IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727
PP 45-51
www.iosrjournals.org

Sql Vs NoSql: NewSql The Solution For Big Data

Dr. Archana Raje[1], Aniket Jagdale[2]
[1](Information Technology, K. J. Somaiya Institute of Management Studies and Research, India)
[2](Information Technology, K. J. Somaiya Institute of Management Studies and Research, India)

Abstract: *SQL Databases also referred to as RDBMS (Relational management Systems). RDBMS is the most typical and ancient approach to database solutions. The information is kept in a very structured approach in style of tables or relations. With advent of Big Data however, the structured approach falls short to serve the wants of massive information systems that are primarily unstructured in nature. Increasing capability of SQL though permits vast quantity of information to be managed; it doesn't really matter as an answer to Big Data systems that expects quick response and fast quick scalability.*
To solve this drawback a reasonably new database system referred to as NoSQL was introduced. This NoSQL system is introduced to provide the quick scalability and unstructured platform for Big Data applications. NoSQL is also called as Not Only SQL. NoSQL databases encompass key-value pair, Documents, graph databases or wide – column stores that don't have a typical schema that it has to follow. It is also horizontally scalable as in comparison with vertical scaling in RDBMS.
NoSQL provided great promises to be an ideal database system for Big Data applications; it however falls short thanks to some major drawbacks like NoSQL doesn't guarantee ACID properties (Atomicity, Consistency, Isolation and Durability) of SQL systems. It is also not compatible with earlier versions of databases. This is where NewSQL comes into limelight. NewSQL is a latest development within the world of databases systems. NewSQL is a relational database with the measurability properties of NoSQL. This paper discusses each of those database systems and tries to seek out the perfect answer for Big Data necessities.

Keywords: *NewSQL, Big Data, Features of NewSQL, Difference between SQL, NoSQL and NewSQL, OLTP (On-LineTransaction Processing) and Big Data, ACID properties, BASE properties, CAP Theorem.*

I. Introduction

Huge amount of information is generated every day. Almost, 2.5 quintillion bytes information that is ordinarily created every day, normally through social media sites. This can be handled by NoSQL databases, Hadoop and Hbase. It is noticed that standard language of SQL is not suitable for volume, velocity and variety data. Thus it's not appropriate for cloud based applications as a result of its limitations to their strict straightforward plan requirements. To fulfill these prerequisites and to remove this limitation of SQL, NOSQL was introduced. Thus to handle a 2.5 quintillion of information, NOSQL had structure to handle such large quintillion information. NOSQL takes care of the SQL issue by having problems such as adaptability. However, this developed to new clashes, for the entire part absence of current access and consistency alternatives and due to OLTP workload that prompted the gathering of NEWSQL. NEWSQL is something that consolidates the elements of each SQL and NOSQL and components of both meets into NEWSQL [1]. It provides constant measuring undertaking of NOSQL frameworks and appropriate for taking care of ACID certifications of SQL information frameworks.

Thus, to compare and contrast SQL, NoSQL and NewSQL databases, in this paper we mentioned SQL concepts followed by NoSQL concepts. For understanding benefits of these databases, we also stated some of prevalent SQL and NoSQL databases. Therefore, to determine whether SQL or NoSQL is better for big data applications, we specified difference between them. The best features of both are combined in NewSQL to provide as solution for big data applications.

II. SQL Database Concepts

2.1 ACID Properties

SQL databases transactions follows ACID (Atomicity, Consistency, Isolation, and Durability) properties to maintain reliability of transactions. This includes atomicity of a transaction that requires completeness of transaction to the fullest else complete rollback. Consistency assures stable state of database i.e. with changes or without changes made by transaction. Isolation states that multiple transactions do not interface with each other. Durability of transaction concentrates on its permanent effect of transaction in the database.

