ISSN No. 0976-5697

Volume 8, No. 1, Jan-Feb 2017

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

A review on NoSQL: Applications and challenges

Abdul Haseeb
Research scholar (M.Tech)
Department of Computer Science and IT
Hyderabad, India

Geeta Pattun
Assistant Professor
Department of Computer Science and IT
Hyderabad, India

Abstract: Now a day the technology is growing rapidly stimulating and generating whopping amount of data. Every day people and companies generate huge amounts of data and this data may be unstructured, semi-structured and structured. That's why we need to design databases which can store this type of data in huge volumes. The name of this database is NoSQL databases. NoSQL database solves this type of problems. NoSQL database is being used widely and it is a commonly known as engines well scale. Therefore, it is useful to investigate how different factors, such as workload, data size and number of simultaneous sessions influence scaling capabilities. In this paper we describe the brief introduction of NoSQL and its categories and also what the benefits of NoSQL are and why we are using now.

Keywords: NoSQL, Graph DB, Key value DB, Column DB, Document DB

I. INTRODUCTION

Now a day's big data analysis are at the center of modern science and commercial. It is generated by users from online like as transactions, emails, videos, audios, images, click streams, logs, posts, search queries, health records, social networking interactions, science data, sensors and mobile phones and their applications. Which are stored in databases at the time of storing the data we have some problems like as how to capture, store, manage, share, analyze and visualize via typical database either structured or unstructured form of data [3].

The major issues of researchers and practitioners is that the growth rate of data exceeds their ability to design appropriate cloud computing platforms for data analysis and update problem of workloads [5]. Due to these problems we need to change the database because the traditional relational databases are proved to be weak for distributed environment. Now the new database released by some scientist for better performance better scalability and solving the major problems. Being schema free, elastic and scalable, NoSQL databases are being appeared [1].

NoSQL is used in 1998 by Mr. CarloStrozziand his lightweight open source relational database. NOSQL means "Not Only SQL" is also used for these databases. They provide storage and retrieval mechanism with less constrained consistency models than traditional relational databases. There are five categories of NoSQL databases like Key-Value Store, Document-Store, Column-Oriented, Graph Database, and data structure store Cassandra, MongoDB, BigTable, neo4j and Redis are the examples of these databases [8].

In this paper we study about what is NoSQL database, and why we use now. NoSQL database are highly scalable, provide better performance, designed to store and process a significant amount of all type of data at a speed 10 times faster than RDBMS, high availability and strong fail over storage capabilities [8]. But Security is a major concern for IT Enterprise Infrastructures. Security in NoSQL databases is very weak, Authentication and Encryption is almost nonexistence or is very weak when we were implemented [3].

II. NOSQL

NoSQL are using with a wide variety of different DB technologies that came into existence in response to the demands presented in building modern applications:

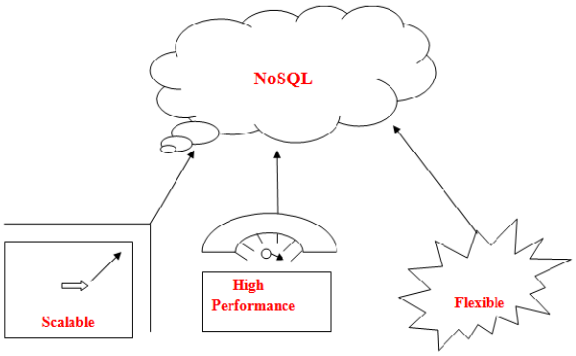

Figure 1.Benefit of NoSQL

NoSQL database solutions are mostly invited by the requirements of eCommerce, Web services, mobile computing and location based services, and social media applications. eCommerce application require predictive analytics, personalization, dynamic pricing, superior customer service, fraud and anomaly detection, real-time order status through supply chain visibility, and Web server access log analysis. All of the above functions are characterized by massive data which need to be fused from multiple sources and processed in real-time [1].

