ORIGINAL RESEARCH PAPER

Engineering

# AN ANALYSIS OF USABILITY OF RDBMS IN CONTRAST WITH NOSQL – RISE OF BIG DATA

**KEY WORDS:** Big Data, NoSQL Database, RDBMS

**Manish Jitendrabhai Vankani**

SOE – MCA Department, RK University, Rajkot (Gujarat).

**ABSTRACT**

In highly dynamic information era, enormous amounts of data have become available to the decision makers. Big data refers to datasets that are not only big, but also high in variety and velocity, which makes them difficult to handle using flat file system or RDBMS. Due to the rapid growth of such data, solutions need to be studied and provided to handle and fetch information from these datasets. Although, decision makers need to be able to find valuable insights from such varied and rapidly changing data such as daily transactions, customer interactions and social network data. Such value can be acquired using big data analytics. This paper intended to provide – features, types and applications of NoSQL databases in Big Data Analytics. This paper is used to help users, especially to the organizations, research scholars, and students to support applications that process large volumes of data.

## INTRODUCTION:

Initially, Flat File systems are used to store the data. Data were stored in flat files and the major drawback with flat files are that each organization employ their own flat files technique and there are no standards to store and retrieve data from files.

RDBMS (Relational Database Management System) is a DBMS based on Dr. E.F.Codd's relational model at IBM's research lab in 1970. The traditional RDBMS is used to store and manipulate all structured data very efficiently. The data is stored in the form of rows and columns in RDBMS. SQL (Structured Query Language) is a programming language used to perform retrieval of data from the 0database. Some of the most popular RDBMS that uses SQL are: Sybase, Oracle, MySQL, Postgres, Microsoft SQL Server, etc.

NoSQL (either non SQL or Non Relational) is a DBMS which was first used by Carlo Strozzi in 1998. It is currently a hot topic for specific type of computing such as to store large structured, semi structured or unstructured data. NoSQL databases are designed to fulfil the big data storage and processing requirements. NoSQL databases have various advanced features compare to traditional RDBMS features. That's why the NoSQL databases are also known as "Not Only SQL" databases. The data is stored by NoSQL databases are either in key-value or graph or column or document form. That types of storage makes the retrieval of data from the database more flexible. It also makes certain operations on data much faster.

**Table 1: Comparison of RDBMS and NoSQL**

| Sr. No. | NoSQL | RDBMS |
|---|---|---|
| 1 | It is used to store large structured, semi structured and unstructured data. | It is used to store the structured data. |
| 2 | The storing of data does not depend on the physical memory of the system. | The data stored on the physical memory of the system. |
| 3 | A transaction must follow any two properties of CAP theorem (Consistency, Availability and Partition tolerance) in NoSQL. | A transaction must follow ACID (Atomicity, Consistency, Isolation, Durability) property in RDBMS. |
| 4 | NoSQL can handle billions of records. | RDBMS can handle few thousands of records. |
| 5 | No need of Normalization. | Table must be normalized. |
| 6 | Data can be scaled vertically as well as horizontally in NoSQL. | Data only can be scaled vertically in RDBMS. |
| 7 | NoSQL databases are cheap compare to RDBMS databases. | RDBMS databases are bit costly compare to NoSQL databases. |
| 8 | NoSQL databases are faster to process. | RDBMS databases are bit slow to process compare to NoSQL databases. |
| 9 | Most NoSQL databases are schema less. | Traditional RDBMS uses strict schema of database design. |
| 10 | MongoDB, HBase, Cassandra, Neo4j etc. are examples of NoSQL. | Oracle, Sybase, MySQL, SQL Server etc. are examples of RDBMS. |

### Characteristics of NoSQL Databases:

To ensure the integrity of data, most of the traditional database systems are based on transactions. It guarantees the consistency of data in different situations for data management. These transactional characteristics are also known as ACID (Atomicity, Consistency, Isolation and Durability). Although, scaling out of ACID system has shown to be a problem. Eric Brewer introduces the CAP theorem which states that it is impossible for a distributed database to simultaneously provide guarantee for more than two out of three properties namely Consistency, Availability and Partition Tolerance. The NoSQL databases can be categorized as under according to CAP theorem:

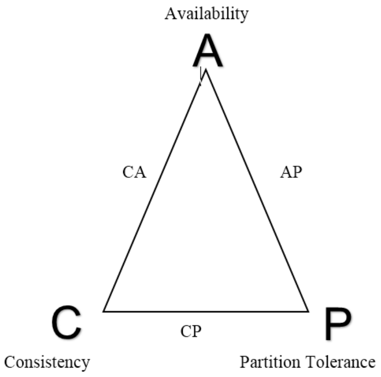

**Figure 1: Eric Brewer's CAP Theorem15**

### 1) Consistency and Partition Tolerance (CP)

The databases ensure the consistency and the system is partitioned across network boundaries are fall under this category. The consistent data is stored in distributed nodes but database support for availability is not good.
For ex.: MongoDB, Redis, BigTable etc.

### 2) Availability and Partition Tolerance (AP)

The databases ensure the availability and the system is partitioned across network boundaries are fall under this category. The data is stored in distributed nodes and available for processing but data is not consistent.

For ex: CouchDB, DynamoDB, Cassandra, etc.

### 3) Consistency and Availability (CA)

The databases ensure consistency and availability are fall under this category. Here, databases mainly use replication approach to ensure data consistency and availability. The database is not focus on partition tolerance.
For ex.: GreenPlum, ElasticSearch, Vertica, etc.

