
Christ University,
Hosur Road, Bangalore, India

Shreeharsha A.B

Dept. of Computer Science,
Christ University,
Hosur Road, Bangalore, India

ABSTRACT

Relational database is widely used in most of the application to store and retrieve data. They work best when they handle a limited set of data. Handling real time huge volume of data like internet was inefficient in relation database systems. To overcome this problem the "NO-SQL" or "Not Only SQL" Database came into existence. This paper discusses about problems with relation databases and how different types of NOSQL Databases are used to efficiently handle the real world problems.

Keywords

NoSQL, Key-Value Store, BigTable, Document Databases, Graph Databases, Berkeley DB, Tokyo Tryant, Voldemart DB, Neo4j, Hadoop

1. INTRODUCTION

The problems with Relation database is that it lacked handling exponential growth of data. Many organizations collect vast amounts of customer, scientific, sales, and other data for future analysis. Traditionally, most of these organizations have stored structured data in relational databases for subsequent access and analysis. However, a growing number of developers and users have begun turning to various types of non-relational, now frequently called NOSQL databases [2]. Their primary advantage of NOSQL Database is that, unlike relational databases they handle unstructured data such as documents, e-mail, multimedia and social media efficiently. The common features of NOSQL databases can be summarized as high scalability and reliability, very simple data model, very simple (primitive) query language, lack of mechanism for handling and managing data consistency and integrity constraints maintenance [1].

2. PROBLEMS WITH RELATION DATABASES

There are three major problems with Relation Databases which makes it inefficient.

First is the data set size. There is a huge growth of information in internet. International Data Corporation (IDC) analysis shown in Figure 1 shows the growth of data from the year 2007 to 2010 to be almost 25 times.

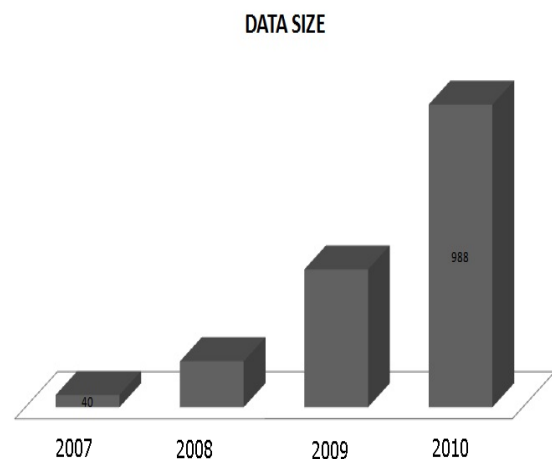


Figure. 1 Scalable of data size from 2007 – 2010

Second is the problem on connectivity. Over time, the information is getting more and more connected. Figure 2 shows the growth of connectivity over the years.

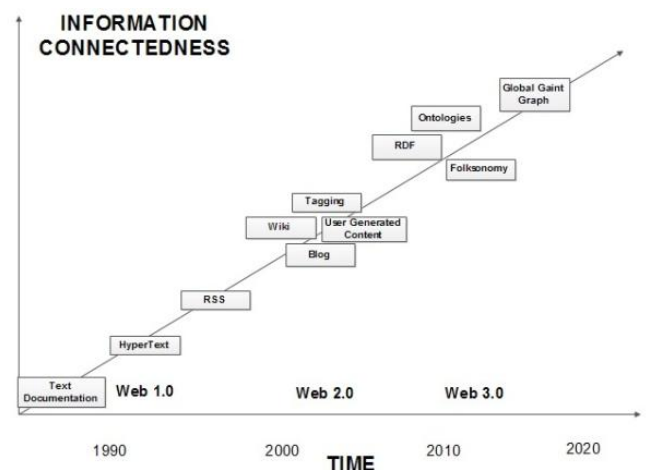


Figure. 2 Growth of connectivity

Third problem is about the semi structure. Semi structure information is information which has few mandatory attributes but many optional attributes. As the information grew, there was need to increase the columns of table, which would lead to sparse tables.

3. NOSQL CATEGORIES

NOSQL can be broken into 4 different categories.

