International Journal of Recent Trends in Engineering & Research
ISSN : 2455-1457 Impact Factor : 3.344

A STUDY OF NOSQL DATABASES AND WORKING OVERVIEWS

Asst. Prof. Raju Sharma[1], Asst. Prof. Yatendra kashyap[2]
[1,2] Department of Computer Science & Engineering Corporate
Institute of Science & Technology, Bhopal

Abstract: In this digital era, volume of data is growing very fast and become more complex in nature (structured and un-structured and hybrid). Volume of complex data sets generally refered as 'Big Data'. To control and retrieve and analyze Big data only RDBMS is not sufficient and it looks traditional or legacy for Big Data. To handle the problem, traditional RDBMS are complemented by specifically designed a rich set of alternative DBMS; such as - NoSQL, NewSQL. In this paper, we focus on different NOSQL databases, characteristics and their working in Big Data Analytics. This paper is intended to help users, especially to the organizations to obtain an independent understanding of the strengths and weaknesses of various NoSQL database approaches to supporting applications that process huge volumes of data for their projects and applications.
Keywords: Structured, un-structured, RDBMS, Big Data, NoSQL Database, NewSQL Database, Big Data Analytics.

I. INTRODUCTION

NOSQL is also known as Not Only SQL,‖ refers to an eclectic and increasingly familiar group of non-relational data management systems; where databases are not built primarily on tables, and generally do not use SQL for data manipulation. NoSQL database management systems are useful when working with a huge quantity of data when the data's nature does not require a relational model.

NoSQL systems are distributed, non-relational databases designed for large-scale data storage and for massively-parallel data processing across a large number of commodity servers. They also use non-SQL languages and mechanisms to interact with data (though some new feature APIs that convert SQL queries to the system's native query language or tool). NoSQL database systems arose alongside major Internet companies, such as Google, Amazon, and Face book; which had challenges in dealing with huge quantities of data with conventional RDBMS solutions could not cope. They can support multiple activities, including exploratory and predictive analytics, ETL-style data transformation, and non mission-critical OLTP (for example, managing long-duration or inter-organization transactions). Originally motivated by Web 2.0 applications, these systems are designed to scale to thousands or millions of users doing updates as well as reads, in contrast to traditional DBMSs and data warehouses [18].

NewSQL systems are relational databases designed to provide ACID (Atomicity, Consistency, Isolation, Durability) -compliant, real-time OLTP (Online Transaction Processing) and conventional SQL-based OLAP in Big Data environments. These systems break through conventional RDBMS performance limits by employing NoSQL-style features such as column-oriented data storage [1].

II. LITERATURE SURVEY IN DETAIL

There are many reasons given in the research of last many years on that basis we can opt any NOSQL approach for the project and application. Some of them is listed below as literature survey. They are:

i. In this paper, we examine a number of SQL and so called "NoSQL" data stores designed to scale simple OLTP-style application loads over many servers. Originally motivated by Web 2.0 applications, these systems are designed to scale to thousands or millions of users doing updates as well as reads, in contrast to traditional DBMSs [1].

ii. I think that as the dust settles it will become evident that NoSQL solutions will work alongside SQL solutions, each doing what they do best. Face book, Twitter, and many other companies are integrating NoSQL databases into their infrastructure right alongside SQL databases. Each has its strengths and weaknesses; neither will entirely displace the other. Some future SQL databases may start to take on features only found in NoSQL, such as elasticity and an ability to scale out to large amounts of commodity hardware [2].

iii. The main aim of this paper is to give an overview of NoSQL databases, about how it has declined the dominance of SQL, with its background and characteristics. It also describes its fundamentals that form the base of the NoSQL databases like ACID, BASE and CAP theorem. ACID property is not used in the NoSQL databases because of data consistency so we get to know how SQL lags data consistency. Later, on the basis of the CAP theorem we described different types of NoSQL databases that are Key-Value databases, Document Store Databases, Columnar based databases and Graph databases with the help of an examples [3].

iv. This paper motivation is to provide - classification, characteristics and evaluation of NoSQL databases in Big Data Analytics. This report is intended to help users, especially to the organizations to obtain an independent understanding of the strengths and weaknesses of various NoSQL database approaches to supporting applications that process huge volumes of data [4].