2.2 Normalization

Normalization is a process of designing databases. The most common normal forms are:

- **First Normal Form (1NF)**: In this normal form, split the tables by separating repeating and non-repeating attributes in different tables. In this form, all domains are simple and in a simple domain, all elements are atomic.
- **Second Normal Form (2NF)**: In this normal form remove partial dependency between table attributes. No attributes of the table (or relation) should be functionally dependent on only one part of a concatenated primary key.
- **Third Normal Form (3NF)**: In this normal form remove transitive dependency between table attributes. Thus, no non-prime attribute is functionally dependent on another non-prime attribute.

It is a prerequisite for a database that it should satisfy the conditions of 1NF and 2NF to fulfill 3NF respectively.

2.3 Scalability

Scalability is the capability of a database to handle a growing amount of data, or its potential to be enlarged to accommodate huge growth rate of increasing data. Database scalability means the ability of a system's database to scale up or down as per the requirement. If the databases do not support scalability then it will damage the business operations. Vertical scaling helps in upgrading the capacity of the existing database server. Most of SQL database supports vertical scaling.

III. NOSQL Database Concepts

The term NoSQL was used by Carlo Strozzi in 1998 to name his lightweight, Strozzi NoSQL open-source relational database that did not expose the standard Structured Query Language (SQL) interface [2]. These days "NoSQL" stands for "Not Only SQL". Edlich et.al. States it is difficult to find common definition for NoSQL-databases [3]. NoSQL-databases don't support a relational approach. They Scale horizontally and are normally available as open-source products. They don't follow defined schema instead follows dynamic schema. For easy integrations with different software products NoSQL provide an API and use a decentralized architecture for the easy replication of data. These databases follow the BASE principle.

3.1 BASE Principle of transactions

BASE is to NoSQL what ACID is to SQL. BASE is the property on NoSQL databases that ensure its reliability in spite of loss of Consistency. BASE stands for Basically Available Soft state eventually consistent.

- **Basically Available**- This states that the system guarantees availability of the data.
- **Soft state** – The system state may change at any time, even when no input is given to the system.
- **Eventually Consistent**- The system can eventually become consistent as its state will modify once not receiving inputs. This suggests that sooner or later the information is updated wherever necessary thus maintaining the consistency of the database.

3.2 CAP Theorem

Eric Brewer suggested, CAP Theorem which is comprises of properties namely Consistency, Availability, and Partition tolerance. Consistency refers that data existing on all machines must be identical after all update operations. Availability guides that data should be made accessible permanently instead of temporary access. Even in case of machine failure and defects database should work properly without any half is known as partition tolerance.

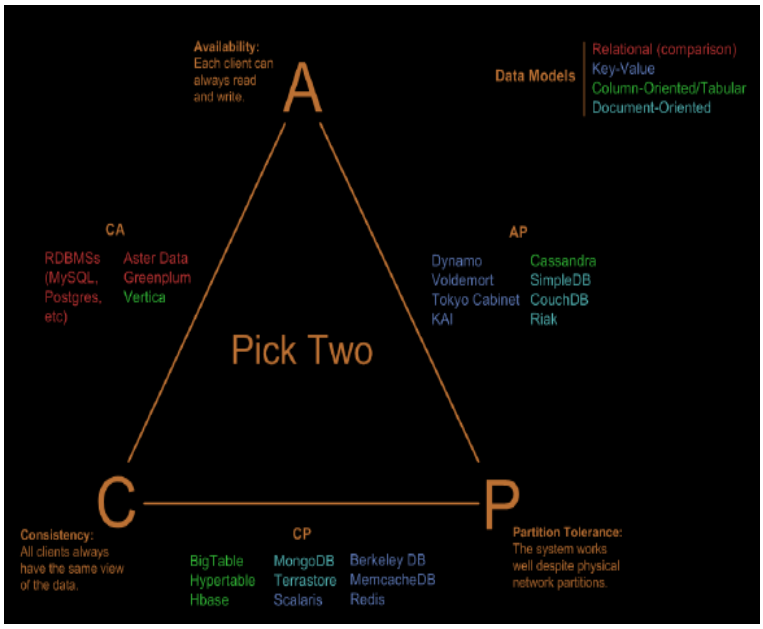

Fig 1[4]. CAP Theorem

3.3 Lacking schema

No specific schema needs to be defined before entering the data into NoSQL databases. The schema construction for the data being entered can be done any moment without affecting applications using them.