- Key Value Stores
- Big Table
- Document Databases
- Graph Databases

Each database is individually good at dealing with size and complexities.

3.1 Key Value Stores

Key value data model means that a value corresponds to a Key. Although the structure is simpler, the query speed is higher than relational database, supports mass storage and high concurrency, etc., It provided support for query and modify operations for data through the primary key [3]. Key values represent bucket of data. For example, in case of a shopping cart mentioned in Figure 3, each shopping cart are represented in individual buckets and represented using a key value which could be user id.

The key values can be serialized using either java serialization or XML. This way is very fast to store as it just writes bits to the discs. Some of key value stores available in market are Berkeley DB, Tokyo Tyrant, Voldemart, Crassandra.

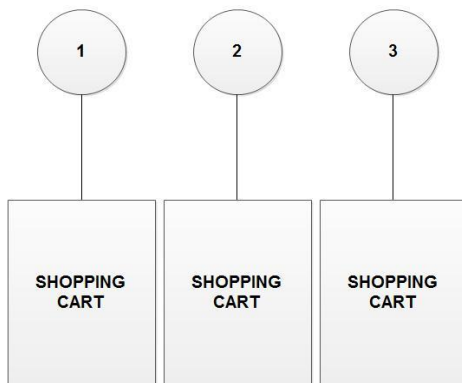


Figure. 2 Shopping cart example of Key Value Store

3.1.1 Berkeley DB

Oracle Berkeley DB is a high-performance embeddable database providing key value storage. Berkeley DB offers advanced features including transactional data storage, highly concurrent access, replication for high availability, and fault tolerance in a self-contained, small footprint software library [4].

3.1.2 Tokyo Tyrant

Tokyo Tyrant is a high performance storage engine, while the TT provides multi-threaded high-concurrency servers, it can handle 4-5 million times read and write operations per second [3]. While ensure the high performance of read and write concurrent, it uses a reliable data persistence mechanism [3].

3.1.3 Searching in key value store

Since everything is stored in a bucket, searching is done through manual indexing function where all essential attributes are added to an index table and searched. This may not be as popular, because it expects the user to have knowledge of index. Search was made easy in key value store

using lucene search engine, which retrieves number of key backs to look up in the key value stores.

3.1.4 Scaling to size

The biggest challenge to NOSQL databases is scaling to size. Scaling to size in today’s world is scaling horizontally, that is adding new machines. There are number of techniques to achieve this.

- Master Slave replication
- Sharding
- Dynamo model

a. Master Slave Replication

In Master Slave Replication, one server is designated as master and all write happens only to the server. Every update is propagated to all other servers. For example, if there are 1 master and 4 slaves machines, the write can happen only on 1 master server where as there would 5 machines which responds to read requests.

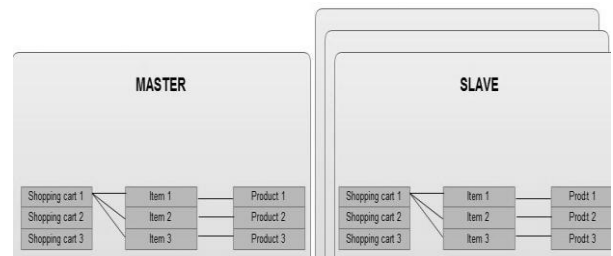


Figure. 3 Master slave replication of read write

b. Sharding

The problem with Master Slave system is, if there more write request than what one machine can handle. This leads to next system called Sharding. In Sharding, completely isolated structures are put into one server according to alphabetical order. For example- Figure 5 shows the example of all customer names starting from A-M are put on one server and N-Z are put on another server. This certainly can handle twice the workload. Similarly there can be ‘n’ number of machines added to ease the work load.

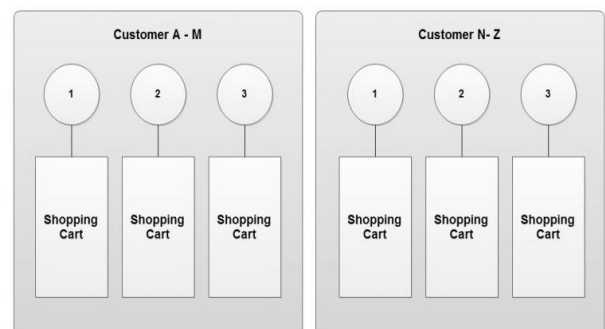


Figure. 4 Shopping cart example for Sharding model.