v. 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 [5].

vi. We focus on so called NoSQL databases which support solving, at least partially, Big Data problems. Some features of NoSQL databases like data models and querying capabilities are presented in more detail. We will also mention an overview of some their representatives. Finally, we point out on actual problems associated with current database research at all [6].

vii. Data aggregation becomes impossible on very large volumes of data when it comes to memory and time consumption. NoSQL databases provide an efficient framework to aggregate large volumes of data. There are different NoSQL databases like Key-value stores, Column Family/Big Table clones, Document databases and Graph databases. This paper compares different NoSQL databases against persistence, Replication, Transactions and Implementation language. This paper also discusses on performance and scalability aspects of different NoSQL databases [7].

viii. The massive growth of technology has greatly stimulated the need to generate data. Every day people and companies generate enormous amounts of data and this data may be structured, unstructured, semi-structured or a combination of all. This has called for the need to design databases which can store this type and volume of data and NoSQL databases have been the antidote crafted and designed to alleviate this problem. However, the NoSQL solution to this Big Data problem has also lead to several other problems therefore this paper seeks to reveal this problem taking a walkthrough into the structure of NoSQL databases. In addition, the paper also

makes a sneak preview of the discovered mainstream (MongoDB vs. Cassandra) NoSQL database security features analysis [8].

ix. In this paper, we evaluate five most popular non-relational databases from three types: Cassandra and HBase from Column Family databases, MongoDB and OrientDB from Document Store databases, and Redis from Key-value Store database [9].

x. In this paper shows that what can go wrong when scale a relational system with traditional techniques like sharding. The problems faced went beyond scaling as the system became more complex to manage. The benefits of data systems built using the Lambda Architecture go beyond just scaling as well as how much more robust your applications made. In this Paper also explore, that there are many other reasons why Big Data applications will be more robust. Although the Lambda Architecture as a whole is generic and flexible which solve the problem of computing arbitrary functions on arbitrary data in real-time by decomposing the problem into several layers [10].

xi. We reviewed the concepts of the relational databases and NoSQL database, motivation behind NoSQL databases and why many of big companies using them. NoSQL databases different in many aspects from traditional databases like structured schema, transaction methodology, complexity, crash recovery and dealing with storing big data which the feature lead to use NoSQL in cloud computing and may be data warehouses. also paper focused in Security because it became most undertaken feature today, in relational databases these feature covered very well however NoSQL has shortage in security mainly because their designer focuses on other purposes than security and generally the NoSQL databases solution still fresh it didn't reach the full maturity yet, for all that we can find many security vulnerabilities in it [11].

xii. Only a few years ago the Scalability and Performance were not such a big problem but the huge amount of data that is collected today is infinitely much more than ten years ago and also the growth of cloud computing results in large data store even more. This paper includes the introduction, causes of migrating towards NoSQL databases, characteristics, classification of NoSQL databases. Finally the security issues in NoSQL Databases are described and the security enforcement mechanism is proposed [12].

xiii. In this study, when a load occurs in an operating system that runs on RDBMS, some functions switch to NoSQL and the system utilizing NoSQL and stores 850% higher in post storage, 20% higher in inquiry list, and provides higher performance by about 58% higher in inquiry list posts, as compared to the system only using RDBMS. Therefore, for the systems requiring performance rather than consistency, applying NoSQL could bring about significant performance improvements [13].

xiv. In this article we described the main characteristics and types of NoSQL technology while approaching different aspects that highly contribute to the use of those systems. We also presented the state of the art of non-relational technology by describing some of the most relevant studies and performance tests and their conclusions, after surveying a vast number of publications since NoSQL's birth [14].

xv. This paper attempts to use NoSQL database to replace the relational database. It mainly focuses on one of the new technology of NoSQL database i.e. MongoDB, and makes a comparison study with MySQL and thus justifies why MongoDB is liked over MySQL. Lastly, a method is suggested to integrate with different-2 technology of these two types of database by adding a middleware (Metadata) between application layer and database layer. But in this paper we are propose with open source language like PHP [15].

xvi. If it is required to use medium data without complex queries and normal day to day functioning, then MySQL is a better but if the data is non-relational and may involve complex queries and joins if used in SQL, then MongoDB gives better performance for basic CRUD operations [16].

xvii. The paper pointed out that there is still a lack of geo-functionalities within document-oriented NoSQL-databases. The currently implemented geo-functions support only very basic operations. Relational databases are still far superior if the user needs to calculate geoinformation on database level [17].

