International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 11 143 – 148

A Survey of Non -Relational Databases with Big Data

Bansari H. Kotecha
Computer Engineering
L.D. College of Engineering
Ahmedabad
bansari1002@gmail.com

Prof. Hetal Joshiyara
Computer Engineering
L.D. College of Engineering
Ahmedabad
hetaljoshiyara@gmail.com

Abstract—The paper's objective is to provide classification, characteristics and evaluation of available non relational database systems which may be used in Big Data Predictions and Analytics .Paper describes why Relational Database Bases Management Systems such as IBM's, DB2, Oracle, and SAP fail to meet the Big Data Analytical and Prediction Requirements. The paper also compares the structured, semi-structured, and unstructured data. The paper also includes the various types of NoSQL databases and their specifications Finally, the operational issues such as scale, performance and availability of data by utilizing these database systems will be compared.

Keywords-*Non-Relational databases, Relatioanl databases,Bigdata*

*****

I. INTRODUCTION

Relational databases ruled the Information Technology (IT) industry for almost 40 years. But last few years have seen changes in the way IT is being used and viewed. Stand alone applications have been replaced with web-based applications, dedicated servers with multiple distributed servers and dedicated storage with network storage. Cloud computing has become a reality due to its lesser cost, scalability and pay-as-you-go model.It is one of the biggest changes in IT after the rise of World Wide Web. Cloud databases such as Big Table, Sherpa and SimpleDB are becoming popular. They address the limitations of existing relational databases related to scalability, ease of use and dynamic provisioning. Cloud databases are mainly used for data intensive applications such as data warehousing, data mining and business intelligence. These applications are read-intensive, scalable and elastic in nature. Transactional data management applications such as banking, airline reservation, online ecommerce and supply chain management applications are write intensive. Databases supporting such applications require ACID (Atomicity, Consistency, Isolation and Durability) properties, but these databases are difficult to deploy in the cloud. [17]. Relational databases exhibit a variety of limitations in meeting the recent Big Data analytics requirement in businesses. While clusters-based architecture has emerged as a solution for large databases, SQL is not designed to suit clusters and this mismatch has led to think of alternate solutions. There are mismatches between persistent data model and in-memory data structures, and servers based on SQL standards are now prone to memory footprint, security risks and performance issues.[24]

A **NoSQL** (originally referring to "non SQL" or "non relational" database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Motivations for 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 by NoSQL databases (e.g. key-value, graph, or document) differ slightly from those used by default in relational databases, making some operations faster in NoSQL and others faster in relational databases. The particular suitability of a given NoSQL database depends on the problem it must solve. Sometimes the data structures used by NoSQL databases are also viewed as "more flexible" than relational database tables. NoSQL databases are increasingly 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.[23]

II. Relational Database Management Systems (RDBMDS)

RDBMS is the standard language for relational database management systems. Data is stored in the form of rows and columns where each table must have one primary key. A **relational database**—or, an SQL database, named for the language it's written in, Structured Query Language (SQL)—is the more rigid, structured way of storing data, like a phone book. Developed by IBM in the 1970s, a relational database consists of two or more tables with columns and rows. Each row represents an entry, and each column sorts a very specific type of information, like a name, address, and phone number. The relationship between

tables and field types is called a **schema**. In a relational database, the schema must be clearly defined before any information can be added.[25].The relations between tables are also stored in the form of the table SQL (Structured Query Language) is a programming language used to perform tasks such as update data on a database, or to retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, DB2, Sybase, Microsoft SQL Server, Access, etc." [7, 8].

- *Limitations for SQL Database*:

1. 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.

2. Complexity: In SQL server's, data has to fit into tables anyhow. If the data doesn't fit into tables, then there is a need to design database structure that will be complex and again difficult to handle." [7, 8].

III. Not Only SQL (NoSQL)

A NoSQL provides mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.[27].NoSQL deals with unstructured schema so, less data can be stored in multiple collections and nodes and it does not require fixed table sachems; in addition it supports limited join queries and it is scaled it horizontally[5, 7, 9].

- *Benefits of NoSQL:*

1. Highly and easily scalable :Relational database or RDBMS databases are vertically scalable. When load increases on RDBMS database, we scale database by increasing server hardware power need to acquire expensive and bigger servers and NoSQL databases are designed to expand horizontally and in Horizontal scaling means that you scale by adding more machines into your pool of resources [18].

