ISSN 2347-3983

International Journal of Emerging Trends in Engineering Research, Vol.3. No.10, Pages : 362 - 367 (2015)
Special Issue of ICACSSE 2015 - Held on October 30, 2015 in St. Ann's College of Engineering & Technology, Chirala, AP, India
http://www.warse.org/IJETER/static/pdf/Issue/icacsse2015sp75.pdf

New Era of NoSQL Databases

ANIL KUMAR JAMPANA
Emirates Group IT Department
Email: ANILK.JAMPANA@GMAIL.COM , ANIL.JAMPANA@EMIRATES.COM

Abstract:

This article provides an overview of NoSQL databases. The digital data is dramatically growing in each minute. NoSQL databases were designed to store and process massive volumes of complex data required in real-time web applications. There are many NoSQL types with different performances, and thus it is important to compare them in terms of performance and verify how the performance is related to the database type. This comparison allows users to choose the most appropriate database according to the specific mechanisms and application needs.

1. Introduction:

Databases are playing vital role in the organizations. Relational databases having various limitations like storage, scalability, and varieties of data. To overcome these limitations, a new database model was developed with new features known as NoSQL.

The definition of NoSQL "non SQL" or "non-relational" or "not only SQL". NoSQL is database that provides a mechanism for storage and retrieval of data.This is modeled other than the tabular relations used in relational databases.It's not a replacement for a RDBMS but compliments it.

NoSQL databases are those databases that are non-relational, open source, distributed in nature as well as it is having high performance in a linear way that is horizontally scalable. Non-relational database does not organize its data in related tables (i.e., data is stored in a non-normalized way).

Symbolic representation of NoSQL

2. Need for NoSQL:

NoSQL (non-relational) is comparatively faster than relational databases. Previously, in SQL, we were using Query language to fetch as well as to store data; for NoSQL we store large data entities using documents in XML formats. XML language is basically used to store structured data in a human readable form.NoSQL databases refer to progressive data management engines that go beyond legacy relational databases in satisfying the needs of today's modern business applications.These NoSQL databases are schema-free, support easy replication, have simple API, eventually consistent, and can handle huge amounts of data.

3. Challenges of NoSQL Databases:

NoSQL solutions are built with specific goals and purposes. These are vertically scalable, high available, ease of development, and high performance.

- **Scalability:** Relational databases are scales vertically – to add capacity (data storage or I/O capacity) we need a bigger server. NoSQL databases are horizontally scalable.In NoSQL Instead of buying a bigger server, we use many commodity servers, virtual machines or cloud instances.
- **Availability:**Since data is stored in a distributed fashion, network failures are common when committing updates. So, it must be able to recover from lost data commits and provide up-to-date data access in acceptable range of latency. This solution must do online upgrades to easily remove a node for maintenance, without

affecting the availability of cluster to handle online operations.

- **Ease of development:** Relational databases require a rigid scheme and, if you want to change the application, you must change the database schema, too. However NoSQL databases have the following advantages: Flexible schema, Simple query language
- **High performance:** The database should provide consistently low latency regardless of task or data size. In general, reading and writing latency of NoSQL databases is very low, because the data is shared between all nodes in a cluster, while the application's working set is in memory. NoSQL solution is built for content management, data management and computing intensive applications, high latency for any Read/Write operations is not accepted.

4. Characteristics of NoSQL databases:

- ➢ NoSQL Databases does not use the relational data model and SQL language.
- ➢ NoSQL Databases stores large volumes of data.
- ➢ NoSQL Databases are supporting distributed environment (spread data to different machines), we use NoSQL without any inconsistency.
- ➢ NoSQL Databases will provide high availability. If any faults or failures exist in any machine, then in this there will be no discontinuation of any work.
- ➢ NoSQL Databases are open source databases, i.e. its source code is available to everyone and is free to use it without any overheads.
- ➢ NoSQL Databases allows data to store in any record that is it is not having any fixed schema.
- ➢ NoSQL Databases does not use concept of ACID properties.
- ➢ NoSQL Databases are horizontally scalable leading to high performance in a linear way.
- ➢ NoSQL Databases won't use joins. Since joins are expensive.

5. The difference between NoSQL and relational database:

The below points are differentiate a relational database from a NoSQL database.