III. CHARACTERISTICS OF NOSQL DATABASESCHARACTERISTICS OF NOSQL DATABASES

There are some Axiomatic of NoSQL which are given below:

a. ACID free

ACID stands for Atomicity, Consistency, Isolation and Durability. ACID concept basically comes from the SQL environment. But in NoSQL we will not use the ACID concept because of Consistency feature of SQL. As in the distributed environment, data is spread to different machines, each machine stores its data and maintenance of consistency is needed. For example, if there is change in one tupple of the table then changes are needed in each and every machine on which that particular data resides. If information regarding an updation spreads immediately, then consistency is given; if not, then inconsistency is carried out [3].

b. BASE

BASE stands for Basically, Available, Soft state, and Eventual consistency. BASE is reverse of ACID. NoSQL databases are divided in between the road from ACID to BASE. After a transaction consistency the state that we will get is soft state not a solid state. The main focus leading behind the BASE is the permanent availability. For example, thinking about the databases in banks, if two persons are accessing the same account in different cities then data updations is needed not just in time but needs some real time databases as well. Those updations need to be done frequently on all machines. Some more examples are online railway reservation, online book trade, etc. [3].

c. CAP

CAP stands for Consistency, Availability and Partition tolerance. CAP is basically a theorem that follows three principles:

i. The data available on all machines should be same in all respects and updations to be made on all machines frequently i.e. consistent data.
ii. Data must be available permanently and should be accessible each and every time i.e. availability.

iii. During machine failure or any faults in the machines database going to work fine without stopping their work i.e. partition tolerance.
In order to guarantee the integrity of data, most of the classical database systems are based on transactions. This ensures consistency of data in all situations of data management. These transactional characteristics are also known as ACID (Atomicity, Consistency, Isolation, and Durability). However, scaling out of ACID-compliant systems has shown to be a problem. Conflicts are arising between the different aspects of high availability in distributed systems that are not fully solvable - known as the CAP- theorem [3].

Many of the NOSQL databases above all have loosened up the requirements on Consistency in order to achieve better Availability and Partitioning. This resulted in systems know as BASE (Basically Available, Soft-state, eventually consistent) [21]. These have no transactions in the classical sense and introduce constraints on the data model to enable better partition schemes. Han, J., Haihong, E., Le, G., & Du, J. (2011) classifies NoSQL databases according to the CAP theorem [19]. Tudorica, B. G., & Bucur, C. (2011), compares using multiple criteria between several NoSQL databases [20]. Primary Uses of NoSQL Database (1) Large-scale data processing (parallel processing over distributed systems); (2) Embedded IR (basic machine-to-machine information look-up & retrieval); (3) Exploratory analytics on semi-structured data (expert level); (4) Large volume data storage (unstructured, semi-structured, small-packet structured). Accordingly, they provide relatively inexpensive, highly scalable storage for high-volume, small-packet historical data like logs, call-data records, meter readings, and ticker snapshots (i.e., ―big bit bucket‖ storage), and for unwieldy semi-structured or unstructured data (email archives, xml files, documents, etc.). Their distributed framework also makes them ideal for massive batch data processing (aggregating, filtering, sorting, algorithmic crunching (statistical or programmatic), etc.). They are good as well for machine-to-machine data retrieval and exchange, and for processing high-volume transactions, as long as ACID constraints can be relaxed, or at least enforced at the application level rather than within the DMS. Finally, these systems are very good exploratory analytics against semi-structured or hybrid data, though to tease out intelligence, the researcher usually must be a skilled statistician working in tandem with a skilled programmer.

