

FUNCTIONING CONCOMITANTLY WITH NOSQL DATABASE IS MORE CHALLENGING

Dr. Brijesh Khandelwal

Associate Professor, Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur

ABSTRACT-

NoSQL is known abbreviation of “Not Only SQL”. From the generations each web provision is using structured database and its access is guaranteed subject to SQL databases. The databases were recognized likewise the spine from claiming at whatever provision the place information might be retrieved utilizing SQL queries. Starting with the most recent couple of quite some time new vibe about NoSQL is hailing in for holding volumes about unstructured information. This will be a self-evident certainty that now a day’s volume for information may be accessible through IoT gadgets which may be altogether unstructured and is versant excessively awful. Handling such sort of information with available quality RDBMS software/ tools is not possible. Here is the place the NoSQL databases originated under spotlight.

RDBMS need aid fit about taking care of volumes about organized information. End users today run different tasks/queries on a specific data-set at the same time. To handle such queries ahead with un-structured and enormous volume for information NoSQL databases would be obliged.

The center about NoSQL databases may be reliability, scalability what's more consistency. The objective of this study is to articulate the variants of NoSQL for various databases and subsequently the architecture of NoSQL database. We will also study that leverage of NoSQL will ensure high performance than RDBMS by applying NoSQL to database systems that implement RDBMS

KEY WORDS-

NoSQL database, RDBMS, CAP theory, Document Database, Key-Value stores, Wide-Column Stores, Graph Stores

1. INTRODUCTION

Information will be produced through a large number wellsprings like benefits of the business courses transactions, interpersonal interaction sites, web servers, and so forth throughout this way, observing and stock arrangement of all instrumentation may be enhanced and remains over organized and additionally unstructured structure. Today's benefits of the business requisitions are Hosting endeavor characteristics like substantial scale, data-intensive, web-oriented, etc. Furthermore these unorganized data sets are accessed from different gadgets including versatile apparatuses. The expression “NoSQL Database” may be utilized for expansive information sets whose extent will be past the capability from claiming

regularly utilized product instruments to capture, manage, and procedure the information inside a bearable passed time.[1] NoSQL database focus at present extending from a couple dozen terabytes should a lot of people peta bytes of information over a single information set. Challenges incorporate capture, storage, search, sharing, analytics furthermore visualizing. Commonplace cases from claiming NoSQL database discovered over current situation incorporates web logs, sensor networks, satellite geospatial data, social information starting with social networks, web content documents, web quest indexing, bring point of interest records, astronomy, climatic science, genomics, biogeochemical, biological, surveillance, medicinal records, photography archives, feature archives, also vast scale ecommerce data.

2. RELATIONAL VS. NoSQL DATABASES

As mentioned earlier that now-a-days volume of unstructured and versatile data is continuously being generated through IoT devices. To handle such kind of data the strength of robust RDBMS is not sufficient and that is where the NoSQL databases came into limelight. RDBMS are capable of handling volumes of structured data only.

End-users today run multiple tasks/queries on a particular data-set simultaneously. To handle such queries on un-structured and huge volume of data NoQL databases are required. The focus of NoSQL databases is reliability, scalability and consistency. The crucial factor about NoSQL is that it can handle huge amount of data by adding more machines and improving the efficiency of data accessibility.

Following table 1 depicts the points of distinctions between Relational database and NoSQL databases.

Table 1- Relational database and NOSQL database

Features	Description of distinction
Schema	In contrast to Relational databases which are table based databases, NoSQL databases (sometimes also referred as distributed databases) are document based, key-value pairs, graph databases or wide-column stores which do not have standard schema definitions whereas NoSQL databases have dynamic schema for unstructured data.
Scalability	If Relational databases are vertically scalable then NoSQL databases are horizontally scalable.
Query Language	SQL is available for Relational databases. The queries in NoSQL

