Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 5; April-June, 2015 pp. 495-499
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Mongo DB - NoSql Database: A Survey

Anu Batra[1], Anjali Pahwa[2] and Divya Sharma[3]

[1,2]M.Tech-CSE (Pursuing), ITM University Sector 23, Gurgaon, Haryana, India
[3]ITM University Sector 23, Gurgaon, Haryana, India
E-mail: [1]anubatra.16@gmail.com, [2]anjalipahwa1912@gmail.com, [3]divya@itmindia.edu

Abstract—*With the increasing requirements of higher volumes of data, agile development and cloud computing, many growing companies have moved towards a variety of databases that do not contain the table and key model which the traditional relational database management systems support. These databases are known as NoSQL Databases. As the applications are becoming more networked and social, the number of data requests is also increasing and thus NoSQL databases like MongoDB is preferred. Through this paper we attempt to eliminate the use of relational databases and use NoSQL databases. The paper foregrounds one such emerging technology of existing NoSQL databases, that is, MongoDB, which is then compared with a widely used relational database management system, MySQL. Along with, the paper compares MongoDB with an existing NoSQL Database of same category, CouchDB and hence justifies when to prefer MongoDB.*

1. INTRODUCTION

NOSQL DATABASES focus on a particular class of problems that includes flexible data stores to applications that uses graph databases and column databases or key value stores. NoSQL databases represent a distributive force in storing and retrieving the data, applying techniques that have often being ignored as they didn't fit the SQL standards.

NoSQL, also called, Not Only SQL database provides mechanism for storing and retrieving the data that is modelled in a form other than the tabular relations as used in relational databases. These databases motivate simplicity of design, horizontal scaling and finer control over scalability. NoSQL databases are significantly growing in industry in terms of their wide use in big data and real-time web applications. [1]

2. HISTORY

The term NoSQL was first coined by Carlo Strozzi in 1998 for the lightweight, open source relational databases that did not introduced the standard SQL interface[1].

Google built a massively scalable infrastructure for its search engine and other applications within few years leading up to 2004 [2]. These applications included Google Finance, Google Earth, Google Maps, Gmail and Google Apps. The approach was to resolve the issue at entry level of application stack. Google's objective was to build a flexible infrastructure for parallel processing of large amount of data. Therefore, Goggle created a mechanism consisting of distributed file system, a column oriented data store and a MapReduce-based algorithm execution environment. In 2004 Google scientists presented Google File System (GFS), a mode of storing data across distributed machines, and Google MapReduce, a distributed complex, retrieval platform running on top of GFS. Google's effort encourages a lot of open-source developers [2].

The designers of open source search engine were the foremost to develop an open source version which recreated the features of Google's framework. Afterwards, key developers united with Yahoo, where they designed a parallel universe that resembled all the components of Google distributed stack. This open-source substitute is Hadoop. Somewhere, regarding its first releases the idea of NoSql emerged. In 2007, Amazon proposed its concepts of distributed highly accessible and consistent data store named, Dynamo [3].

With the affirmation of NoSQL from two prime web giants: Google and Amazon several new products appeared in the area. Many developers begin experimenting with the concept of using these techniques in their applications and many enterprises became amenable to acquire knowledge about the technology and using it. Within 5 years, NoSQL and its related concepts of big data management became ubiquitous and use cases appeared from many well-known companies like Facebook, Yahoo, EBay, IBM etc.

3. CHARACTERISTICS

- Can handle large data volumes.
- Provides scalable replication and distribution, potentially distributed around the world.
- Queries return answer quickly.
- Schema less.
- ACID transaction properties are not needed-BASE is used.
- Allows to transparently scaling up the network by adding more number of nodes at a very low cost.
- Open source development.
- High throughput.

4. MERITS OF NOSQL DATABASES

A. Big Data Magnification

Big Data is said to be one of the most important factors influencing the growth as well as popularity of NoSQL in businesses. The infinite array of technologies of data collection varying from simplest online actions to GPS tools, from smartphones and tablets to advanced sensors and many more that multiplies the data growth.

One of the primary reasons of using NoSQL is handling the use of Big Data project. A big Data project is normally described as:

1. **Higher data pace-** bulky data entering rapidly from various discretesites.
2. **Data patterns-**data stored can be regulated, well-regulated or unregulated.
3. **Data proportion-**contains numerous gigabytes, terabytes of data capacity.
4. **Data problem-** the data is managed and stored at discrete geographical areas and data centres.