IV. GERNERAL CATEGORIES CLASSIFICATION OF NOSQL DATABASES

There are some examples are given to show different categories of NoSQL databases they are: Leavitt, N. (2010), classifies NoSQL databases in three types: Key-value stores – e.g. SimpleDB [22]; column-oriented databases - e.g. Cassandra [8][9] [23], HBase[9] [24], Big Table [7] [25]; and document-based stores - e.g. CouchDB [22], MongoDB [23]. In this section, we classify NoSQL Databases in four basic categories, each suited to different kinds of tasks [3] [7] [9] –

i. Key-Value stores;
ii. Document databases (or stores);
iii. Wide-Column (or Column-Family) stores;
iv. Graph databases.

4.1 Key-Value stores

Typically, these DMS store items as alpha-numeric identifiers (keys) and associated values in simple, standalone tables (referred to as ―hash tables). The values may be simple text strings or more complex lists and sets [3]. Data searches can usually only be performed against keys, not values, and are limited to exact matches. See figure: 1.

Car	
Key	Attributes
1	Make: Nissan Model: Pathfinder Color: Green Year: 2003
2	Make: Nissan Model: Pathfinder Color: Blue Color: Green Year:2005 Transmission: Auto

Figure1: Key/Value Store NoSQL Database (Source: www.readwritewebcomimages.com)

Primary Use the simplicity of Key-Value Stores makes them ideally suited to lightning-fast, highly-scalable retrieval of the values needed for application tasks like managing user profiles or sessions or retrieving product names [9].

4.2 Document databases

This type of database is designed to manage and store documents. These documents are encoded in a standard data exchange format such as XML, JSON (JavaScript Option Notation) or BSON (Binary JSON) [22]. Unlike the simple key-value stores described above, the value column in document databases contains semi-structured data – specifically attribute name/value pairs.

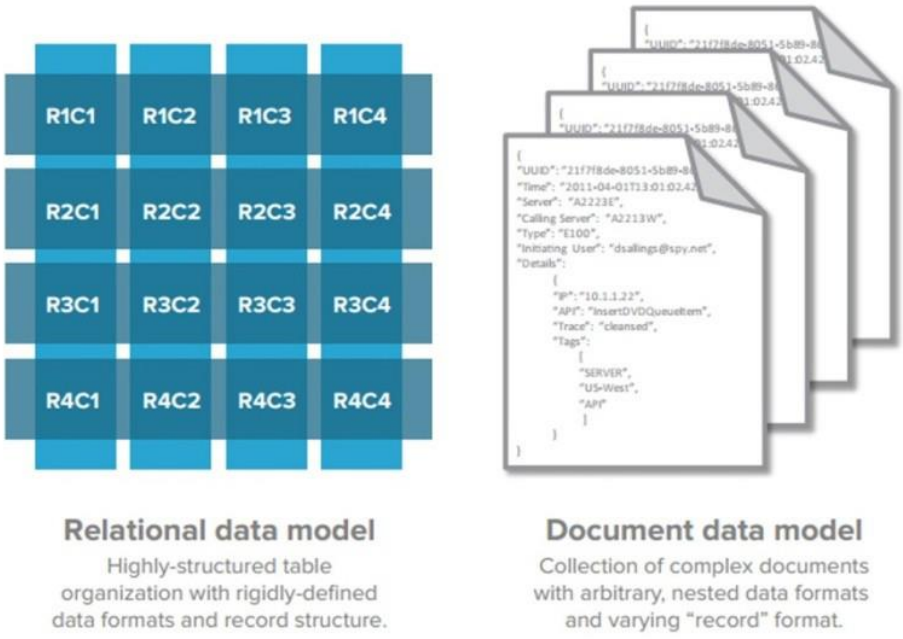

Figure2: Document Store NoSQL Database
(Source: http://gigaom.com/2011/07/29/couchbase-2-0-unql-sql-nosql/)

Primary Use: Document databases are good for storing and managing Big Data-size collections of literal documents, like text documents, email messages, and XML documents, as well as conceptual documents like de-normalized (aggregate) representations of a database entity such as a product or customer [22].

