International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 7 (2017), pp. 1645-1651

http://www.ripublication.com

NOSQL Database and Its Comparison with RDBMS

Dr. A. B. Raut

Professor & Head, Dept. of CSE,

H.V.P.M's COET, Amravati-444605, India.

Abstract

NOSQL database is a rising trend in world of data storage. NOSQL is said to be alternative to the most widely used relational databases for storing data but as the name suggests, it does not completely replace SQL but compliments it in such a way that they both can exist. In this paper we will discuss about NoSQL database, types of NOSQL database, advantages and disadvantages of NoSQL in comparison with relational databases.

Keywords: NoSQL, Relational Database, Data Storage

INTRODUCTION

Experts say that the world's data is doubling every two years. Today these large scale websites like Google, Facebook, Twitter and even YouTube are generating massive amount of data every day. Therefore efficient storage and retrieval of this data is been a challenging task. Larger amounts of transactions and experimentation result require organized storage solutions and database is the key solution to satisfy this need of storing and retrieving data in an organized manner.

A database is a means of storing information in such a way that data can be stored and retrieved from it whenever required. Since their inception in the 1960's different types have emerged, each using its own representation of data and technology for handling transactions. They began with navigational databases which were based on linked-lists, moved on to relational databases, afterwards object-oriented and recently **NoSQL** emerged and has become a popular trend.

Two of the most widely used database types are relational databases and NoSQL

databases. Although NoSQL databases are relatively new compared to other types they have become popular due to their ability to handle very fast unrelated and unstructured data, mainly because they do not require a fixed schema and they use metadata heavily in order to achieve rapid performance.

A Relational Database is one that presents information in tables with rows and columns. A table is referred to as a relation in the sense that it is a collection of objects of the same type (rows). Data in a table can be related according to common keys or concepts, and the ability to retrieve related data from a table is the basis for the term relational database. A Database Management System (DBMS) handles the way data is stored, maintained, and retrieved.

A NoSQL (originally referring to "not only SQL") database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. NoSQL gained popularity in 21st century and was triggered by the needs of companies such as Facebook, Google and Amazon.com.

Although the two types differ in many aspects depending on the implementation they could be used for similar applications although it is not recommended as one is not meant as an alternative to the other [2].

RELATIONAL DATABASE

Initially data was stored in files. However, as the amount of data increased, it was not convenient to access the data using files. It was a slow and inefficient process. As the amount of data grew, it was very difficult to maintain the data and fetch any record. Hierarchical and Network databases were designed as storage mechanisms but they did not provide a standard method to access the data.

With the need to manage data and the desire for a standard method to access data, SQL came into existence.

Main Focus of RDBMS is on **ACID** properties:

- **Atomicity** – Each transaction is atomic. If one part of it fails, the entire transaction fails (and is rolled back)
- **Consistency** – Every transaction is subject to a consistent set of rules (constraints, triggers, cascades)
- **Isolation** – No transaction should interfere with another transaction
- **Durability** – Once a transaction is committed, it remains committed

ACID is important but only when system is of kind banking, finance, safety systems etc which can be overhead for systems that needs to share data enormously like Google, Amazon etc.

RDBMS doesn't quite fit for some following requirements:

- Distributed
- Scalability
- Control over performance characteristics
- High availability
- Low Latency
- Cheap

Hence to satisfy these needs the concept of NoSQL came into existence.

NoSQL DATABASE

The main motivation behind this approach include: simplicity of design, simpler "horizontal" scaling to clusters of machines, which is a problem for relational databases, and finer control over availability. The data structures used in NoSQL databases (e.g. key-value, graph, or document) differs slightly from those used by default in relational databases, making some operations faster in NoSQL. The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used in NoSQL databases are also viewed as "more flexible" than tables in relational database.

NoSQL databases are increasingly [4] used in big data and real-time web applications. NoSQL systems are also sometimes called "Not only SQL" to emphasize that they may support SQL-like query languages.

3.1 NOSQL DATABASE TYPES:

A. ***Document stores:*** The notion of "documents" is the central concept here with documents being the equivalent of records in relational databases and collections being similar to tables. Document stores includes MongoDB, CouchDB
B. ***Graph stores****: a*re used to store information about networks, such as social connections. Graph stores include Neo4J and HyperGraphDB.
C. ***Key-value*** *stores* are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or "key"), together with its value. Key-value stores includes MemcacheDB, Azure Table Storage..
D. ***Wide-column*** *stores* such as Cassandra and HBase are optimized for queries over large datasets, and store columns of data together, instead of rows. Column stores include Hadoop, Cassandra, and Hypertable.

ARCHITECTURE OF MONGODB

In MongoDB [2], data or records are called documents and the documents are stored in a binary JSON (BSON) format. Records—documents—are further organized in collections. However, in the same collection, the schema of one document can be different from the other. Hence, flexible schemas are supported. Because developers are familiar with JSON formats, using BSON is simpler. Nested data structures and array formats are also supported with MongoDB. To implement the same in a relational DB would mean multiple tables with foreign key relationship. It also supports typed data, such as integer, string, date time, double, and so forth. It supports indexing and querying. A query can vary from key-value, range, or geospatial to aggregation framework queries. Indexes enhance performance when querying the data. Indexes can be declared on unique/single and multiple fields. An index also can be on fields from a nested structure. As with SQL, it also supports verifying the execution plan to optimize query performance. MongoDB provides a replica set: a failover mechanism. There is only one Primary database that allows a write operation and multiple secondary servers only for read operations. A minimum of three servers is required for a replica set: Primary, Secondary and, Arbiter. Arbiter doesn't store any data; it's only used during failover to decide which server will be the next primary server.

