*International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868*
*Foundation of Computer Science FCS, New York, USA*
*Volume 5– No.4, March 2013 – www.ijais.org*

# Type of NOSQL Databases and its Comparison with Relational Databases

Ameya Nayak
Dept. of Computer Engineering
Thakur College of Engineering and Technology
University of Mumbai

Anil Poriya
Dept. of Computer Engineering
Thakur College of Engineering and Technology
University of Mumbai

Dikshay Poojary
Dept. of Computer Engineering
Thakur College of Engineering and Technology
University of Mumbai

## ABSTRACT
NOSQL databases (commonly interpreted by developers as 'not only SQL databases' and not 'no SQL') is an emerging alternative to the most widely used relational databases. As the name suggests, it does not completely replace SQL but compliments it in such a way that they can co-exist. In this paper we will be discussing the NOSQL data model, types of NOSQL data stores, characteristics and features of each data store, query languages used in NOSQL, advantages and disadvantages of NOSQL over RDBMS and the future prospects of NOSQL.

## General terms
NOSQL, relational databases, data stores

## Keywords
ACID, BASE properties, CAP theorem, DBaaS, scalability

## 1. INTRODUCTION
The problem with relational model is that it has some scalability issues that is performance degrades rapidly as data volumes increases. This led to the development of a new data model i.e. NOSQL. Though the concept of NOSQL was developed a long time ago, it was after the introduction of database as a service (DBaaS) that it gained a prominent recognition. Because of the high scalability provided by NOSQL, it was seen as a major competitor to the relational database model. Unlike RDBMS, NOSQL databases are designed to easily scale out as and when they grow. Most NOSQL systems have removed the multi-platform support and some extra unnecessary features of RDBMS, making them much more lightweight and efficient than their RDMS counterparts. The NOSQL data model does not guarantee ACID properties (Atomicity, Consistency, Isolation and Durability) but instead it guarantees BASE properties (**B**asically **A**vailable, **S**oft state, **E**ventual consistency).It is in compliance with the CAP (Consistency, Availability, Partition tolerance) theorem.

## 2. TYPES OF NOSQL
NOSQL can be categorized into 5 types

## 2.1 Key-Value Store Databases
The key-value data stores are pretty simplistic, but are quiet efficient and powerful model. It has a simple application programming interface (API). A key value data store allows the user to store data in a schema less manner. The data is usually some kind of data type of a programming language or an object. The data consists of two parts, a string which represents the key and the actual data which is to be referred as value thus creating a 'key-value' pair. These stores are similar to hash tables where the keys are used as indexes, thus making it faster than RDBMS Thus the data model is simple: a map or a dictionary that allows the user to request the values according to the key specified. The modern key value data stores prefer high scalability over consistency. Hence ad-hoc querying and analytics features like joins and aggregate operations have been omitted. High concurrency, fast lookups and options for mass storage are provided by key-value stores. One of the weaknesses of key value data sore is the lack of schema which makes it much more difficult to create custom views of the data.

Key value data stores can be used in situations where you want to store a user's session or a user's shopping cart or to get details like favourite products. Key value data stores can be used in forums, websites for online shopping etc. Although key-value data stores existed for long time ago, the development of large number of recent key value data store was influenced by the introduction of Amazon's Dynamo. Some notable DBaaS providers using key-value data stores are mentioned below.

### *2.1.1 Amazon DynamoDB*
Amazon DynamoDB is a newly released fully managed NOSQL database service offered by Amazon that provides a fast, highly reliable and cost-effective NOSQL database service designed for internet scale applications. It is implemented using Amazon's Dynamo model. It offers low, predictable latencies at any scale. It stores data on solid state drives (SSD) instead of traditional hard drives thus providing faster access to the data. The data is replicated synchronously across multiple AWS Availability Zones in an AWS Region to provide built-in high availability and data durability. It replicates data across at least three data centers, thus providing high availability and durability even under complex failure scenarios.