- Developers are working with applications that create massive volumes of new, rapidly changing data types — structured semi-structured, unstructured and polymorphic data.
- Long gone is the twelve-to-eighteen month waterfall development cycle. Now small teams work in agile sprints, iterating quickly and pushing code every week or two, some even multiple times every day.

- Applications that once served a finite audience are now delivered as services that must be always-on, accessible from many different devices and scaled globally to millions of users.
- Organizations are now turning to scale-out architectures using open source software, commodity servers and cloud computing instead of large monolithic servers and storage infrastructure.

Relational databases were not designed to cope with the scale and agility challenges that face modern applications, nor were they built to take advantage of the commodity storage and processing power available today.

A. Why NoSQL

There are various regions to move on NoSQL like as

- The growing of Big Data- In that High data velocity, Data variety, Data volume, and Data complexity.
- Data Availability always
- Real Location Transparency.
- New era Transactional Capabilities
- Flexible Data Architecture
- High Performance Architecture
- Highly Intelligence

III. NoSQL Database Types

There are various type of database which are representing through the hierarchy order in this figure.

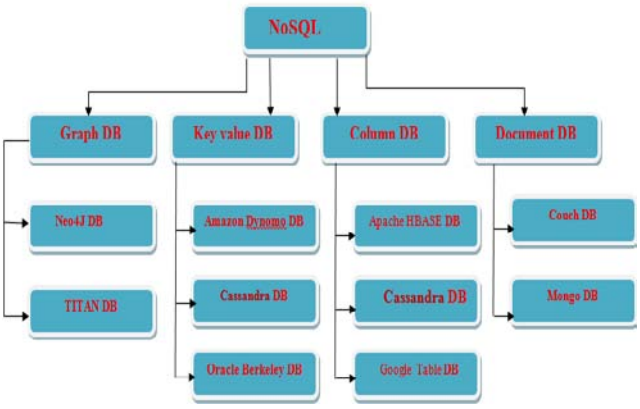

Figure 2: Types of Database

A. Graph Database Stores

Graph Database stores are utilized to store information about system of information, for example social connections. Graph stores include Titan DB and Neo4J.

1. **Neo4J**-Neo4j is an open source graph database. Neo4j does not support data approval, encryption and examining. The communication between client and server depends on SSL protocol [10].Nodes/vertices in the graph are depicted using properties and the connections between nodes are typed and relationships can have their own relative properties. Deployed on compute clusters in a solitary data center or over different geographically distributed data centers. Highly versatile and existing applications have 32 billion relationships, 32 billion nodes, and 64 billion properties. Client interfaces like Java, REST, Cipher (SQL-like) and Gremlin.
2. **Titan DB-**Titan is an OLTP distributed graph database fit for supporting a huge number of simultaneous clients connecting with a solitary gigantic scale diagram spoke to over a cluster of machines. The current distribution's pluggable storage architecture gives network to Apache HBase, Apache Cassandra, and Oracle's BerkeleyDB. Titan is a native Blueprints empowered graph database and in that capacity, it supports the full TinkerPop stack of advancements [15].

B. Key-value Database stores

Key-value Database stores are the most straight forward NoSQL databases. Each and every item in the database is put away as a attribute name (or "key"), together with its value. Cases of key-value stores are Cassandra, Amazon Dynamo DB and Berkeley DB.

1. **Amazon Dynamo DB**-is a quick and adaptable NoSQL database utilized in Amazon. It supports both document data model and key value. Data encryption is not bolstered in Dynamo but rather the communication amongst client and server utilizes https protocol. Authentication and approval is upheld by dynamo and requests should be marked utilizing HMAC-SHA256 [10].Amazon's Dynamo is different approach that offered an eventually consistent replication mechanism with tunable consistency levels. Dynamo's open-source variant [6].
2. **Cassandra DB**-Apache Cassandra is a NoSQL database that is classified as having a place with Column Family sort databases. Cassandra was originally created by Facebook and these days are under control of Apache Software. This database was outlined to handle extensive measure of distributed data and requests while providing no single purpose of disappointment and programmed data distribution over a cluster [11].
3. **Berkeley DB-**Oracle Berkeley DB is a superior embeddable database giving key value storage. Berkeley DB offers advanced highlights including value-based information stock piling, replication for high availability, highly concurrent access and adaptation to non-critical failure in an independent, small footprint software library [14].