### Types of NoSQL Databases:

NoSQL Databases can broadly be categorized into following four types:

### 1. Key-Value store NoSQL database

The Key-Value store is the simplest NoSQL data stores to use. It uses the Associative Array as their fundamental data model. In this model, the data is represented as collection of key-value pairs in such a way that each key appears at least once in the collection. Some databases support ordering of keys in this category. There are various hardware deployment needed, out of which some users maintain data in RAM while others implement HDD.

Use cases: Following are some use cases of key-value databases:
- Storing and retrieving user's session information for web application
- Storing user profiles and preferences
- Storing user's shopping cart data for online stores

Examples of Key-value: Dynamo, Riak, Redis, HyperDex, Dynomite, Aerospike, Voldemorte, CouchBase, FoundationdB, MemcacheDB, OrientDB, Azure Table Storage(ATS), Berkeley DB etc.

### 2. Document store NoSQL database

In Document based databased data are stored as documents and organized as a collection of documents. The documents are flexible. Each document can have any number of fields. These are designed for storing, retrieving and managing document-oriented information. Document stores offer great performance and horizontal scalability.

The documents are of standard formats such as XML, PDF, JSON etc. In relational database, a record inside the same database will have same data fields and the unused data fields are kept empty, but in case of document stores, each document may have similar as well as dissimilar data. Documents in the database are addressed using a unique key that represents that document. These keys may be a simple string or a string that refers to URI or path.

Document stores are more complex as compared to key-value stores as they allow to include the key-value pairs in document also known as key-document pairs.

Use cases: Following are some use cases of key-value databases:
- E-commerce platforms
- Content management systems
- Analytics platforms
- Blogging platforms

Examples: MongoDB, Apache CouchDB, Elasticsearch, MarkLogic, CouchBase, RethinkDB, Clusterpoint, DocumentDB, OrientDB, etc.

### 3. Column based NoSQL database

These databases organize the data in tables, very much similar to the RDBMS. Although, they store the content by columns instead of rows. Column-based databases do not store data in tables but store the data in distributed architecture. In column-based database, each key is associated with one or more attributes.

These databases are also known as BigTable clones or Columnar databases. A column database stores its data in such a mane that it can be aggregated rapidly with less I/O activity. It provides very high performance and highly scalable architecture. They can be useful for data warehousing, data mining and analytics application.

Use cases: Following are some use cases of key-value databases:
- Blogging Platforms
- Content Management Systems
- Systems that maintain counters
- Services that have expiring usage
- Systems that require heavy write requests

Examples: Cassandra, Apache Hadoop Hbase, Hypertable, Accumulo, Druid, Vertica, etc.

### 4. Graph based NoSQL database

These databases apply the computer science graph theory for storing and retrieving data. They focus on interconnectivity of different parts of data. Units of data are visualized as nodes and relationships among them are defined by edges connecting the nodes. The Graph databases can navigate millions of records using index free adjacency technique, in which every node consist a direct pointer which points to the adjacent node.

Graph databases provides schema less and efficient storage of semi structured data. Graph databases are ACID compliant and offer rollback support. They can be very powerful when your data is highly connected and related in some way.

Graph databases can be used for a variety of applications like social networking applications, bioinformatics, content management, security and access control, network and cloud management etc.
Use cases: Following are some use cases of key-value databases:

- Fraud Detection
- Graph based search
- Network and IT operations
- Social Network

Examples: Neo4J, OrientDB, Giraph, AllegroGraph, MarkLogic, InfiniteGraph, Virtuso, Stardog, etc.Future Research Directions for NoSQL:

The future work will focus on benchmarking and performance of several NoSQL databases available with respect to the size of datasets as well as the different type of the data sets.

There are large number of companies using NoSQL. They are as under:

- Google
- Facebook
- LinkedIn
- Adobe
- Foursquare
- Mozila
- McGraw-Hill Education

## CONCLUSION:

Choosing between a NoSQL and a traditional relational database is always going to come down to your company's particular needs. And there are, of course, situations for which you might want to use both types, as they can often complement each other.

If you deal with a lot of data types, and/or you want or need to build powerful web and cloud applications for a distributed and quickly growing user base, then you will need your database to be multi-model, flexible, easily scalable, distributed and always on, which means you will need a NoSQL database that can handle those requirements.

## REFERENCES:

[1] http://en.wikipedia.org/wiki/NoSQL.
[2] Use relational DBMS, N. Saying good-bye to DBMSs, designing effective interfaces, Communications of the ACM, vol. 52, no. 9, (2009).
[3] N. Leavitt, "Will NoSQL databases live up to their promise?", Computer, vol. 43, no. 2, (2010), pp. 12-14.
[4] Find InfiniteGraph from: http://objectivity.com/.
[5] Find Neo4j from: http://www.neo4j.org/.

[6] Find MongoDB from: http://www.mongodb.org/.
[7] Find Hypertable from: http://hypertable.org/.
[8] Find Riak from: http://basho.com/products/riak-overview/.
[9] Find Redis from: http://redis.io/.
[10] Find DynamoDB from: http://aws.amazon.com/dynamodb/.
[11] Find SimpleDB from: http://aws.amazon.com/simpledb/.
[12] Find Cassandra, Accumulo, Hbase, CouchDB from: http://www.apache.org/.
[13] ACID detail from web: http://en.wikipedia.org/wiki/ACID.
[14] Microsoft SQL Server Databases from web: http://www.microsoft.com/en-us/sqlserver/default.aspx.
[15] Brewer's CAP Theorem, By Julian Browne on January 11, 2009 web: http://www.julianbrowne.com/article/viewer/brewers-cap-theorem.