STUDY OF DIFFERENCES BETWEEN SQL AND MONGODB

The main reason why one would need to move to NoSQL databases is necessity in huge data storage (also called Big Data), scalability and performance reasons. Here are some tables showing differentiation of terminologies and operations on data between a NoSQL database called MONGODB and RDBMS (SQL)

Table 1. Terminologies in SQL & corresponding in MongoDB

SQL	MONGODB
Database	Database
Table	Collection
Row	document or BSON document
Column	Field
Index	Index
table joins	embedded documents and linking
primary key (specify any unique column or column combinations as primary key)	primary key (the primary key is automatically set to the _id field in MongoDB)
aggregation (e.g. by group)	aggregation pipeline

Table 2. Create statements in SQL & MongoDB

SQL Schema Statements	MongoDB Schema Statements
CREATE TABLE users (id MEDIUMINT NOT NULL AUTO_INCREMENT, user_id Varchar(30), age Number, status char(1),	db.createCollection("users") or db.users.insert({ user_id: "abc123", age: 55, status: "A" })
DROP TABLE users	db.users.drop()

Table 3. Insert statements in SQL & MongoDB

SQL INSERT Statements	MongoDB insert() Statements
INSERT INTO users(user_id, age, status) VALUES("bcd001", 45, "A")	db.users.insert({ user_id: "bcd001", age: 45, status: "A" })

Table 4. Retrieving statements in SQL & MongoDB

SQL SELECT Statements	MongoDB find() Statements
SELECT * FROM users	db.users.find()

ADVANTAGES OF NOSQL

The relational database (RDBMS) has been [8] the dominant model for database management. But, today, non-relational, "cloud," or "NoSQL" databases are gaining mindshare as an alternative model for database management. The various advantages are:

6.1. Scalable

For years, database administrators have relied on *scale up* — buying bigger servers as database load increases — rather than *scale out* — distributing the database across multiple hosts as load increases. However, as transaction rates and availability requirements increase, and as databases move into the cloud or onto virtualized environments, the economic advantages of scaling out on commodity hardware become irresistible. RDBMS might not scale out easily on commodity clusters, but the new breed of NoSQL databases are designed to expand transparently to take

advantage of new nodes, and they're usually designed with low-cost commodity hardware in mind.

1.2. Big data

Just as transaction rates have grown out of recognition over the last decade, the volumes of data that are being stored also have increased massively. O'Reilly has cleverly called this the "industrial revolution of data." RDBMS capacity has been growing to match these increases, but as with transaction rates, the constraints of data volumes that can be practically managed by a single RDBMS are becoming intolerable for some enterprises. Today, the volumes of "big data" that can be handled by NoSQL systems, such as Hadoop, outstrip what can be handled by the biggest RDBMS.

1.3.Economical

NoSQL databases typically use clusters of cheap commodity servers to manage the exploding data and transaction volumes, while RDBMS tends to rely on expensive proprietary servers and storage systems. The result is that the cost per gigabyte or transaction/second for NoSQL can be many times less than the cost for RDBMS, allowing you to store and process more data at a much lower price point.

CONCLUSION

There are few limitations in SQL database:

Scalability: Users have to scale relational database on powerful servers that are expensive and difficult to handle. To scale relational database it has to be distributed on to multiple servers. Handling tables across different servers is a chaos.

Complexity: In SQL server's data has to fit into tables anyhow. If your data doesn't fit into tables, then you need to design your database structure that will be complex and again difficult to handle.

RDBMS is a great tool for solving ACID problems when data validity is crucial, when you need to support dynamic queries.

NoSQL is a great tool for solving data availability problems, when it's more important to have fast data than up-to-the-minute just updated data, when you need to scale based on changing requirements.

Pick the right tool for the job.

REFERENCES

[1] Leavitt, N.,"Will NoSQL Databases Live Up to Their Promise?" Computer, vol.43, no.2, pp.12-14, Feb. 2010 doi: 10.1109/MC.2010.58

[2] MongoDB, http://www.mongodb.org/about/introduction/

[3] Clarence J M Tauro, Aravindh S, Shreeharsha A. B, "Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases", International Journal of Computer Applications (0975 – 888) Volume 48– No.20, June 2012 doi:10.5120/7461-0336

[4] http://en.wikipedia.org/wiki/NoSQL

[5] Sareen Pankaj, Parveen Kumar "NOSQL DATABASE AND ITS COMPARISON WITH SQL DATABASE", International Journal of Computer Science & Communication Networks,Vol 5(5),293-298

[6] Apache Cassandra, http://wiki.apache.org/cassandra/

[7] http://www.gmv.com/blog_gmv/nosql-database-challenge

[8] http://www.dummies.com/how-to/content/10-advantages-of-nosql-over-rdbms.html

[9] Jing Han; Haihong, E.; Guan Le; Jian Du; , "Survey on NoSQL database," Pervasive Computing and Applications (ICPCA), 2011