4.3 Wide-Column (or Column-Family) Stores (Big Table-implementations)

Like document databases, Wide-Column (or Column-Family) stores (hereafter WC/CF) employ a distributed, column-oriented data structure that accommodates multiple attributes per key. While some WC/CF stores have a Key-Value DNA (e.g., the Dynamo-inspired Cassandra)[8], most are patterned after Google's Big table, the petabyte-scale internal distributed data storage system Google developed for its search index and other collections like Google Earth and Google Finance. These generally replicate not just Google's Big table data storage structure, but Google's distributed file system (GFS) and Map Reduce parallel processing framework as well, as is the case with Hadoop, which comprises the Hadoop File System (HDFS, based on GFS) + Hbase (a Bigtable-style storage system) + Map Reduce [7].

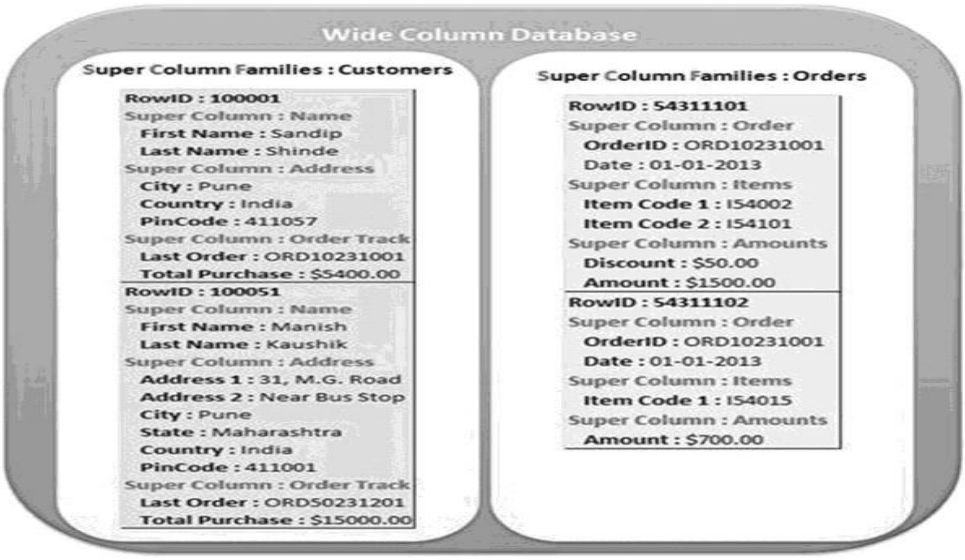

Figure3: Wide-Column Store NoSQL Database
(Source: http://bi-bigdata.com/2013/01/13/what-is-wide-column-stores/)

4.4 Graph Databases

They are similar to object-oriented databases as the graphs are represented as an object-oriented network of nodes (conceptual objects), node relationships (—edges) and properties (object attributes expressed as key-value pairs). They are the only of the four NoSQL types discussed here that concern themselves with relations, and their focus on visual representation of information makes them more human-friendly than other NoSQL DMS. NoSQL, in its incarnation at least, is a relatively new technology. However, it has already attracted a significant amount of attention due to its use by massive websites like Amazon, Yahoo, Facebook, which have data utilization rates that bring relational databases to a crawl.

V. CONCLUSION

We can conclude here is that the Computational and storage requirements of applications such as for Big Data Analytics, Business Intelligence and social networking over huge volume of datasets, there is a need of NOSQL database approaches because SQL alone is not sufficient to manage unstructured data sets. Due to the development of horizontally scalability as well as vertical scalability on demand basis such as in cloud computing, best way to reduce the complexity of retrieving data from big data NoSQL approach is required.

NoSQL is a large and expanding field, for the purposes of this paper - characteristics (features and benefits of NoSQL databases); classification (categories four on their features); comparison and evaluation (with a matrix on basis of few attributes- design, integrity, indexing, distribution, system) of different types of NoSQL databases; and current state of adoption of NoSQL databases. This paper provides an independent understanding of the various NoSQL database approaches to supporting applications that process huge volumes of data; as well as to provide a global overview of this non-relational NoSQL databases.