### *2.1.2 RIAK*
Riak is a distributed, fault tolerant, open source database developed by Basho technologies using C, Erlang and JavaScript. It implements principles from Amazon's Dynamo paper. It has a flexible data schema. It offers high availability, partition tolerance and persistence. Components of Riak are Riak Clients, Webmachine, Protocol Buffers, Riak Replication, Riak SNMP/JMX, Riak KV, Riak Search, Riak Pipe and Riak Core

It can be used for following purposes

- Managing personal information of the user for social networking websites or MMORPGs(Massively Multiplayer Online Role Playing Games)
- To collect checkout or POS(Point of sales) data
- Managing Factory control and Information systems
- Building Mobile Applications on cloud etc

Riak should be avoided for highly centralized data storage projects with fixed, unchanging data structures. Riak is used by Mozilla, AOL and Comcast.

## 2.2 Column-Oriented Databases

Column stores in NO SQL are actually hybrid row/column store unlike pure relational column databases. Although it shares the concept of column-by-column storage of columnar databases and columnar extensions to row-based databases, column stores do not store data in tables but store the data in massively distributed architectures. In column stores, each key is associated with one or more attributes (columns). A Column store stores its data in such a manner that it can be aggregated rapidly with less I/O activity. It offers high scalability in data storage. The data which is stored in the database is based on the sort order of the column family.

Column oriented databases are suitable for data mining and analytic applications, where the storage method is ideal for the common operations performed on the data. Some of the notable DBaaS providers using column oriented Databases are mentioned below.

### *2.2.1 Big Table*

Google's Big Table is a compressed high performance database which was initially released in 2005 and is built on the Google File System (GFS). It was developed using C and C++. It offers consistency, fault tolerance and persistence. It is designed to scale across thousands of machines and it is easy to add more machines to it. The Big Table implementation has three major components: a library that is linked into every client, one master server, and many tablet servers. Tablet servers are used to manage a set of tablets (same as tables in RDBMS). The master server handles schema changes, performs tasks like assigning tablets to tablet servers, balancing tablet server load, garbage collection etc. Big Table is not distributed outside Google, but it is available as a part of Google app engine. Big Table is used by a number of Google applications such as Gmail, YouTube and Google Earth.

### *2.2.2 Cassandra*

Cassandra was developed by Apache Software Foundations and was released in 2008. It was developed using Java. It is based on both Amazon's Dynamo model and Google's Big table. Thus it involves concepts of both key-value stores and column stores. It offers feature like high availability, partition tolerance, persistence, high scalability etc. It has a dynamic schema. It can be used for a variety of applications like social networking websites, banking and finance, real time data analytics, online retail etc. Cassandra is being used by Adobe, Digg, eBay, Twitter etc. The disadvantage of Cassandra is that reads are comparatively slower than writes.

## 2.3 Document Store Databases

Document Store Databases refers to databases that store their data in the form of documents. Document stores offer great performance and horizontal scalability options. Documents inside a document-oriented database are somewhat similar to records in relational databases, but they are much more flexible since they are schema less. The documents are of standard formats such as XML, PDF, JSON etc. In relational databases, 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 slightly more complex as compared to key-value stores as they allow to encase the key-value pairs in document also known as key-document pairs.

Document oriented databases should be used for applications in which data need not be stored in a table with uniform sized fields, but instead the data has to be stored as a document having special characteristics. Document stores serve well when the domain model can be split and partitioned across some documents. Document stores should be avoided if the database will have a lot of relations and normalization. They can be used for content management system, blog software etc. Some notable DBaaS providers using document data stores are mentioned below.

### *2.3.1 MongoDB*

MongoDB was developed by 10gen and was initially released in 2009. It was developed using C++. It is a high performance and efficient database. It provides features like consistency fault tolerance, persistence. MongoDB provides additional features like aggregation, ad hoc queries, indexing, auto sharding etc. In MongoDB the documents are mainly stored in BSON (Binary JSON) format. BSON documents contain an ordered list of elements consisting of field name, type and value. BSON is efficient both in storage space and scan speed when compared to JSON. MongoDB uses GridFS as a specification for storing large files. MongoDB is well suited for applications like content management systems, archiving, real time analytics etc. MongoDB is currently being used by MTV networks, Foursquare, The Guardian etc. It is also being used in projects like CERN's LHC, UIDAI Aadhaar which is India's unique identification project. The disadvantages are that it can be unreliable and indexing takes up lot of ram.