B. Persistent Data Accessibility

NoSQL database has environments that are developed with distributed architecture which leads to no point of failure and also there redundancy integrated for both functions as well as data that are extremely important in hardware failure. If a database server crashes, the other servers in the system continue with the task without losing the data, hence depicting true fault tolerance.

Through this NoSQL database framework provides permanent accessibility across data centres as well as in cloud. When utilized judiciously, NoSQL databases provide high performance at large scale. This is extremely advantageous as the system can be updated without taking the database offline.

C. BASE Property

The BASE acronym is used to describe the properties of NoSQL databases. It is referred to as opposite of ACID.

1. **Basically Available:** The database is basically available, means that, if some part of the database are currently unavailable, other part of the database continue the function.
2. **Soft State:** Data may be time-dependent based on users interaction with some expiration after an interval of time data must be updated or accessed in order to remain relevant in the system.
3. **Eventually Consistent:** Modified data may not become consistent across the entire system immediately but will become consistent with time. So, the data will be consistent in the future.

D. Location Independent

Location independence refers to the ability of reading and writing to the database without considering where that I/O operation are physically performed and to have any write functionality generated from the location, in order to make available at the other sites containing all other users and machines.

E. Elastic Scaling

NoSQL databases are developed in order to expand transparently taking advantage of new nodes, and are usually designed using low cost commodity hardware in mind.

F. Flexible Data Models

The crucial aim that made businesses move to NoSQL database system is the increase in data models flexibility that exists in most NoSQL databases. NoSQL data models are often named as schema-less because they allow the use cases that cannot be modelled using RDBMS. NoSQL databases accept all types of data without any difficulty than RDBMS, which depends on predefined schema. This feature of RDBMS restrained the flexibility as the predefined schema strongly controls the database and its data organization.

Performance factors also come to play, particularly when “wide rows” are concerned along with many updates which can affect their performance. Such kind of situations can be easily handled by NoSQL data models which also provide rapid execution for reading as well as writing operations.

G. Economy

NoSQL databases contain inexpensive servers that manage the rapidly increasing data and high volume transactions. As a result of which the cost of a transaction per second for NoSQL is many times less as compared to that or RDBMS.

5. CHALLENGES FOR NOSQL DATABASES

A. It has a very low focus

NoSQL databases are designed primarily for storage and provide very little functionality except that as a result they are very restricted. As far as transactions are concerned, relational database is still a better choice. Also, NoSQL is not so good with backup.

B. Maturity

RDBMS systems are used since long time. RDBMS systems are highly stable and function well. As compared to this, NoSQL databases are still in production and many its key features are still have to be implemented.

C. Support

Enterprises want affirmation regarding the key system failure, that is, if the main system failed then they will get appropriate support. RDBMS vendors provide a high level support to enterprises.

As compared to this, since the NoSQL systems are open source so there are a lot of firms extending their support for NoSQL databases, but there companies are small scale creations with no global reach and no support for resources.

D. Analytics and business intelligence

Designing NoSQL databases aimed at meeting the growing requirements of current Web applications. Therefore, many features of NoSQL are aligned regarding the requirements of these applications.

Data in an application is of importance in businesses if it goes beyond the inset-delete-update and read cycle. Businesses mine information so as to improve their efficiency well as their competitiveness. NoSQL databases support the feature of ad-hoc queries and analysis. An elementary query requires programming experts but most of the BI tools do not provide NoSQL connectedness.

E. Administration

NoSQL databases were designed to provide support with zero-administration solutions, but the present NoSQL databases lack such goal. NoSQL demands expertise skills in order to install and enough effort to maintain.

F. Experts

Millions of developers across the world are accustomed with the RDBMS concepts. Compared to this NoSQL developers are still learning, it's very easy to find an RDBMS expert with a good experience than a NoSQL expert.

G. Standardization and Open Source

NoSQL databases are "open source", this can be viewed as its greatest strength or its greatest weakness. There are not many definite benchmarks for NoSQL databases so far, that is, it is not possible to create two databases that are equal.

Getting an implementation to work with existing infrastructure can be something random, while this support is non-uniform as compared to traditional databases implemented so far.