3.4 Auto balancing

NoSQL divides your data among multiple servers automatically, with no assistance required from applications.

3.5 Integrated caching

NoSQL database cache data in system memory, so that it can increase data throughput and increase the performance in advance.

IV. Prevalent SQL Databases

4.1 MySQL Community Edition

MySQL Database is frequently used, popular open-source database. The MySQL Community Edition includes Pluggable Storage Engine Architecture and Multiple Storage Engines such as InnoDB, MyISAM, NDB (MySQL Cluster), Memory, Merge, Archive, CSV, and more. MySQL advantages includes replication which replicates database across multiple host and servers. Shrading operation which is beneficial for highly scalable approach for improving the throughput and overall performance of high-transaction.

4.2 MS-SQL Server Express Edition

A Microsoft product MS-SQL, is a database having good reliability, quantifiability, stability options. It is amazingly dominant and user friendly database. This can be utizied for developing variety of applications suitable for web and mobile handsets for several data types. It supports both structured or as well as unstructured data and with built-in support for relational data, XML, and spatial data—plus, increase temporal data granularity with date and time data types. The advantages and strengths of MS-SQL includes various tools which are helpful for intergrated development environment. It's mirroring mechanicm which is usefel for disaster recovery and cloud back-up support.

4.3 Oracle 11g Express Edition Database

Oracle database has benefits such as easy upgradation to new and advanced version. Varied support across multiples operting systems such as Linux and Windows. This also provides facilites which are easily managable, productive, secure and reliable.

V. NOSQL Databases

5.1 Key-Value Store Databases

One of the type of NoSQL database is a Key-Value Store databases. This is a data storage paradigm designed for storing, retrieving and manageing associative arrays which is a data structure known today as a dictionary or hash. This databases treat the data as single opaue collections which may have different fields for

every record which results in considerable fleexibility. It is observed they often results providing benefits in term of saving memory and performace enhancement.

Instance of Key-Value Store Databases includes Amazon DynamoDB and Riak. This design is suitable for internet scalable applications because of it's high reliable, fast and cost-effective service of NoSQL. For faster access of data it stores data on solid state drives as different than traditional hard drives.This feature provides high readiness and stability in complicated failure conditions.

Another example RIAK is developed by Basho technologies which is built on a set of core services providing highly reliable, scalable distributed framework. This is highly optimized for IoT and time series data.

5.2 Column-Oriented Databases

Data stored in Column-Oriented databases that stores data tables by column rather than a row. Thus, the database can access the data it needs more precisely to answer a query instead of scanning entire data. This reults in faster query performance of large databases. **HBase and Cassandra** are examples of column-oriented databases.

5.3 Document Store Databases

Document Store Databases are designed for storing, retrievinand and managing document-oriented information which is semi-structured data. In document store databases it accepts the documents in any format such as XML, JSON, PDF etc. **MongoDB, CouchDB** and **DocumentDB** are the examples of document store databases.

5.4 Graph Databases

In Graph Databases, it uses graph structure for scemantic quires with Graph with nodes,edges and properties to represent and store data. In this the graph (or edge or relationship) directly relates data items in the store. Graph databases in comparison to relational databases are faster and easily accessible as queries are expressed as traversals. **Dgraph, OreintDb, SAP HANA** and **Neo4j** are examples of graph databases.

5.5 Object-Oriented Databases

An Object Oriented databases represents information in the form of objects same as object oriented programming. Relational databases are table oriented. Object relational databases uses a hybrid of both approaches. This supports almost all major object oriented concepts. These databases suggests class as tables, object as tuple and class attributes as columns in relational databases. Object oriented databases uses agile development methodology. **db4o, Smalltalk** and **Cache** are the examples of object-oriented database engines.