The problem with sharding is that when a new machine has to be added. It requires the online system to be shut down to shuffle the data across the new server.

c. Dynamo Model

Amazon came up with solution to the problem with sharding which is a dynamo model. In this model, not the entire online system is brought down to add a new machine. Instead, only a portion of the online system is brought down. There are three different approaches of DYNAMO model. The first approach is called “Basically Available”. This method basically has Amazon online selling few products like books, where as it would not be able to sell music. This is better than having Amazon offline completely. The next method is called “Soft State” where it checks most recent value. For example, it there were 20 copies of a book about one hr ago. It assumes there are copies available and allows the customer to order for the book. The third method is the most effective DYNAMO approach called “Eventually Available” which means photo uploaded on a social network server in India may not show up immediately in U.S. Whereas, eventually it will be available in U.S server in sometime.

Linkedin developed database called Voldemart is based on Dynamo model. This is another example of a key value store. It is developed in Java.

3.2 BIGTABLE

Search engine Zvents develop open source distributed data storage system hyper table [3] by drawing big table.

A BigTable is a light, scattered, constant multidimensional sorted map. Indexing of the map is done by a row key, column key, and a timestamp. In BigTable, un-interpreted arrays of bytes are used as values. BigTable stores structured data. Any type of data from text to serialized objects can be stored by applications. It does not impose any size constraint for each value. A table is allowed to have limitless number of columns. Data is indexed using row and column names that can be arbitrary strings [5].

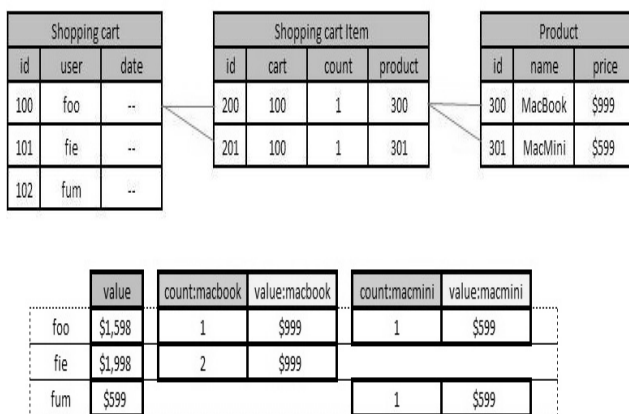


Figure.6 Normalized table VS Big table

Google developed its own big table model called Google BigTable. Google BigTable has been designed to scale into the petabyte range across hundreds or even thousands of computers, and also to ease the addition of more machines without much reconfiguration, thereby making the fullest use of the resources [5].

Google BigTable is built on top of the Google File System, Chubby and stored in an immutable data structure called SSTable which facilitates the storage of log and data files [5]. Chubby is used by BigTable to store the root tablet, schema details, access control lists, coordinate and identify tablet servers [5].

Hbase is the open source version of BigTable. Hbase emulates most of the functionalities provided by BigTable. Like most non SQL database systems, Hbase is written in Java. Hbase is an Apache open source project and aims to provide a storage system similar to BigTable in the Hadoop distributed computing environment. Hadoop Distributed File System (HDFS) is a distributed file system structure for operating on common hardware structures (commodity computers) characterized by low cost implementation [6]. Figure 7 explains Architecture of HDFS.