6. KINDS OF NOSQL DATABASES

1. Key Value Store

Design of these databases aimed at storing schema-less data. The data consists of a key that contains the index, and a value.

Example: Cassandra, DynamoDB, BerkeleyDB, Riak.

2. Column Store

These are also known as wide-column stores. These databases are designed to store the data in columns rather than rows. This description is counter to the standard databases, column stores enables better performance and also provides with a scalable architecture.

Example: BigTable, HBase and HyperTable etc.

3. Document Databases

It is an extension to the key value stores NoSQL databases, here the "documents" are concerned in which data is stored and every document is allocated a unique value known as key that is used to fetch the required document. Document databases are designed to store, retrieve and manage the document-oriented information.

Examples: MongoDB and CouchDB.

4. Graph Database

This type of NoSQL databases are designed to represent the data in the form of a graph and contain interconnections among the elements, with infinite relations among them.

Example: Neo4J and Polyglot.

7. ARCHITECTURE OF DOCUMENT ORIENTED NOSQL DATABASES

MongoDB is a document-oriented database. MongoDB has originated from the term "humongous". It is schema-less, scalable and is developed using C++ language [4]. It is one of the new NOSQL databases. Many websites like eBay, Foursquare, Viacom has adopted MongoDB as a backend.

MongoDB holds a set of collections. Collections have no predefined schema and stores data as documents. In this context, document in MongoDB is referred to as a set of fields and can be considered as a collection of rows. It can store complicated structures like lists or documents. There are many collections and each collection can have any kind of document and each document has an ID associated with it, which is used as a primary key [5].

The features of MongoDB include load-balancing and replicating data over multiple machines in order to store files because of which it can be used as a file system. Master-slave and replica sets are the two types of replications supported by MongoDB.

Master-Slave replication consists of a master server and a slave server. The master server has full access over the data and that can write every change to the slave servers. The slaves have only read access to the data. On the other hand, replica sets is similar to Master-Slave, but in this there is a mechanism to select a new master server if the original one goes down.

MongoDB is composed of three components named as- Shard Nodes, Configuration servers and Routing services.

Shard Nodes stores the actual data of the database. Each shard contains a node along with its replica which contains the data for that shard [5]. Queries for reading and writing to the data are carried on respective shards.

The replica of the node consists of many servers, among which one server acts as a primary server and all other are secondary servers. If the primary server fails, the work is automatically passed to one of the secondary servers. All the read and write operations are made at the primary server and after that all consistent reads are done by the secondary servers.

Configuration Servers are composed of a group of several servers that makes a cluster. The configuration servers functions to store metadata and routing information of the MongoDB cluster and also indicates which data is present on which shard [5].

Routing Servers functions to handle the client’s request. They performs the tasks that requested by the client. Depending upon the different types of queries issued by the clients, routing servers are responsible for sending the requests to the respective shards and combining the response before sending it to the client. Routing servers are also called mongos. Mongos can run in parallel.

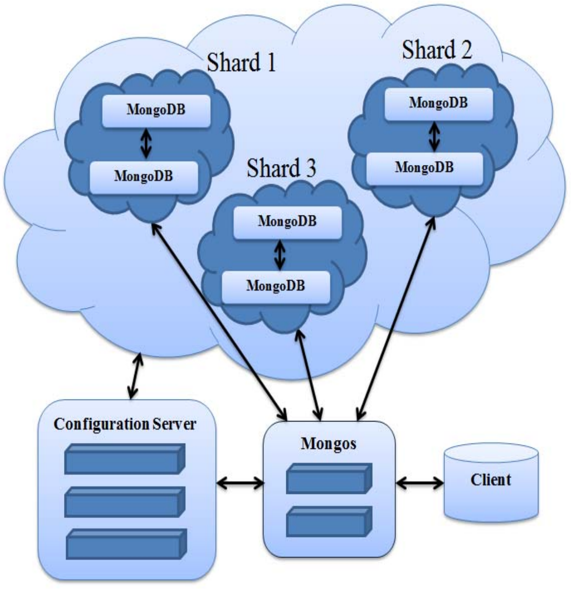

Fig. 1: Architecture of MongoDB

In order to use maximum of the available memory MongoDB makes use of memory-mapped files which in turn boost up its performance. To index the MongoDB database B-Trees are used. In order to partition their collection in MongoDB cluster user specified shard key is used. MongoDB enables horizontal scaling across multiple nodes by supporting auto sharding. Sharding partitions the data over multiple servers by using shard key so as to maintain the order. Thus sustains the shard-rebalancing automatically.