	database are focused on documents' collection and the query language is Unstructured Query Language(UnQL). The syntax of UnQL usage changes in different databases.
Support to Data storage	Relational databases are not good choice for hierarchical data storage. Whereas, NoSQL database can always be favored for the hierarchical data storage because it tracks the key-value pair nature of storing data which is similar to JSON data. NoSQL database are highly preferred for big data.
Database Properties	Relational databases emphasize on ACID properties (Atomicity, Consistency, Isolation and Durability) whereas the NoSQL database follows the Brewers CAP theorem (Consistency, Availability and Partition tolerance)
Classification	Relational databases may not be classified on the basis of data nature or data stores. NoSQL databases can be classified on the basis of way of storing data as document store databases, column store database, graph databases, key-value store databases, and XML databases.
Examples	Examples of Relational databases are: MySql, Oracle, Postgres , MS-SQL etc. Whereas the examples of NoSQL databases are: MongoDB, BigTable, Hbase, Redis, RavenDb, CouchDb etc.

Since the appearance of relational database management system (RDBMS), most of the recent information systems are built by utilizing it. RDBMS uses foreign-keys to avoid data duplication. Also, it has very high reliability and portability because it supports standard structured query language (SQL) [2] The transactions in the database use attributes, such as atomicity, consistency, isolation, durability (ACID), which ensures that data integrity and processing results are stably managed. The characteristic of RDBMS is that there is high data reliability. However, this results in performance degradation. Meanwhile, from among these information systems, some systems only require high-performance rather than high reliability. In this case, if we only consider performance, the use of NoSQL provides many advantages. Like in data transmission protocol we have an appealing choice of UDP instead of TCP irrespective of data loss. It is possible to reduce the maintenance cost of the information system that continues to increase in the use of open source software based NoSQL. And has a huge advantage that is easy to use NoSQL..

NoSQL is the general name for the collection of databases that do not use SQL or a relational data model [3]. NoSQL is a useful database for application development productivity increase and for dealing with large amounts of data. In particular, it is used for rapid data

storage, increase of data concurrency and easy integration of the data. According to the report by Digital Universe, the amount of digital information is expected to more than double every two years. From this prospective, it is predicted that the most frequently used RDBMS will reach a limit of processing rapidly growing data [5]. The NoSQL data processing technique emerged as a solution. The schema of NoSQL is not fixed, and NoSQL is used as a method of storing data using key-value. Also, it does not generally guarantee the integrity of the database. In addition, it ensures higher performance than RDBMS by allowing data duplication and the high throughput of database systems.

NoSQL does not use relational data model. It is quiet useful database for application development, productivity increase and for dealing with large amounts of data. It can run well on clusters and it does not have defined schema which facilitates in handling any kind of unstructured data. The popularity index of NoSQL database is rising high as mostly it is an open-source in nature.

2.1 FEATURING NoSQL

The major hassles developer face while importing/exporting data to a different format of RDBMS. Besides that exchange of data between relational data structures and in-memory data structure of the application NoSQL is a solution for both mentioned problems.

NoSQL makes use of the Sharding method, which stores divided data into other servers. The unit of data is a shard, which is divided in Sharding, and dispersing and storing each shard that is split techniques are Feature-based Shard, Key-based Sharding, the Lookup table, and so on.[4]

There is one critical issue with NoSQL that it does not guarantee all three features (Consistency, Availability, Partition Tolerance) at the same time. According to the CAP theory in any system only two features can be picked.

Cap theory says to pick any two of them for effective performance AP, CP or CA

2.2 CAP Theorem

CAP theorem enables DDBS designers to choose two of three desirable properties: consistency (C), availability (A), and partition tolerance (P). Hence, only CA systems (consistent and available, but not partition-tolerant), CP systems (consistent and partitiontolerant, but not available), and AP systems (available and partition-tolerant, but not consistent) may occur.[6] Refer Figure 1.

2.2.1 Consistency – All the servers over the network will have same copy of data. So, whichever server will answer the request will provide the similar set of data. Consistency, informally, simply means that each server returns the right response to each request, i.e., a response that is correct according to the desired service specification. However, multiple possible correct responses may prevail.[7]

2.2.2 Availability- Request will always be responded (If the server is not in a working position then too responding that “System is not working”. Availability simply means that each request eventually receive a response. Noticeably, a fast response is generally found better than a slow

response, but for the purpose of CAP, it turns out that even requiring an eventual response is sufficient. Practically, a response that is sufficiently late is just as bad as a response that never occurs.[7]