C. Column Database stores

Cassandra and HBase are advanced for queries over vast datasets, and store columns of data together, rather than rows. Cassandra utilizes steady hashing to ensure a good circulation of key ranges (partition of data, or shards) to storage nodes.

1. **Cassandra DB-**functions admirably with applications that share its casual semantics (for example keeping up client information in online stores but is not solid match for more traditional applications requiring strong consistency. We now decided to embark on a re-design of Cassandra that preserves some of its contents (such as its partitioning of data based on predictable hashing) however replaces others with the point of farcifying consistency [6]. Cassandra utilizes the Gossip convention for coordination purposes in a distributed establishment [19]. Gives possible consistency Client interfaces and Pycassa (Python binding), phpcasa (a PHP wrapper), command line/shell, Thrift, and Cassandra Query Language (CQL). Well known for creating financial services applications. Cassandra is a distributed, eventual consistency, highly scalable and structured key-value store. Cassandra unites the distributed systems technologies from Dynamo and the data model show from Google's BigTable. Like Dynamo, Cassandra is eventually reliable. Like BigTable, Cassandra gives a column family-based data model richer than typical

key/value systems. It was publicly released by Facebook in 2008 [4].

2. **Apache HBase DB-** HBase is a distributed column-oriented database based on top of the Hadoop document frame work. It is an open-source extends and is on a level plane versatile. HBase is a data model that is like Google's huge table intended to give speedy arbitrary access to immense measures of organized information. It use the adaptation to non-critical failure gave by the Hadoop File System (HDFS). It is a part of the Hadoop biological system that gives random real-time read/compose access to data in the Hadoop File System. One can store the data in HDFS either straightforwardly or through HBase. Data consumer reads/accesses the data in HDFS randomly utilizing HBase. HBase sits on top of the Hadoop File System and provides read and compose access. HBase and OrientDB, showed the slow execution time during read operations. HBase presented the worst result, and it is 1.86 circumstances bring down contrasted with the Column Family database Cassandra. Given a substantial number of records, HBase demonstrated more trouble amid execution of peruses. In HBase, parts of a similar record might be put away in various plate documents and these outcomes in an expanded execution time. HBase is streamlined for the execution of upgrades, however for peruses, as we will see later on, HBbase demonstrates a decent execution over the workload H with 100% updates [12]. HBase depends on an ace slave structure and needs the database handle per hub itself, an appropriately arranged Hadoop Distributed File System (HDFS) and a Zookeeper framework with a specific end goal to oversee distinctive HBase forms [17].
3. **Google BigTable-**Open source communities like apache and many companies like Google and Amazon have been making use of this and they are contributing to its progress. Data replication is the technique which ensures availability and fault tolerance. The space of NoSQL key value data stores are infect defined by the Google's BigTable. Projects developed by Google are stored by BigTable. Google earth is one such example. Amazon has in the meantime developed its own system [12].

D. Document databases

Document data base represents a complex data structure commonly known as document. As such different key value pairs are present in a document. Key array pairs are nested Document data base is to ensure big data storage and efficient query processing. It is least interested in high performance of read and writes concurrent operations. The family of Document data base has following types.

1. **Mongo DB-**MongoDB was initially developed by 10gen. Its purpose is to manage, store JSON like document collections. This Document database ensures query able and indexable data despite being nested in complex hierarchies. Document are here stored in collections and these collections are further stored in databases [7]. For comprehension collection may be treated as a table in relational database management system, the only difference being database management system has a schema. In addition to the features of availability, the document data base is scalable with the use of shardirings and replica sets [3]. MongoDB can be viewed as middle between relational and non-relational databases and also MongoDB does not bolster joins thus it requires denormalization [13].