Relational Database	NoSQL Database
It has a fixed schema.	No fixed schema.
Supports transactions.	Does not support transactions.
Supports powerful query language.	Supports very simple query language.
These databases are table based databases.	These databases are document based, key-value pairs, graph databases or wide-column stores
These databases are vertically scalable.	These databases are horizontally scalable.
Primarily structured data	Structured, with semi/unstructured
Moderate velocity data	High velocity data (devices, sensors, etc.)
Allows joins	There is no joins
Examples: MySql, Oracle, Sqlite, Postgres and MS-SQL	Examples: MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb

6. Architecture of NoSQL:

The Architecture of NoSQL is basically divided into two parts.

1) Components and 2) Data Storage

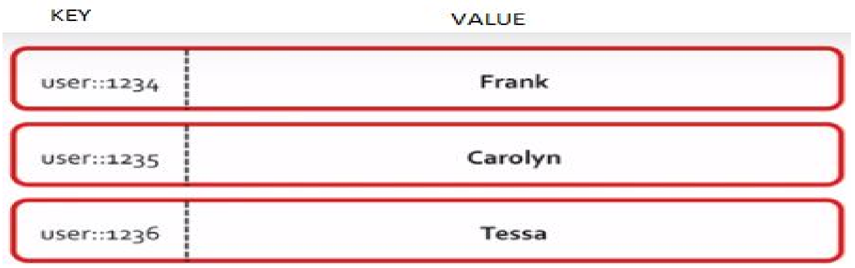
Components:

1) Modelling Language: It describes the structure of the database and also defines schema on which it is based. Data is stored in the form of rows and columns using XML formats. And each data (value) corresponding to it is assigned a key that is unique in nature. For faster access of data, the model is built in a suitable environment.
2) Database Structure: Each and every database while building uses its own data structures and stores data using permanent storage device.
3) Database Query language: All the operations are performed on the database that are creation, updation, read and delete (CURD).
4) Transactions: During any transaction in the data, there may be any type of faults or a failure; then, the machine will not stop working.

NoSQL data store types:

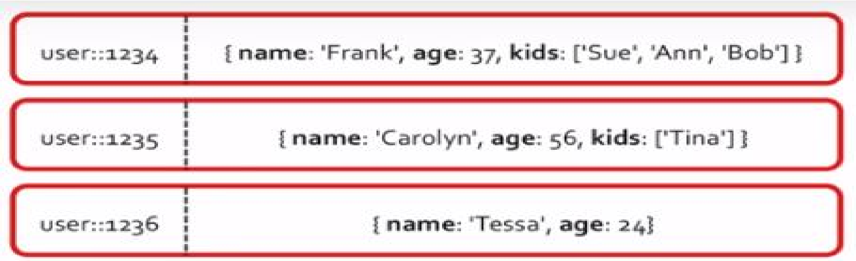
Based on the way data stores these NoSQL databases are mainly divided into four categories. Viz.,

1) Key value databases: (Dynamo, Voldemort, Rhino DHT…)
The key value databases name itself states that it is a combination of two things that is key and a value.
Key is a unique identifier to a particular data entry.
Value is a kind of data that is pointed by a key.
The below mentioned figure shows the Key value database



KEY	VALUE
user::1234	Frank
user::1235	Carolyn
user::1236	Tessa

2) Document Stores Databases: (CouchDB, MongoDB, Lotus Notes, Redis …)
This type of database store unstructured (text) or semi-structured (XML) documents which are usually hierarchal in nature.Each database residing in the document stores points to its fields using pointers as it uses the technique of hashing. Document Stores Databases are schema free and are not fixed in nature.
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



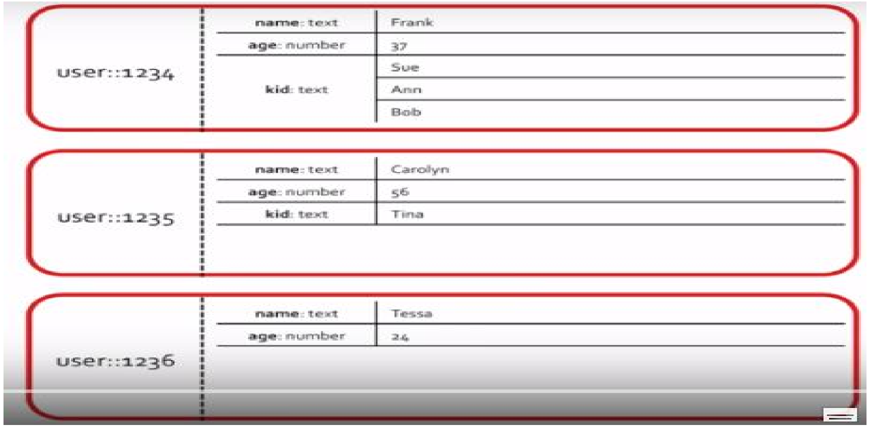
user::1234 { **name**: 'Frank', **age**: 37, **kids**: ['Sue', 'Ann', 'Bob'] }

user::1235 { **name**: 'Carolyn', **age**: 56, **kids**: ['Tina'] }

user::1236 { **name**: 'Tessa', **age**: 24}

3) Columnar Databases: (BigTable, Cassandra, HBase …)
Store and process very large amounts of data distributed over many machines.
Keys point to multiple columns.