2. Maintaining NoSQL Servers is Less Expensive :The elasticity and scalability of cloud databases is what makes them less costly because the pricing model for cloud computing is pay-as-you-go.The complexity of a traditional RDBMS can be just as expensive to implement in the Cloud as they are on-premise because they do not scale out well.[29].

3. Redundancy:Redundant copies of data are important so if "the primary copy is destroyed, another copy is available for use."Redundant copies can be stored over a wide geographic area or within the same data center on different physical server racks.Distributed, redundant copies of data ensure high availability.[29]

4. Support for all datatypes:Cloud-based NoSQL databases "offer flexible and dynamic schema that accepts all key data formats" including structured, semi-structured, and unstructured.An RDBMS can only handle structured data.[29]

5. Easier manageability:Tools or sets of tools for carrying out "routine administrative operations" are provided by the vendor. Usually these tools are accessed via a web browser.[29]

6. Support Integrated Caching :NoSQL database supports caching in system memory, so it increases data output performance and SQL database where this has to be done using separate infrastructure." [5, 7, and 8].

- *Limitations of NoSQL:*

1. "NoSQL databases are open source which is its greatest strength but at the same time, it can be considered as its greatest weakness because there are not many defined standards for NoSQL databases; so, no two NoSQL databases are equal.

2. No stored procedures in Mongodb (NoSql database).

3. GUI mode tools to access the database is not flexibly available in market.

4. It is so difficult to find NoSQL experts because it is the latest technology and NoSQL developers are in learning mode [5, 6, 10, and 11].

- *Various types of Non-Relational database*
  - **Key-value model**—the least complex NoSQL option, which stores data in a schema-less way that consists of indexed keys and values. KVSs have become the most popular way to access Internetscale "cloud" storage systems.[1]Examples Cassandra, Azure, LevelDB, and Riak.[19,25]

 ISSN: 2321-8169
143 – 148

Table 1: Comparison of Riak, Redis and Voldemort

K-V stores / Properties	Riak	Redis	Voldemort
Language	Erlang	C, C++	Java
Fault tolerance	Replication	Replication	Data partition, Replication, RAID repair
Data model	Buckets, Keys-Values	Data structures	Structured blob/text
Community	Apache	BSD	Apache
Protocol	Http/REST or custom binary	Telnet-like, binary safe	Http
Data storage	Bitcask, LevelDB, Volatile memory, file system	Volatile memory, File system	TSconfig, LevelDB, HDFS, GridGain
Query language	Bhttp, Javascript, REST, Erlang	API calls	API calls
Map Reduce	YES	NO	NO
Replication mode	Multi master replication	Master Slave replication	Symmetric Replication
Operating system	Cross platform	Linux Mac OS Windows	Linux, Windows, MAC OS
Best for	High availability, Partition tolerance, Persistence	For rapidly changing data, Frequently written, rarely read statistical data	Application with large requirement on data capacity

- **Column store**—or, wide-column store, which stores data tables as columns rather than rows. It's more than just an inverted table—sectioning out columns allows for excellent scalability and high performance. Examples: HBase, BigTable, HyperTable.[19,25]

Table 2:Comparison of Hbase, Cassandra and Accumulo

Column stores / Properties	Hbase	Cassandra	Accumulo
Language	Java	Java	Java
Fault tolerance	Replication, Partitioning	Replication, Partitioning	Replication
Data model	BigTable	BigTable and Dynamo	BigTable
Community	Apache	Facebook	Apache
Protocol	Custom API, Thrift, Reset	Thrift	Thrift
Data storage	HDFS	Inspired by Amazon's Dynamo for storing data	HDFS
Query language	Apl calls, Reset XML, Thrift API	Apl calls, Thrift API	Java API, Thrift API, REST calls
Map Reduce	YES	YES	YES
Replication mode	Maser-Slave Replication	Master-Slave replication	Multi-master replication
Best for	Real-time access, Do bulk operation (indexing, ...)	When you write more than you read (logging), Must use Java	Access on the cell level

Document database—taking the key-value concept and adding more complexity, each document in this type of database has its own data, and its own unique key, which is used to retrieve it. It's a great option for storing, retrieving and managing data that's document-oriented but still somewhat structured. Examples: MongoDB, CouchDB.[19,25]

Table 3: Comparison of MongoDB and CouchDB

DB / Properties	MongoDB	CouchDB
Language	C++	Erlang
Fault tolerance	Replication	Replication
Data model	Document oriented (BSON)	Document oriented (JSON)
Community	AGPL and others	Apache
Protocol	TCP/IP	HTTP/REST
Data storage	Volatile memory, file system	Volatile memory, file system
MapReduce	YES	YES
Replication mode	Maste-Slave replication	Multi-master replication
Best for	Dynamic queries, Defining indexes, Good performance on a big DB.	Accumulating, Occasionally changing data, Pre-defined queries to be run, Web use cases and mobile applications.