2.2.3 Partition Tolerance- Here, the system continues to function as one unit even if an individual server fails or it can't be reached. In contrast to other two properties, this property can be realized as a statement regarding the basic system: communication among the servers is not reliable, and the servers may be partitioned into multiple groups that can't communicate with each other. For some purposes, we simply treat communication as faulty whereby messages may be delayed or lost forever. Again practically, it is worth mentioning that a message that is delayed for very long may be considered lost as well.[7]

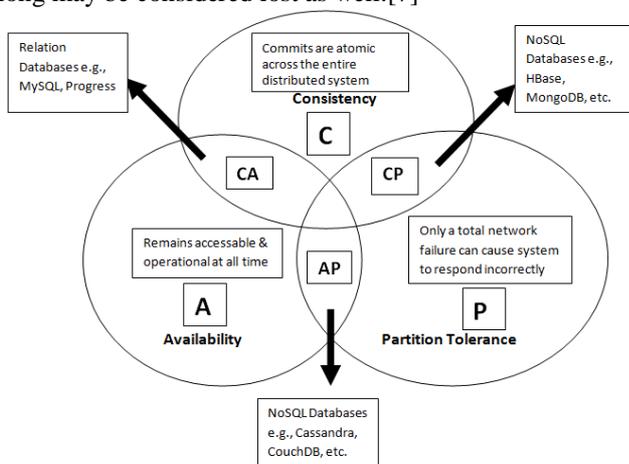


Figure 1: CAP Illustration

3. ARCHITECTURE OF NOSQL DATABASES

The architecture of NoSQL databases is flexible; it depends upon the type of data that is to be stored in this. The NoSQL databases are available in different forms. Each form of them has its unique features. NoSql databases can be classified as follows-

3.1 Key-Value Store Databases-

The 'key-value' data stores have simple application programming interfaces (APIs). A key value data store allows user to store the data in a 'schema-less' way. The data consists of two parts, first one is a string that represents the key and the second one is the actual data which is to be referred as value thus making a "key-value" pair. These kinds of data stores are similar to hash tables where the keys are used as indexes. This approach makes it faster than RDBMS. The modern key-value data stores prefer high scalability over consistency. One of the weaknesses of key-value data store is the lack of schema that makes it very difficult to create custom views of the data. Key-value data stores can be used in situations where we want to store a user's session or a user's shopping cart or to get details like favorite products. Key-value data stores can be used in forums, websites for online shopping etc. [10]

3.2 Document Database -

These databases are synonymous to content management system like web analytics, real-time analytics, blogging platforms, web analytics and many more. Document databases are not used for systems based on complex transactions where we need to operate on multiple operations based on complex queries.

Document database contains documents which are similar to records in a table that narrates the explanation of data in a document. Documents can be complex as well as simple. Document database can also use nested data. Unlike relational databases where schema is well defined in advanced. In document database we need not to define the logical structure. Instead of columns and their data types only structure of the document need to be defined.

Document databases offer wonderful performance with horizontal scalability. 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 dissimilar as well as similar data. Documents in the database are addressed using a unique key. Document stores are slightly more complicated as compared to key-value stores as they allow to cover the key-value pairs in document also known as key-document pairs. Document databases should be used for applications where data needs not to be stored in a table with identical sized fields. However, the data has to be stored as a document containing special characteristics/ features. Document databases will serve excellent when the domain model can be split and partitioned across some documents. Document database stores should always be avoided if the database contains a lot of relations and normalization. These databases can be favored for content management system, blog software etc.[10]

3.3 Column-Oriented Databases-

Column data stores in NoSQL are in fact a hybrid row/column data store contrasting pure relational column databases. Even though, it shares the theory of column-by-column storage of columnar databases and columnar extensions to row-based databases, column stores don't store data in tables but store it in extremely distributed architectures. In column stores, each key is associated with one or more attributes (columns). A Column oriented database stores its data in such a way that it can be combined rapidly with less I/O effort. Such databases offer high scalability. The data that is stored in such database is based on the sort order of the column family. Column oriented databases are suitable for analytic applications and data mining. In these applications the storage methods are ideal for the common operations executed on the data.[10]