   Some of its features are

   - Like the relational database it features its chest.
   - It supports complex data types example the complex data stores of BJSON are stored in MongoDB.
   - It has a very powerful query language as it is indexible, queryable relational database.
   - Mass data can be accessed at very high speed. Data exceeding 50 GB can be accessed 10 times faster than ordinary MySQL. Because of this speed, may projects with increasing data are switching towards MongoDB.
2. **Couch DB-**Couch DB is another fault tolerant, flexible document database NoSQL database. It runs on Hadoop Distributed file systems (HDFS) and in an open source apache project. To secure the database authentication based on both password and cookies is used and it does not support data encryption. The password are encrypted using PBK DF2 hash algorithm and is sent over the network using SSL protocol. But couch DB is has potential for script injection and denial of security.

IV. BENEFITS OF NOSQL

NoSQL approach is a true standard in executing frameworks to deal with Big Data. Numerous organizations worldwide are at present applying high performance conveyed NoSQL arrangements. Administrations of such organizations as IBM, Amazon, Facebook, Twitter, Google, Oracle, and so on [18]. When we are comparing to RDBMS and NoSQL databases than we found NoSQL are more scalable and provide high performance, and their data model addresses several issues that the RDBMS is not designed to address. When client are giving the I/O (data) than access is slower, mapping database or its parts into primary memory increases the performance and reduces the execution time of querying [12]. Various types of NoSQL databases available for example graph storage, key-value store, document store and wide column store [16]. NoSQL database innovation offers advantages of versatility and accessibility through even scaling, replication, and improved information models, however particular usage must be picked right on time in the architecture configuration prepare [9].

The main features of NoSQL are as follows

- It has huge volumes of fastly changing structured, semi-structured, and unstructured data which is generated by users.
- It has also quick schema iteration, agile sprints and frequently code pushes quickly.
- It supports Object-oriented programming language and easy to understand and also user use easily and flexible with short period of time.
- NoSQL generally geographically distributed scale-out architecture where it is not expensive, and also monolithic architecture.