The Fig. below [Fig. 1] describes the architecture of MongoDB consisting of Shards, Configuration Servers and the mongos.

8. COMPARISON BETWEEN MONGODB AND MYSQL

A. Based on Terms/Concepts

Table 1: Comparison based on Concept

SQL Terms/Concepts	MongoDB Terms/Concepts
Data is represented in tables and rows.	Data is represented as a collection of BSON Documents
Uses indices	Uses indices
Tables can be joined	Documents are embedded and linked.
Primary Key is set implicitly	Primary Key is set Explicitly
Definite schema	Schema less
Supports ATOMIC Transactions.	Does not support transaction, but every single transaction is ATOMIC.

B. Based on Schema

Table 2: Comparison based on Schema

SQL Schema	MongoDB Schema
A. Create Command Create Table employee (id MEDIUMINT NOT NULL, emp_id Varchar (30), phone_no Number, designation char(1), PRIMARY KEY(id))	Created with first insert () operation implicitly. [4] db.employee.insert({emp_id:”e123”,phone_no:999548755,designation:”Clerk”})
B. Drop Command DROP TABLE employee	db.employee.drop()
C. Insert Command Insert into employee(emp_id, phone_no, designation) VALUES(“bcd001”,45”A”)	db.employee.insert({emp_id: ”e123”,phone_no:999548755, designation : Clerk”})
D. Select Command SELECT * FROM employee	db.employee.find()
E. DELETE Command DELETE FROM employee WHERE designation =”Manager”	db.employe.remove({ status:”Manager”})

9. COMPARISON BETWEEN MONGODB AND COUCHDB

Table 3: Comparison between CouchDB and MongoDB

CouchDB	MongoDB
Document Oriented – JSON based data model	Document Oriented but based on BSON data model
Uses HTTP/REST Protocol Interface	Uses Custom Protocol over TCP/IP interface
Implemented using Erlang	Implemented using C++
Support Master-slave and master-master Replication	Support Master-master Replication.
CouchDB can operate on BSD, Android, Solaris, Linux and Windows	MongoDB can operate on Linux

10. WHEN TO PREFER MONGODB?

If the data to be model in a relational database system seems to be complex. If de-normalizing the database schema or coding becomes a barrier in performance. If serialized arrays or BSON objects are required to be stored, then it is better to switch off MongoDB. If the schema is not pre-define or the records are to be stored in the same collection that have different fields.

11. CONCLUSION

The evolution and implementations of NoSQL Databases are examined in this document. Various features and challenges faced while developing NoSQL were considered. It was concluded that NoSQL is well-suited to all those databases that do not obey the die-hard RDBMS concept. The NoSQL databases were predominantly classified in 4 categories. The data stores in each category has strengths either at addressing the enterprise or dealing with cloud related concerns like easy to operate or providing higher scalability, availability etc. Each NoSQL database should be used in such a way that it should meet the system requirements.

We also examined the architecture of one of the NoSQL database named, MongoDB in this document. It is one of the new ways to preserve the data; it suits well on small scale projects and is also easy to learn. MongoDB has good performance but in the case where there is a need to process more data that makes use ofcomplicated queries, then another database is preferred. Operations like insertion, deletion as well as updating of data are supported by MongoDB in a good way which makes MongoDB most suited to the projects having simple access to data.

REFERENCES

[1] Dr. S. George, NOSQL-NOTONLY SQL, International Journal of Enterprise Computing and Business System Volume 2, 2 July 2013.

[2] Strozzi, Carlo: NoSQL- A Relational Database Management System, 2008-2010.

[3] Shashank Tiwari, Professional NOSQL, John Wiley & Son, Inc.

[4] Sanobar Khan, Prof.Vanita Mane, SQL SUPPORT OVER MONGODB USING METADATA, International Journal of Scientific and Research Publications, Volume 3, Issue 10, October 2013.

[5] Wumuti Naheman, Jianxin Wei, REVIEW OF NOSQL DATABASES AND PERFORMANCE TESTING ON HBASE, International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC); 12/2013.