VI. SQL In Bigdata

Structured Query Language has ruled over database world for many decades and is presently being invested in by Big Data companies and organization. [5] Data that is not interactive becomes useless and it is not beneficial to use them. Therefore, we use SQL which enables interaction with data and allows a broad question to be asked against a single database design. [5] Since SQL is schema-oriented , the structure of the data should be known in advance which is difficult to obtain in big Data. Also, SQL doesn't have the capabilities to process unpredictable and unstructured information. Hence, Currently companies are switching to NoSQL so these flaws of SQL will be fixed.

VII. NOSQL Better For Bigdata Applications

The data in Big Data applications varies widely. The data is collected from different sources like social media, mobile phones, etc. The data can be personal information of the user, location data, machine data, sensor generated data, etc. To handle such a data scalability and flexibility is of utmost importance.

Scaling in SQL systems means spending money on expensive hardware at a single node. This vertical scaling is not an effect or action but an economical approach. NoSQL is horizontally scalable can be implemented easily within the Big Data applications.

Scalability in NoSQL is as easy as adding a server node into the system [6]. The load on the system is thus shared between the nodes. Flexibility is already present within the NoSQL databases as it does not have to be fixed or restricted to a certain schema unlike Relational Databases.

VIII. Drawbacks Of NOSQL

NoSQL is still in its infant stage. There is a long way to go for it to become richly functional and stable system. Because of still being in the early stage there are very less advanced expertise in this field [6].

BASE properties which are provided by NoSQL are unreliable in nature in comparison to ACID properties provided by SQL databases. ACID properties of transactions are vital in various cases such as banking firms.

ACID	BASE
ACID represents **A**tomicity, **C**onsistency, **I**solated and **D**urability.	**BASE** represents **B**asically **A**vailable, **S**table state, **E**ventually consistent.
Mainly focused on Consistency and Availability.	Mainly focused on Availability and Partition tolerance.
Strong Consistency is provided.	**Weak Consistency** is provided.
Pessimistic approach is followed.	**Optimistic** approach is followed.
Complex mechanisms are implemented.	**Simple and fast** to use.
Primarily used where data reliability and consistency is very important.	Primarily used where data availability and speed is important.

IX. Difference between SQL and NOSQL

Relational Databases (RDBMS) are referred to as SQL databases and Non-relational/distributed or online databases represents NoSQL databases.

	SQL Databases	NoSQL Databases
Development History	In 1970 to work with first generation of data storage and applications	In 2000 to suppport scalability and replication. Also suitable for unstructured data
Forms	Relational schema table oriented databases	There are different forms of NoSQL Databasses such as Document store databases, Key-value databases, Column-Oriented databases, Object oriented databases and graph databases
Schemas	Each record adhere to static schema definition which must be finalised and sealed before data entry	Schemas are dynamic.Informaion can be added on the at runtime.
Scalability	Basically vertically scalable	Horizontally scalable, meaning allocate data across different servers
Support for ACID Property	Commonly are ACID obedient	Do not support ACID property of transaction for improving performance and scalabiltiy
Query Language	Queries are developed using Structured Query Language(SQL)	Queries are based on collection of documents also called as Unstructured Query Language (UnQL). Syntax varies as the database changes.
Examples	Microsoft-SQL Server, Oracle, MySQL and SQLite.	MongoDB, CouchDb, BigTable, Redis, Cassandra, Hbase, Neo4j, Riak, DynamoDB and db4o
For complex queries	Perfect fit for the complex queries	Not ideal for complex queries
Data type	Not much ideal for hierarchical data storage	Relatively better for the hierarchical data storage as it follows the key-value combinations to store data. Used for large datasets (i.e. for big data)
For high transactional based application	Perfect fit for high OLTP type applications,	Not ideal for high workload and complex OLTP operations
Vendor support	Outstanding support and updates is given for all SQL databases by vendors	Need to depend on community support as it is a open-source.
Data Manipulation	Operations are done using Select, Insert and Update statements	Operations are done using different object-oriented API's

X. NEWSQL

NewSQL is a technology that focuses at making current relational SQL highly scalable. It's an endeavor to combine NoSQL and SQL. SQL provides ACID properties but isn't fast enough when it comes to concurrency. NoSQL aims at Brewers CAP theorem but doesn't necessarily provide ACID properties. NewSQL tries to provide relational DBMS that has same scalability as NoSQL for OLTP while still providing ACID properties [7].