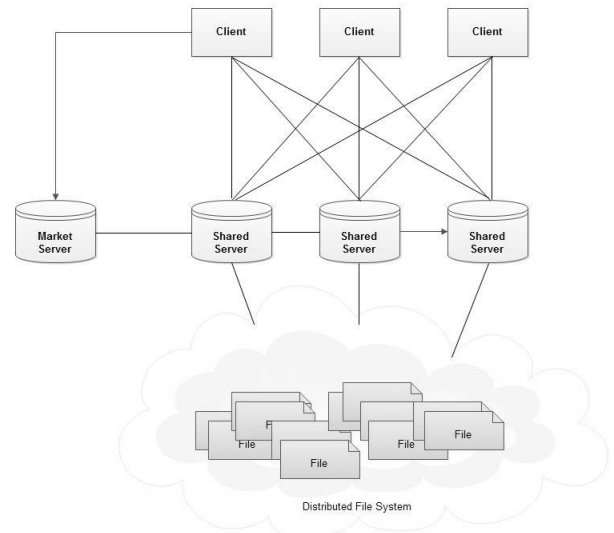


Figure. 5 Architecture of HBASE

3.3 DOCUMENT DATABASE

Document database is not concerned about high performance read and write concurrent, but rather to ensure that big data storage and good query performance [3]. Typical document database are Mongo DB, Couch DB.

Couch DB is one most popular document database which is flexible, fault-tolerant database, which supports data format called JSON. It provides REST-style API to ensure data consistency, Couch DB comply with ACID properties. In addition, Couch DB provides a P2P-based distributed database solution that supports bidirectional replication. However, it also has some limitations, such as only providing an interface based on HTTP REST, concurrent read and write performance is not ideal and so on [3].

Mongo DB is non-relational database, which features the richest and most like the relational database. It supports complex data types which uses BSON data structures to store complex data type [3]. It uses powerful query language which allows most of functions like query in single-table of relational databases, and also support index. High-speed access to mass data: when the data exceeds 50GB, Mongo DB access speed is 10 times than My SQL. Because of these characteristics of Mongo DB, many projects with increasing data are considering using Mongo DB instead of relational database.

3.4 GRAPH DATABASE

Graph Database works with 3 core abstraction, Node, relationships between nodes and key value pairs which can be attached to nodes and relationships.

Neo4j is a high-performance, NOSQL graph database with all the features of a mature and robust database [7]. It is a graph

database which is supported by neo technologies. It is developed in Java and it supports master-slave replication.

Allegrograph is another graph database which is developed by Franz Inc [8]. Allegrograph supports ad-doc query language called SPARQL. It provides REST protocol architecture.

4. CONCLUSION

This paper describes the limitations of traditional databases and the advantages of NO SQL database. According to each type of data models, introduce the current mainstream NO SQL database, and objective analysis of their strengths and weaknesses respectively, which will help user to choose NO SQL database in practice.

Companies need to consider the following options when deciding which properties NO SQL:

- Data Model
- CAP Support
- Multi Data Center Support
- Capacity
- Performance
- Query API
- Reliability
- Data Persistence
- Business Support

5. ACKNOWLEDGMENT

We are heartily thankful to Prof. Jibrael Jos, Prof. Joy Paulose and Dr. N. Ganeshan whose encouragement, guidance

and support enabled us to develop an understanding of the subject.

6. REFERENCES

- [1] Okman, Lior; Gal-Oz, Nurit; Gonen, Yaron; Gudes, Ehud; Abramov, Jenny; , "Security Issues in NoSQL Databases," Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi : 10.1109/TrustCom.2011.70
- [2] Leavitt, N.; , "Will NoSQL Databases Live Up to Their Promise?," Computer , vol.43, no.2, pp.12-14, Feb. 2010doi: 10.1109/MC.2010.58
- [3] Jing Han; Haihong, E.; Guan Le; Jian Du; , "Survey on NoSQL database," Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on , vol., no., pp.363-366, 26-28 Oct. 2011 doi: 10.1109/ICPCA.2011.6106531
- [4] Oracle Corporation, [online], <http://www.oracle.com/us/products/database/berkeley-db/db/overview/index.html> (Accessed: 9 February 2012).
- [5] Ramanathan, Shalini; Goel, Savita; Alagumalai, Subramanian; , "Comparison of Cloud database: Amazon's SimpleDB and Google's Bigtable," Recent Trends in Information Systems (ReTIS), 2011 International Conference on , vol., no., pp.165-168, 21-23 Dec. 2011doi: 10.1109/ReTIS.2011.614686