### *2.3.2 CouchDB*

CouchDB was developed by Apache software foundation and was initially released in 2005. It was developed using C++. It uses JSON documents to store data and provides RESTful HTTP API to create and update database documents. It provides JavaScript as a query language. It provides a built in web application called FULTON which can be used for administration. It is highly available, fault tolerant and persistent. It implements Multi-Version Concurrency Control (MVCC) thus providing concurrent access to users. CouchDB has great replication and synchronization capabilities. It can be used for applications involving occasionally changing data on which pre-defined queries have to be used. It can be used in cases where network connection may or may not be available, but the application must keep on working, like in the case of mobile device based applications. It can be used for CRM (Customer Relationship Management) and CMS systems. CouchDB is being used by websites like

LotsOfWords.com and friendpaste.com also by facebook apps like Horoscope, Birthday Greeting Cards etc. Some of the drawbacks of CouchDB are temporary views in CouchDB on large datasets are really slow, not good at dealing with relational data, no support for ad-hoc queries.

## 2.4 Graph Databases

Graph databases are databases which store data in the form of a graph. The graph consists of nodes and edges, where nodes act as the objects and edges act as the relationship between the objects. The graph also consists of properties related to nodes. It uses a technique called index free adjacency meaning every node consists of a direct pointer which points to the adjacent node. Millions of records can be traversed using this technique. In a graph databases, the main emphasis is on the connection between data. Graph databases provides schema less and efficient storage of semi structured data.. The queries are expressed as traversals, thus making graph databases faster than relational databases. It is easy to scale and whiteboard friendly. Graph databases are ACID compliant and offer rollback support.

Graph databases can be used for a variety of applications like social networking applications, recommendation software, bioinformatics, content management, security and access control, network and cloud management etc. It is very difficult to achieve 'sharding' in Graph databases. Graph databases are difficult to cluster. Neo4j is one of the notable DBaaS provider using graph data stores.

### *2.4.1 Neo4j*

MongoDB was developed by Neo Technology and was initially released in 2007. It was developed using Java. It is a high performance graph database which provides object oriented, flexible network structure. It is based on a Property graph data model which comprises of nodes and relationship along with their properties. It is reliable, ACID compliant, highly available and scalable. It offers REST interface and Java API quiet convenient to use. It can also be embedded into jar files. It uses CYPHER as its query language. Neo4j must be used in software involving complex relationships like social networking, recommendation engines etc. Sharding is not possible in Neo4j. Neo4j must be avoided if relationships do not exist among the data. Some of the fortune 500 companies that use Neo4j are Adobe, Accenture, Cisco, Lufthansa, Telenor and Mozilla.

## 2.5 Object Oriented Databases

An object oriented database is a database in which the data or the information to be stored is represented as an object (similar to an object used in the concept of object oriented programming language). Thus object oriented database can be considered as a combination of object oriented programming (OOP) and database principles. Object data store offers all the features of OOP such as data encapsulation, polymorphism and inheritance. The class, objects, and class attributes in such databases are comparable to a table, tuple and columns in a tuple in RDBMS respectively. Each object has an object identifier which can be used to uniquely represent that object. Access to data is faster in case of object oriented databases because object can be directly retrieved using pointers. Object oriented databases makes modern software development processes easier to be agile.

Object oriented databases should be used in applications involving complex object relationships, changing object structures or if the application defines members that are collections. Object oriented databases are being used in scientific research, telecommunication, computer aided drafting etc. But the downfall of object oriented databases is that it is tied to a specific programming language. Also it is difficult to scale once it exceeds it physical memory size. Object data stores should be avoided when data and relationships are simple. Db4o is a DBaaS provider using Object oriented databases.