XI. Architecture Of NEWSQL

An ideal DBMS should scale elastically vertically as well as horizontally. It should allow new machines to be hosted easily in a system that's already up and running [8]. This wasn't possible to be performed efficiently with SQL. So NewSQL uses technology that's used in cloud computing and distributed applications. It implements distributed database technology. The databases are generally distributed. They follow the three tier architecture having three layers:

An administrative tier, a transactional tier and a storage tier [8]. SQL provided vertical scaling but there was no provision for horizontal scaling. The NewSQL model provides horizontal scaling along with vertical scaling. The databases used are distributed while still providing ACID properties. They perform efficiently in heavy workload and are very secure and robust. NewSQL provides high efficiency for distributed services.

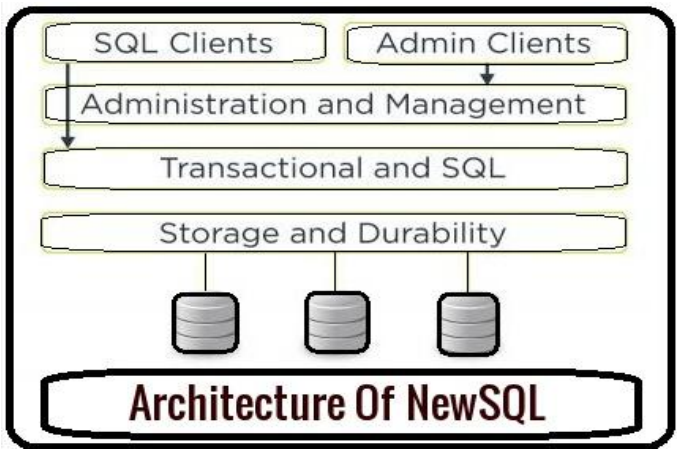

Fig 2[9]. NewSQL Architecture

XII. Properties Of NEWSQL

NewSQL is a relational database. It supports ACID properties. Its schema is a combination of SQL and NoSQL. It provides horizontal scalability. It has cloud support and can also be used for OLTP. It supports SQL but query complexity is very high. NewSQL gives high performance by keeping all data in RAM. Scalability is acquired by employing partitioning and replication in such a way queries usually do not have to communicate between multiple machines. They get the required information from a single host. This is why NewSQL is the best option for those who want to develop highly scalable and efficient OLTP systems.

Distinguishing Feature	Old SQL	NoSQL	NewSQL
Relational	Yes	No	Yes
ACID	Yes	No(Provides CAP)	Yes
SQL	Yes	No	Yes
OLTP	Not fully supported	Supported	Fully supported
Scaling	No	Yes	Yes
Complex Query Handling	Little	Great	Excessive
Distributed	No	Yes	Yes

XIII. NEWSQL and Bigdata

Big data is the term for a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications [10]. The amount of information is so huge it requires thousands of servers running in parallel to work with this data. It involves many challenges like capturing of data, storage of data, processing and analysis of data. NewSQL allows you to work with this Big Data more efficiently.

13.1 Application Of NewSQL: GoogleSpanner

One application of NewSQL is Google Spanner. Spanner is a globally distributed database which is scalable. It is deployed at Google. It is based on NewSQL. It is a database that performs sharding of data that is horizontal partition of data and it is spread across many Paxos state machines [11]. Paxos state machines are used to solve consensus which is the process of agreeing upon one result when there are multiple participants.

The datacenters are spread all over the world. The databases should be available globally. So for this purpose replication is used. The replication isn't completely random. It considers the geographic locality and what kind of data is required more frequently. Spanner works dynamically. It reshards and migrates data automatically for the purpose of load balancing [11]. For achieving low latency and high availability most applications would probably replicate data over three or five datacenters in one geographic region. Spanner is useful when applications want strong consistency and are distributed over a large area. Spanner performs versioning of the data and stores the time stamp which is the same as the commit time. Unrequired data can be deleted with proper policies for handling old data. Spanner is very useful for OLTP concerning Big Data. It uses SQL query language.