- **Graph database**—have data that's interconnected and best represented as a graph. This method is capable of lots of complexity. Examples: Polyglot, Neo4J.[25]

Table 4: Comparison of Non-Relational database

TYPES	PERFORMANCE	SCALABILITY	FLEXIBILITY	COMPLEXITY
KEY-VALUE STORE	high	high	high	none
COLUMN STORE	high	high	moderate	low
DOCUMENT	high	variable (high)	high	low
GRAPH DATABASE	variable	variable	high	high

IV. Comparison of Structured and Unstructured data

Structured data sets are those where the activity of processing and output is predetermined and highly organized. Structured systems are designed. Payroll, Inventory control systems, point of sale systems, airline reservations are all forms of structured systems since they are using structured data- the data which is stored and displayed as a set of rows and tables. In contrast, unstructured data sets are the data that have little or no predetermined form or structure. Unstructured data sets include email, contracts, blogs, and other communications. A person who performs a communications activity in an unstructured system has wide latitude to structure the message in whatever form is desired. The rules of unstructured systems are fewer and less complex [9, 10, and 13]. The structured and nanostructured data can be different from technical, organizational, structural, functional point of view. Each has its own environment and needs to be treated and used accordingly. The structured and nanostructured data can be different from technical, organizational, structural and functional point of view. [4, 11, 15, 18].

Relational databases are highly structured: all the data in the table are stored as rows and columns. Each column has a data type which is mostly normalized. The SQL is suitable

to relational databases to store and retrieve data in a structured way. Queries are Plain English commands. There are always fixed number of columns although additional columns can be added later. Most of the tables are related to each other with primary and foreign keys thus providing "Referential Integrity" among the objects. The major vendors are ORACLE, SQL Server, MySQL, PostgreSQL, etc. [7, 9].

V. Bigdata Performance and scalability

In order to effectively and efficiency use the underlying databases of Big Data, we need to either scale up approach or scale out approach to deal with the concurrent global users, commonly referred as Big Users [3, 4].

Scale up approach refers to a centralized architecture in which functionalities are added to existing servers based on the increase number of global concurrent users and these servers becomes bigger and bigger [3,4]. While scaling-out refers to a distributed architecture where a commodity of servers are added to meet the requirements of global users [3, 6, and 6]. The scale-out approach of NoSQL databases is very much easier. If huge number of users start using application then another commodity server is added very simply. There is no need to modify the application since the application always sees a single (distributed) database [16].

Along with performance, cost and scalability of NoSQL databases, the flexibility is also equally attractive. As users come and go, commodity servers/virtual machines can be quickly added or removed from the server pool by keeping track of the user population and thus operating cost is also reduced. The NoSQL databases are highly fault tolerant databases because the load is distributed across many commodity servers and thus support in continuous operations [10, 11, and 13].

A distributed scale-out approach also usually ends up being cheaper than the scale-up alternative. This is a consequence of large, complex, fault-tolerant servers being expensive to design, build and support. [7, 10, and 11].

When a user needs to run a query on a set of data, the desired information needs to be collected from many tables and joined before it can be provided to the application. Similarly, when writing data, the write needs to be coordinated and performed on many tables. When data is relatively low-volume and when it is flowing into a database at a low velocity, a relational database is usually able to capture and store the information. But, today's applications are often built on the expectation that massive volumes of data can be written (and read) at speeds near real-time [1, 2, and 3].

VI. CONCLUSION

In this paper the characteristics of Non Relational Database Systems and Big Data were presented. Also, the types of data each approach has to deal which were discussed. The paper also describes the comparison of variuos types of NoSQL databases. The operational issues such as scale, performance, and availability of data by utilizing these database systems were also compared. Non-Relational databases can be used to leverage the power of Bigdata. Bigdata can be handeled more specifically, precisely and easily with it.

REFERENCES

[1] "Big Data: Volume, Velocity, Variability, Variety", http://nosql.mypopescu.com/post/6361838342/bigdata-volume-velocity-variability-variety, Accessed April 2017.

[2] B. Wiederhold, "18 essential Hadoop tools for crunching big data", Network World, www.googletagmanager.com/ns.html. Accessed April, 2017.