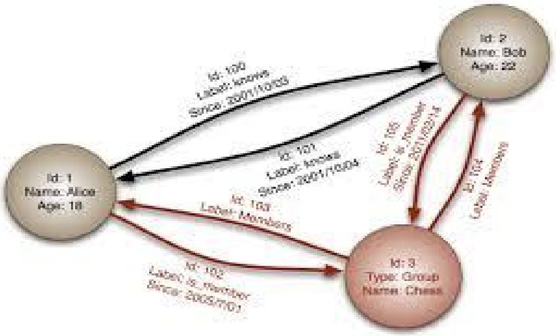
(1) Columnar databases are faster than row based databases while querying.
(2) In columnar databases, assignment of storage unit is done to each and every column.
(3) In the columnar DB only the required columns are read, so reading is faster in this case.



user::1234	**name**: text	Frank
	age: number	37
	kid: text	Sue Ann Bob
user::1235	**name**: text	Carolyn
	age: number	56
	kid: text	Tina
user::1236	**name**: text	Tessa
	age: number	24

4) Graph databases: (Neo4J, FlockDB, GraphBase, InfoGrip …)

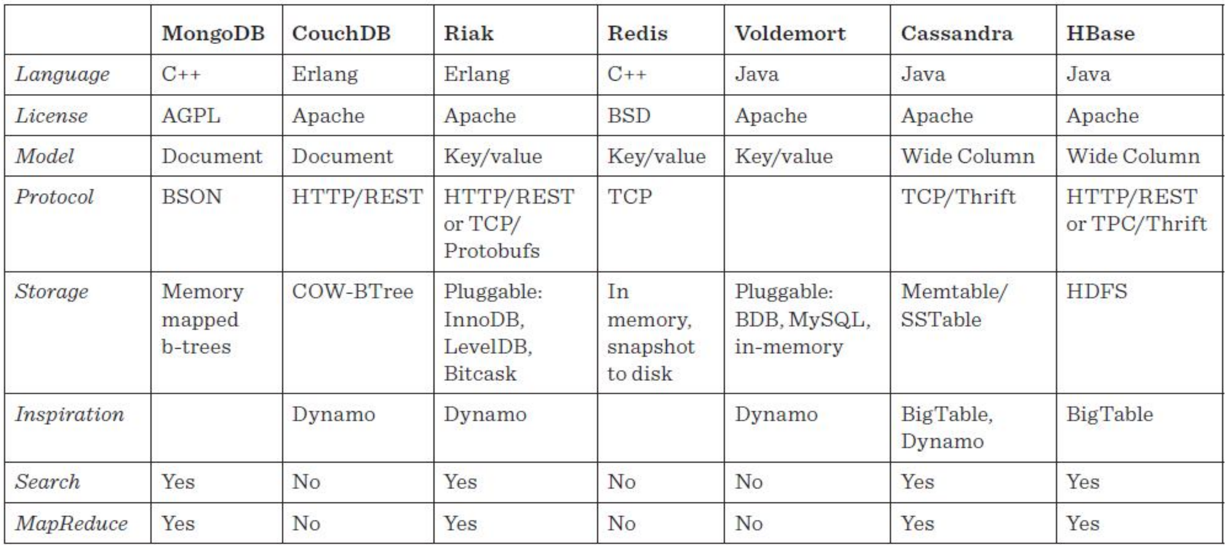
Graph Databases are built with nodes, relationships between nodes (edges) and the properties of nodes.
(1)Nodes represent entities (2) Properties represent attributes (3) Edges represent relationships



7. How NoSQL Databases Differ From Each Other?

There are a variety of different NoSQL databases on the market with the key differentiators between them being the following:

- Architecture: Some NoSQL databases like MongoDB are architected in a master/slave model in somewhat the same way as many RDBMS's. Others (like Cassandra) are designed in a 'masterless' fashion where all nodes in a database cluster are the same. The architecture of a NoSQL database greatly impacts how well the database supports requirements such as constant uptime, multi-geography data replication, predictable performance, and more.
- Data Model: NoSQL databases are often classified by the data model they support. Some support a wide-row tabular store, while others sport a model that is either document-oriented, key-value, or graph.
- Data Distribution Model: Because of their architecture differences, NoSQL databases differ on how they support the reading, writing, and distribution of data. Some NoSQL platforms like Cassandra support writes and reads on every node in a cluster and can replicate / synchronize data between many data centers and cloud providers.
- Development Model: NoSQL databases differ on their development API's with some supporting SQL-like languages (e.g. Cassandra's CQL).



	MongoDB	CouchDB	Riak	Redis	Voldemort	Cassandra	HBase
Language	C++	Erlang	Erlang	C++	Java	Java	Java
License	AGPL	Apache	Apache	BSD	Apache	Apache	Apache
Model	Document	Document	Key/value	Key/value	Key/value	Wide Column	Wide Column
Protocol	BSON	HTTP/REST	HTTP/REST or TCP/ Protobufs	TCP		TCP/Thrift	HTTP/REST or TPC/Thrift
Storage	Memory mapped b-trees	COW-BTree	Pluggable: InnoDB, LevelDB, Bitcask	In memory, snapshot to disk	Pluggable: BDB, MySQL, in-memory	Memtable/ SSTable	HDFS
Inspiration		Dynamo	Dynamo		Dynamo	BigTable, Dynamo	BigTable
Search	Yes	No	Yes	No	No	Yes	Yes
MapReduce	Yes	No	Yes	No	No	Yes	Yes

The above figure shows the differences between various NoSQL databases.

8. Benefits with NoSQL Databases:

- Always On Architecture: Constant uptime that keeps you always open for business.
- Elastic Scalability: Scale online for any sized workload.
- Fast Linear-Scale Performance: Predictable fast performance for Web, mobile and IOT applications.

- Easy Data Distribution: The gold standard in distributing data across multiple data centers, geographies, and cloud providers.
- Flexible Data Management: Easily handles all modern data types.
- Operational Simplicity: Hundreds of nodes are simple to operate and manage.
- Agile Development: Fast time to market for new Web, mobile and IOT applications.
- Cost reductions: Since these database will work on commodity hardware.
- Time reductions: These databases are faster so the processing time will be reduced.

9. Conclusion:

The popularity of NoSQL databases has increased tremendously. The amounts of data are being stored and processed today are increased rapidly. To cater this huge amount of data, organisations are looking for NoSQL databases. These databases havingnumerous advantages, compared to relational databases, especially for large volumes of data, high velocity and different varieties of data (i.e., structured, un-structured and semi structured).

The purposes of this paper –The characteristics (features and benefits of NoSQL databases); differences between RDBMS and NoSQL databases; classification of NoSQL types (categories four and their features); This study report motivation to provide an independent understanding of the strengths and weaknesses of various NoSQL database approaches for supporting applications that process huge volumes of data; as well as to provide a global overview of this non-relational NoSQL databases. Choosing a NoSQL database suitable for a particular application can be complicated because not all NoSQL solutions are the same. Each solution is optimized for different workloads and different use cases. Therefore, each has its own advantages and disadvantages. However, the most important things to consider when working with large volumes of data are: latency, efficiency, availability, horizontal scaling and ease of development.

10. References:

1) NoSQL - http://nosql-database.org/.
2) Datastax Documentation: http://docs.datastax.com/en/cassandra/2.1/cassandra/gettingStartedCassandraIntro.html
3) Couchbase : http://www.couchbase.com/nosql-resources/what-is-no-sql
4) Apache. Hbase: http://hbase.apache.org
5) Graph Databases, NOSQL and Neo4j from: http://www.infoq.com/articles/graph-nosql-neo4j
6) MongoDB, NoSQL Databases (http://www.mongodb.com/nosqlexplained)
7) Highly Scalable Blog,NoSQL Data Modeling Techniques (https://highlyscalable.wordpress.com/2012/03/01/nosql-data-modeling-techniques/)

Author Biography:

Anil Kumar Jampana is a Sr. Database Developer with 14 years of IT experience. He is currently working in Emirates group IT department, UAE. He has evaluated and studied different NoSQL databases, focusing on their performance comparison and characteristics.His main research fields are big data, business intelligence, database knowledge management, and NoSQLand databases performance evaluation. He has published many articles on database in the working organisation.