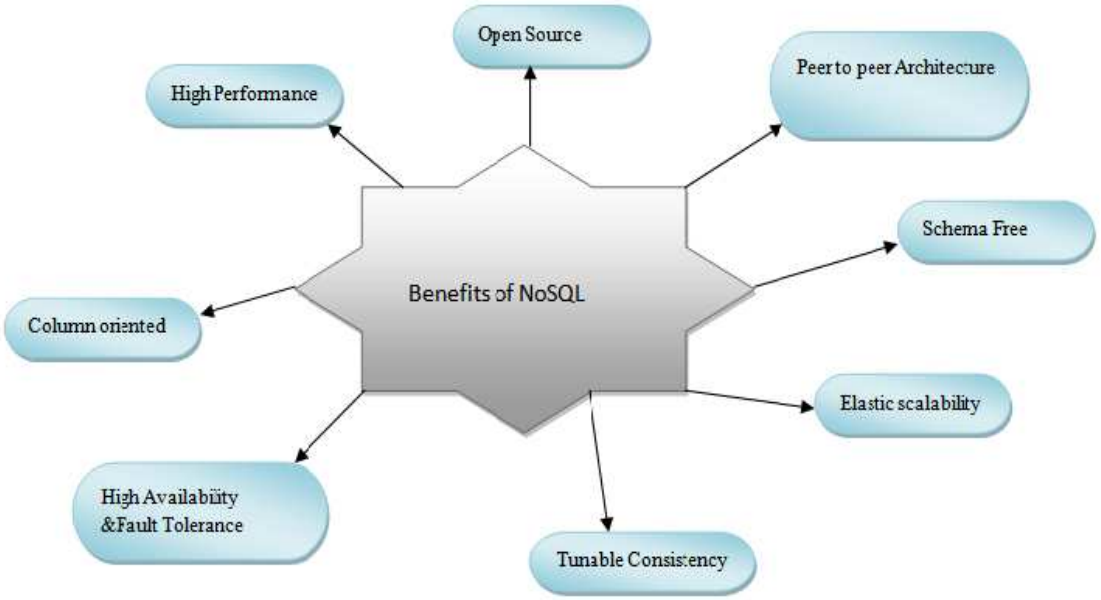

Figure 3: Benefits of NoSQL

A. **Comparison between NoSQL Databases**

Table 1: Comparison between NoSQL Databases

DB /Criteria	Authentic-ation	Authorization	Auditing	Communication protocol	Potential for attack	Data Encryption	Data Model	Scalability	Data size	Data complexity
Neo4J	-	Not Support	Not Support	SSL	-	Not Support	Graph	Low	Low	High
Dynamo DB	Support	-	-	http	-	Not Support	Key/Value Document	High	High	Low
Cassandra	Support	Not Support	Not Support	SSL	Script injection (in CQL) and DOS	Not Support	Key/Value	High	High	Low
Hbase	Support	Support	-	SSH	Not support for DOS and injection	Not Support	Column Oriented	High	High	Low
Google Bigtable	Not Support	-	-	-	-	Not Support	Column Oriented	High	High	Low
CouchDB	Support	-	-	SSL	Script injection and DOS	Not Support	Document	Middle	Middle	Middle
MongoDb	Not Support	Not Support	-	SSL	Script injection	Not Support	Document	Middle	Middle	Middle

V. NOSQL AND RELATIONAL DB

The NoSQL database maintains consistency models which are merely constrained in contrast to the relational database for collection and retrieval of data. NoSQL is optimized for use in various fields of engineering and traditional industries based on real time web applications and big data. The primary operation of NoSQL is to simplify retrieval and appending of huge data using data processing operations. Simplicity of design, horizontal scalability and finer control over availability are the key feature of NoSQL and with these technical features are showing efficient and reliable results. Fig-4 illustrates the advantage of NoSQL over the relational databases in big data utilization environment.

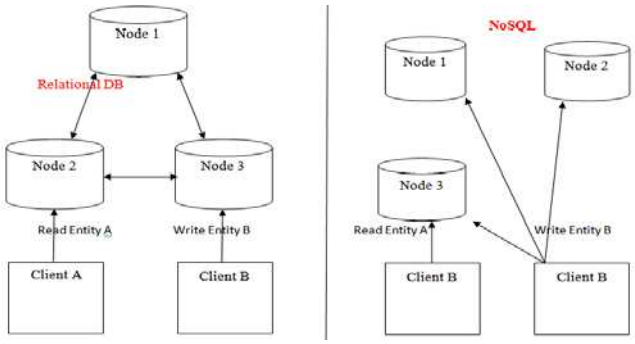

Figure 4: Relational DB Vs NQSQL DB

Relational databases load data horizontally (Fig-4) which is not the case in NoSQL, as it does not distributed data in this pattern. A relational database also does not distribute logical entities across multiple tables, as it in being stored at one place. Relational databases do not ensure referential integrity between logical entities. This feature enables them to naturally distribute

data across a huge number of database nodes and also write the independence. Consider a user write entity B to A database cluster with 3 nodes; it is certainly going to spread evenly across all of them. A distributed RDBMS having client 1 will not be able to see any changes until all the nodes are acknowledged with a two phase commit or it will get blocked till this happens. Synchronization in RDBMS also requires data to be read from other nodes for ensuring referential integrity. These things happen during the transaction this blocking client B. This illustrates the fact that such kind of solution can scale horizontally [2]. It also means that it can leverage its distributed nature for high availability. This has huge importance in the cloud, where there is tendency for every node to fail at any moment. The above discussion makes it clear that solution offered by NoSQL lakes some features that define on RDBMS solution due to reason of scalability. Numerous organizations worldwide are as of now applying high performance Conveyed NoSQL arrangements [18].

VI. CONCLUSION

Now in this modern era people are moving on SQL to NoSQL. In NoSQL have lots of features in the perspective of huge amount of storage management and their utilization. We will plan for enhancement the security issues for better use of recourses in future.