3.4 Graph Stores -

Graph databases store data in the form of a graphs. 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 comprises of characteristics/

properties related to nodes. It uses a technique called 'index free adjacency' which means that each node consists of one direct pointer which points to an adjacent node. Score of millions of records can effectively be traversed using this technique. In a graph databases, the main emphasis is given to the connection between data. Graph databases provides schema less and efficient storage of semi structured data. The queries in such stores are expressed as traversals which make graph databases faster than relational databases. It also has good scalability. Graph databases are ACID compliant (as RDBMS) and offer rollback support. Graph databases can also be used for a variety of applications like recommendation software, social networking applications, content management, bioinformatics, security and access control, network and cloud management etc.[10] It is very difficult to achieve 'sharding' in Graph databases. Graph databases are difficult to cluster.

4. CONCLUSION-

In the database systems the NoSQL databases have been considered to be quite new. However, these are being developed on known and existing theory of Relational ones. NoSQL databases systems still have various limitations. The NoSQL database architecture is also having variant nature. There is neither a common standard nor any common query language for querying NoSQL databases. Of course, it seems impractical for big data. Yet, New more accurate, formal and common query system may be evolved in times to come. Each NoSQL database behaves in a different way and does things differently. Relatively these databases are immature and constantly evolving. NoSQL database does not support strict ACID properties, hence there is no guarantee of successful storing into the data store. This article describes the limitation of relational database along with CAP and different categories of NoSQL databases. In absence of specific tool this article compares the strength and limitation of each the data model on conceptual basis only. Hence, Functioning concomitantly with NoSQL database seems to be more challenging. Nevertheless, substitute of CAP needs to be worked upon with NoSQL databases.

Limitations of NoSQL databases and its use in a cloud computing environment are the areas which need more research in future.

REFERENCES-

- [1.] <https://www.thoughtworks.com/insights/blog/nosql-databases-overview>
- [2.] E. Hewitt, Cassandra: The Definitive Guide. Beijing: O'Reilly, 2011.
- [3.] 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 <http://dx.doi.org/10.3745/JIPS.04.0006>
- [4.] E. F. Codd, "A relational model of data for large shared data banks," Communications of the ACM, vol. 13, no. 6, pp. 377-387, Jun. 1970.

- [5.] J. Gantz and D. Reinsel (2011, June). "Extracting value from chaos," IDC iView [Online]. Available: <http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>
- [6.] Daniel J. Abadi, Yale University, "Consistency Tradeoffs in Modern Distributed Database System Design", 0018-9162/12/\$31.00 © 2012 IEEE Published by the IEEE Computer Society FEBRUARY 2012
- [7.] Seth Gilbert, Nancy A. Lynch, "Perspectives on the CAP Theorem" National University of Singapore, Massachusetts Institute of Technology
- [8.] Abhishek Prasad , Bhavesh N. Gohil, "A Comparative Study of NoSQL Databases", International Journal of Advanced Research in Computer Science, Volume 5, No. 5, May-June 2014, ISSN No. 0976-5697
- [9.] Biswajeet Sethi, Samaresh Mishra, Prasant Kr. Patnaik, "A Study of NoSQL Database", Internation Journal of Engineering Research & Technology (IJERT) ISSN:2278-0181, Vol. 1, Issue 4, April 2014
- [10.] Ameya Nayak, Anil Poriy, Dikshay Poojary, "Type of NOSQL Databases and its Comparison with Relational Databases", International Journal of Applied Information Systems (IJ AIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5– No.4, March 2013 – www.ijais.org
- [11.] <https://www.slideshare.net/rahuldausa>



Dr. Brijesh Khandelwal presently working as Associate Professor at Amity School of Engineering & Technology, Amity University Chhattisgarh, Raipur. Dr. Khandelwal has rich & diverse experience in academia and has several publications in International/ National Journals & Conferences. Dr. Khandelwal did MCA from Lucknow University in year 1994. In 2001, he became Sun Certified Programmer with Sun Microsystems. He was awarded PhD (Applied Economics) from University of Lucknow in year 2007. He did MBA in 2010 from Punjab Technical University. In 2010 he also became licentiate in Life Insurance from Insurance Institute of India, Mumbai. He also has been awarded PhD on Computer Science in year 2017 from Shri Venkateshwara University, Gajraula.