[3] A. K. Zaki, "NoSQL Database: New Millennium Database for Big Data , Big Users, Coud Computing and Its Security Challenges," http://esatjournals.org/Volumes/IJRET/2014V03/I15/IJRET20140315080.pdf, Accessed May 2015.

[4] L. Arthur, "What is Big Data", htpp::://www.forbes.com/sites/liasaarthur/2013/08/15/what-is-big-data/, Accessed May 2107

[5] S. Penchikala, "Virtual Panel: Security Considerations in Accessing NoSQL Databases‖, Nov. 2011. http://www.infoq.com/articles/nosql-data-security-virtual-panel., Accessed May 2015.

[6] Oracle Databases from web: http://www.oracle.com/us/products/database/overview/index.h tml, Accessed May 2017.

[7] Relational Database Management System (RDBMS) vs noSQL. http://openproceedings.org/html/pages/2015_edbt.html, Accessed April 2017.

[8] "Relational -database-management-system-rdbms-vs-nosql/, http://www.loginradius.com/engineering/relational-database-management-system-rdbms-vs-nosql/ Accessed April 2017.

[9] M. Ramachandran "Relational Vs Non-Relational databases", http://bigdata-madesimple.com/relational-vs-non-relational-databases, Accessed May 2017.

[10] L. P. Issac "SQL vs NoSQL Database Differences Explained with few Example DB", http://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/, Accessed May 2017.

[11] Sherpa Software. "Structured and Unstructured Data: What is It? http://www.sherpasoftware.com/blog/structured-and-unstructured-data-what-is-it/, Accessed May 2017.

[12] F. Chang, et al. "Bigtable: A distributed storage system for structured data.", ACM Transactions on Computer Systems (TOCS) 26.2 (2008): 4.

[13] D. Gosain, "A survey and comparison of relational and non-Relational Databases‖. IJERT,Vol 1,Issue 6,2012.

[14] *L. Okman, , N. Gal-Oz, Y. Gonen, E. Gudes, and J. Abramov, , "Security Issues in NoSQL* Databases," Trust, Security and Privacy in Computing and Communications (TrustCom), 2

[15] IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi: 10.1109/TrustCom.2011.70

[16] Cristina Basescu, Christian Cachiny, Ittay Eyalz, Robert Haasy, Alessandro Sorniottiy, Marko Vukolicx, and Ido Zachevskyz," Robust Data Sharing with Key-Value Stores"

[17] Indu Arora1 and Dr. Anu Gupta "Cloud Databases: A Paradigm Shift in Databases" https://www.semanticscholar.org/paper/Cloud-Databases-A-Paradigm-Shift-in-Databases-Arora-Gupta in 2012

[18] Find Source: www.couchbase.com/why-nosql/nosql-database.

[19] Ahmed Oussous, Fatima-Zahra Benjelloun, Ayoub Ait Lahcen, SamirBelfkih,"Comparison and Classification of NoSQL Databases for Big Data"

[20] "The Four Pillars of Big Data for the CMO", http://www.certona.com/the-four-pillars-of-big-data-for-the-cmo/, Accessed August 2015.

[21] Unlocking the Magic of Unstructured Content", *IDS White Paper*, http://inmoncif.com/registration/whitepapers/unlocking-final.pdf, Accessed August 2015.

[22] Z.Liu, B. .Jiangz and J. Heer, "imMens: Real-time Visual Querying of Big Data" Eurographics Conference on Visualization (EuroVis) 2013 Volume 32 (2013), Number 3.

[23] Pankaj Sareen , Parveen Kumar, " NOSQL DATABASE AND ITS COMPARISON WITH SQL DATABASE"

[24] Sitalakshmi Venkatraman , Kiran Fahd, Samuel Kaspi, Ramanathan Venkatraman ," SQL Versus NoSQL Movement with Big Data Analytics "http://www.mecs-press.org/ijitcs/ijitcs-v8-n12/IJITCS-V8-N12-7.pdf in 2016

[25] Carey wodehouse," SQL vs. NoSQL Databases: What's the Difference?"https://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/, Accessed in November 2017

[26] C. White, "Using Big Data for Smarter Decision Making", BI Research, July 2011.

[27] James Serra ,"Relational database vs Non –Relational Databases " www. Jamesserra.com" in August 2015

[28] S. Grimes, "Unstructured Data and the 80 Percent Rule", http://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/, Accessed in November 2017.

[29] "Database Management in the Cloud Computing Era" http://www.itmanagerdaily.com/database-management/ Accessed in 2017