VII. REFERENCES

[1] Gudivada, Venkat N., Dhana Rao, and Vijay V. Raghavan. "NoSQL systems for big data management." Services (SERVICES), 2014 IEEE World Congress on. IEEE, 2014.

[2] Hwang, J. S., Lee, S., Lee, Y., & Park, S. (2015). A Selection Method of Database System in Big data Environment: A Case Study From Smart Education Service in Korea. International Journal Advance Soft Computing Application, 7(1), 9-21.

[3] Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., & Abramov, J. (2011, November). Security issues in nosql databases. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on (pp. 541-547). IEEE.

[4] Wang, G., & Tang, J. (2012, August). The nosql principles and basic application of cassandra model. In Computer Science & Service System (CSSS), 2012 International Conference on (pp. 1332-1335). IEEE.

[5] Swamy, R., & Srinivasa, K. (2013). Explicit Data Encryption Architecture for Cassandra. International Journal of Engineering Innovations and Research, 2(4), 376.

[6] Garefalakis, P., Papadopoulos, P., Manousakis, I., & Magoutis, K. (2013, June). Strengthening consistency in the cassandra distributed key-value store. In IFIP International Conference on Distributed Applications and Interoperable Systems (pp. 193-198). Springer Berlin Heidelberg.

[7] Zaki, A. K. (2014). NoSQL databases: new millennium database for big data, big users, cloud computing and its security challenges. International Journal of Research in Engineering and Technology (IJRET), 3(15), 403-409.

[8] Mohamed, M. A., Altrafi, O. G., & Ismail, M. O. (2014). Relational vs. nosql databases: A survey. International Journal of Computer and Information Technology, 3(03), 598-601.

[9] Klein, J., Gorton, I., Ernst, N., Donohoe, P., Pham, K., & Matser, C. (2015, February). Performance evaluation of nosql databases: A case study. In Proceedings of the 1st Workshop on Performance Analysis of Big Data Systems (pp. 5-10). ACM.

[10] Sahafizadeh, E., & Nematbakhsh, M. A. (2015). A Survey on Security Issues in Big Data and NoSQL. Advances in Computer Science: an International Journal, 4(4), 68-72.

[11] Abramova, V., Bernardino, J., & Furtado, P. (2014, June). Testing cloud benchmark scalability with cassandra. In Services (SERVICES), 2014 IEEE World Congress on (pp. 434-441). IEEE.

[12] Abramova, V., Bernardino, J., & Furtado, P. (2014). Which nosql database? a performance overview. Open Journal of Databases (OJDB), 1(2), 17-24.

[13] Kanade, A., Gopal, A., & Kanade, S. (2014, February). A study of normalization and embedding in MongoDB. In Advance Computing Conference (IACC), 2014 IEEE International (pp. 416-421). IEEE.

[14] Tauro, C. J., Aravindh, S., & Shreeharsha, A. B. (2012). Comparative study of the new generation, agile, scalable, high performance NOSQL databases. International Journal of Computer Applications, 48(20), 1-4.

[15] Available on (http://espeed.github.io/titandb/)

[16] Iqbal, N. & Islam, M. (2016).From Big Data to Big Hope: An outlook on recent trends and challenges. Journal of Applied Computing, 1(1):14-24.

[17] Waage, T., & Wiese, L. (2014, November). Benchmarking Encrypted Data Storage in HBase and Cassandra with YCSB. In International Symposium on Foundations and Practice of Security (pp. 311-325). Springer International Publishing.

[18] Kozlov, A. A., Aleshina, A. A., Kamenskikh, I. S., Rovnyagin, M. M., Sinelnikov, D. M., & Shulga, D. A. (2016, February). Increasing the functionality of the modern NoSQL-systems with GPGPU-technology. In NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), 2016 IEEE (pp. 242-246). IEEE.

[19] Waage, T., Jhajj, R. S., & Wiese, L. (2015, October). Searchable encryption in apache cassandra. In International Symposium on Foundations and Practice of Security (pp. 286-293). Springer International Publishing.