REFERENCES

i. Rick Cattell, "Scalable SQL and NoSQL Data Stores", SIGMOD Record, December 2010 (Vol. 39, No. 4).
ii. Greg Burd, "NoSQL", login: VOL. 36, NO. 5, OCTOBER 2011.
iii. Vatika Sharma, Meenu Dave, "SQL and NoSQL Databases", International Journal of Advenced Research in Computer Science and Software Engineering 2 (8), August- 2012, pp. 20-27.
iv. A B M Moniruzzaman and Syed Akhter Hossain, "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison", International Journal of Database Theory and Application Vol. 6, No. 4. 2013.
v. Ameya Nayak, Anil Poriya and Dikshay Poojary "Type of NOSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013.
vi. Jaroslav Pokorný, "New Database Architectures: Steps towards big data processing", IADIS European Conference Data Mining 2013.
vii. Clarence J. M. Tauro, Baswanth Rao Patil and K. R. Prashanth, "A Comparative Analysis of Different NoSQL Databases on Data Model, Query Model and Replication Model", Elsevier Publications 2013.
viii. Kudakwashe Zvarevashe, Tatenda Trust Gotora, "A Random Walk through the Dark Side of NoSQL Databases in Big Data Analytics", IJSR,Volume 3 Issue 6, June 2014.
ix. Veronika Abramova , Jorge Bernardino and Pedro Furtado , "Which NoSQL Database? A Performance Overview", Open Journal of Databases (OJDB), Volume 1, Issue 2, 2014.
x. Rakesh Kumar, Neha Gupta, Shilpi Charu and Sunil Jangir, "Manage Big Data Through NewSQL", National Conference on Innovation in Wireless Communication and Networking Technology - 2014.
xi. Mohamed Ahmed Mohamed, "Relational Vs. NOSQL databases: A Survey", International Journal of Computer and Information Technology (ISSN: 2279 – 0764)Volume 03 – Issue 03, May 2014.
xii. Asadulla Khan Zaki, "NoSQL Databases: New Millennium Database for Big Data, Big Users, Cloud Computing and Its Security Challenges", Volume: 03 Special Issue: 03, May-2014, NCRIET-2014.
xiii. Yong-Lak Choi, Woo-Seong Jeon , and Seok-Hwan Yoon, "Improving Database System Performance by Applying NoSQL", J Inf Process Syst, Vol.10, No.3, pp.355~364, September 2014.
xiv. Joao Ricardo Lourenço , Bruno Cabral, Paulo Carreiro, Marco Vieira and Jorge Bernardino, "Choosing the right NoSQL database for the job: a quality attribute evaluation", Springer, 2015.
xv. Lokesh Kumar , Dr. Shalini Rajawat and ,Krati Joshi, "Comparative analysis of NoSQL (MongoDB) with MySQL Database", International Journal of Modern Trends in Engineering and Research (IJMTER) Volume 02, Issue 05, [May– 2015].
xvi. Rajat Aghi, Sumeet Mehta, Rahul Chauhan, Siddhant Chaudhary and Navdeep Bohra, "A Comprehensive Comparison of SQL and MongoDB databases", International Journal of Scientific and Research Publications, Volume 5, Issue 2, February 2015.
xvii. Stephan Schmid, Eszter Galicz and Wolfgang Reinhardt, "Performance investigation of selected SQL and NoSQL databases", AGILE 2015 – Lisbon, June 9-12, 2015.
xviii. R. Cattell, (2010) "Scalable SQL and NoSQL Data Stores," ACM SIGMODRecord, vol. 39.
xix. Han, J., Haihong, E., Le, G., & Du, J. (2011, October). Survey on nosql database. In *Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on* (pp. 363-366). IEEE.
xx. Tudorica, B. G., & Bucur, C. (2011, June). A comparison between several NoSQL databases with comments and notes. In *Roedunet International Conference (RoEduNet), 2011 10th* (pp. 1-5). IEEE.
xxi. Graph Databases, NOSQL and Neo4j from: http://www.infoq.com/articles/graph-nosql-neo4j.
xxii. Find SimpleDB from: http://aws.amazon.com/simpledb/
xxiii. Find Cassandra from: http://cassandra.apache.org/
xxiv. HBase Databases from web: http://hbase.apache.org/
xxv. Chang, Fay, et al. "Big table: A distributed storage system for structured data."*ACM* Transactions on Computer Systems (TOCS) 26.2 (2008): 4.