### *2.5.1 db4o*

db4o was started by Carl Rosenberger in 2000 and the product was first shipped in 2001. In 2004 it was commercially launched as Db4objects Inc and was then acquired by Versant Corporation in 2008. db4o was developed using Java and C#. It provides a GUI called Object Manager Enterprise (OME) which can be used for various purposes like database connection, browsing databases, building queries and even administrative functions. It provides Native Queries (NQ) which allows the users to use common object oriented programming languages like Java, C# orVB.Net instead of query languages like SQL. It provides function that allows the user to store an object in a single command It also provides db4o Replication System which allows synchronizing relational backend with db4o. A major drawback of db40 is that it does not provide built in support to export or import data from JSON, XML or text files which is provided by other data stores. It does not provide features like referential integrity, OLAP tools offered by SQL. Some of the Fortune 500 companies that use db4o are BMW, Bosch, IBM, Intel and Seagate.

## 3. QUERY LANGUAGE

A query language can be defined as a computer language which can be used to manipulate the data inside a database. NOSQL does not use SQL (Structured Query Language) which is the most commonly used query language by relational databases as its query language. Also NOSQL does not have a standard query language. Most of the NOSQL database providers have created their own query language, for example Cassandra supports CQL (Cassandra query language), MongoDB uses mongo query language etc. Therefore it becomes difficult for the user to switch from one NOSQL database provider to another. Hence there is a need for a common query language like SQL which can be used for all NOSQL databases.

UnQL (pronounced as 'uncle') is one such collective effort to bring a familiar and standardized data definition and data manipulation language to the NOSQL platform. The acronym UnQl stands for Unstructured Query Language. UnQl is being developed by the creators of Couch and SQLite. UnQl is considered as the superset of SQL. It provides SQL like syntax thus providing familiarity to the developers. The concepts and syntax involved in UnQL is appropriate for the unstructured, self-describing data formats. It also provides features to allow for selection and manipulation of complex document structures. It provides the flexibility of the NOSQL schema-free design as well as the structured table format of the relational database. It can be used for querying data stored in JSON (JavaScript Object Notation) format as well as document databases and non-relational stores. It is open for users, vendors and the academic community for further development.

## 3. NOSQL DATABASES v/s RDBMS

NOSQL databases have both advantages as well as disadvantages over relational databases.

## 3.1 Advantages of NOSQL over Relational

- Provides a wide range of data models to choose from
- Easily scalable
- Database administrators are not required
- Some of the NOSQL DBaaS providers like Riak and Cassandra are programmed to handle hardware failures
- Faster , more efficient and flexible
- Has evolved at a very high pace

## 3.2 Disadvantages of NOSQL over Relational

- Immature
- No standard query language
- Some NOSQL databases are not ACID compliant
- No standard interface
- Maintenance is difficult

# 4. FUTURE PROSPECTS FOR NOSQL

Although NOSQL has evolved at a very high pace, it still lags behind relational database in terms of number of users. The main reason behind this is that the users are more familiar with SQL while NOSQL databases lack a standard query language. If a standard query language for NOSQL is introduced, it will surely be a game changer.

There are a few DBaaS providers over the cloud like Xeround which works on the hybrid database model, that is, they have the familiar SQL in the frontend and NOSQL in the backend. These databases night not be as fast as a pure NOSQL database but they still provide features of both relational as well as NOSQL databases to the user. Thus a lot of disadvantages of both relational as well as NOSQL databases may be covered up. With a few more advancements in this hybrid architecture the future prospects for NOSQL databases in DBaaS are excellent.

# 5. CONCLUSION

This paper describes the pros and cons of NOSQL databases. This paper also describes the advantages and disadvantages of each of the data stores and cases when a particular data stored can be used. Users must first consider various parameters like query language, the interface, availability, redundancy, consistency and analyze the pros and cons of various data models before choosing a particular data model.

# 6. REFERENCES

[1] 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

[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] db4o , http://www.db4o.com/about/

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

[7] Pramod J. Sadalage *and* Martin Fowler, **“**NoSQL Distilled**”**

[8] Riak , http://basho.com/technology/why-use-riak/

[9] 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

[10] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber,” Bigtable: A Distributed Storage System for Structured Data”, Google Inc

[11] UnQL , http://www.unqlspec.org/display/UnQL/Home

[12] Apache CouchDB , http://wiki.apache.org/couchdb/

[13] Amazon DynamoDB, http://aws.amazon.com/dynamodb/

[14] Neo4j , http://www.neo4j.org/learn