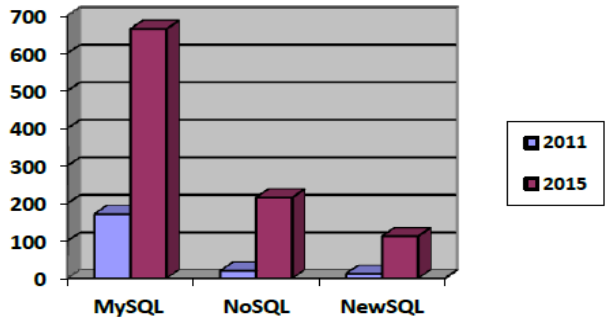

Fig 3[3]. Increase in revenue from 2011-15

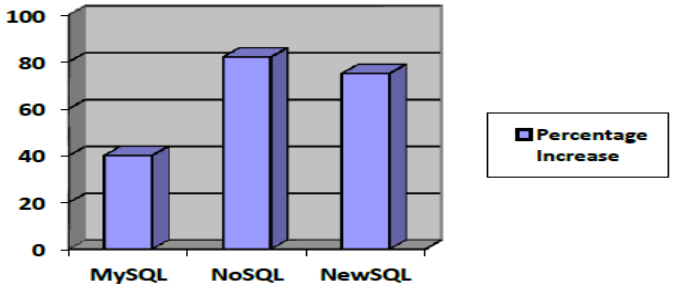

Fig 4[3]. Percentage increase in 2011-15

XIV. Conclusion

SQL provides ACID compliancy with vertical scalability while NoSQL is perfect fit for horizontal scaling providing BASE properties. However NoSQL does not consists of ACID properties which are necessary for a reliable and consistent database. The demand of modern enterprises and organizations where data is growing day by day and all they work with is Big Data especially even while working in OLTP system, NewSQL is the ideal choice. NewSQL is improvement of SQL providing horizontal scalability while maintaining ACID properties. This also allows working with Big Data by acquiring the ability to work concurrently, it also maintains ACID compliancy. NewSQL has found the perfect spot between consistency, scalability, speed and availability. While still being in its earlier stage, NewSQL ticks all the right boxes to be an ideal database for Big Data and OLTP applications.

References

[1]. http://www.ijptonline.com/wp-content/uploads/2016/10/18351-18361.pdf
[2]. Supriya S.Pore, Swalaya B. Pawar, "Comparative Study of SQL & NoSQL Databases", May 2015.
[3]. Matthew Aslett :"451 Research delivers market sizing estimates for NoSQL, NewSQL and MySQL ecosystem" May 22nd, 2012
[4]. https://mongodbforabsolutebeginners.blogspot.in/2016/06/acid-and-cap-theroems.html
[5]. Ryan Betts : "SQL Time-Tested and still flourishing" , VoltDB, Bedford.
[6]. Jenny Richards, Advantages and Disadvantages of NoSQL databases – what you should know, Hadoop360, September24,2015,http://www.hadoop360.com/blog/advantages-anddisadvantages-of-nosql-databases-what-you-should-k
[7]. Stonebraker, Michael "NewSQL: An Alternative to NoSQL and Old SQL for New OLTP Apps". Communications of the ACM Blog. , 2012
[8]. A B M Moniruzzaman: "NewSQL: Towards Next- Generation Scalable RDBMS for Online Transaction Processing (OLTP) for Big Data Management", Nov 2014.
[9]. Rakesh Kumar "Critical Analysis Of Database Management Using NewSQL" International Journal of Computer Science and Mobile Computing, Vol.3 Issue.5, May-2014
[10]. Vinay Jain "Rise of NewSQL". International Journal for Research in Emerging Science and Technology.
[11]. Hoff, Todd : "Google Spanner's Most Surprising Revelation: NoSQL is Out and NewSQL is In".Retrieved 2012-10-07.