ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

NoSQL: Database for Distributed Systems a Survey

Nusrat Jahan[1], Kusum Munde[2]

Assistant Professor, Dept. of Computer Engg, S.R.C.O.E, Savitribai Phule Pune University, Maharashtra, India[1]

Assistant Professor, Dept. of Computer Engg, S.R.C.O.E, Savitribai Phule Pune University, Maharashtra, India[2]

ABSTRACT: The recent development in the internet market and the growing of new IT technologies with new difficulties and new ideas such as NoSQL which is now becomes a very popular instead to the relational databases particularly when working with the big data. This paper contains introduction of NoSQL database along with the important variations between conventional relational databases and NoSQL databases. This paper also describes ACID properties and CAP theorem. Relational databases are not suitable for modern web applications that can support an incredible number of contingency users by growing the burden across a assortment of application web servers behind a lot balancer.

KEYWORDS: NoSQL; SQL; ACID rules; CAP theorem; BASE properties.

I. INTRODUCTION

In the computing system (web and business applications), there are enormous data that comes out every day from the web. A large section of these data is handled by Relational database management systems (RDBMS). The idea of relational model came with E.F.Codd's 1970 paper "A relational model of data for large shared data banks" [1] which made data modelling and application programming much easier. Beyond the intended benefits, the relational model is well-suited to client-server programming and today it is predominant technology for storing structured data in web and business applications.

Over the last few years we have seen the rise of a new type of databases, known as NoSQL databases that are challenging the dominance of relational databases. NoSQL was developed in late 2000s to deal with limitations of SQL databases, especially scalability, multi-structured data, geo-distribution and agile development sprints. Motivations for this approach include: simplicity of design, simpler "horizontal" scaling to clusters of machines and finer control over availability. 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.

II. CLASSICAL RELATIONAL DATABASE FOLLOW THE ACID RULES

A database transaction must be atomic, consistent, isolated and durable. Often these four properties of a transaction is acronymed as ACID. Below we have discussed these four points [2].

Atomic: A transaction is a logical unit of work which must be either completed with all of its data modifications, or none of them is performed at all.

Consistent: At the end of every transaction, all data must be left in a consistent state.

Isolated: Modifications of data performed by a transaction must be independent of another transaction. Unless this happens, the outcome of a transaction may be erroneous.

Durable: When the transaction is completed, effects of the modifications performed by the transaction must be permanent in the system.

DOI: 10.15680/IJIRCCE.2016. 0411013

III. DISTRIBUTED SYSTEMS

A distributed system consists of multiple computers and software components that communicate through a computer network (a local network or by a wide area network). A distributed system can consist of any number of possible configurations, such as mainframes, workstations, personal computers, and so on. The computers interact with each other and share the resources of the system to achieve a common goal [3].

The challenges arising from the construction of distributed systems are the heterogeneity of their components, openness, security, scalability – the ability to work well when the load or the number of users increases – failure handling, concurrency of components, transparency and providing quality of service.

With the development of the Internet and cloud computing, there need databases to be able to store and process big data effectively, demand for high-performance when reading and writing, so the traditional relational database is facing many new challenges and become inadequate to process large data and hence NoSQL database created. NoSQL is designed for distributed data stores where very large scale of data storing needs required.

IV. ADVANTAGES/ DISADVANTAGES OF DISTRIBUTED COMPUTING

A. DISTRIBUTED COMPUTING SYSTEM HAS FOLLOWING ADVANTAGES [4]:

- **Reliability (fault-tolerance):** The important advantage of distributed computing system is reliability. If some of the machines within the system crash, the rest of the computers remain unaffected and work does not stop.
- **Scalability:** In distributed computing the system can easily be expanded by adding more machines as needed.
- **Sharing of Resources:** Shared data is essential to many applications such as banking, reservation system. As data or resources are shared in distributed system, other resources can be also shared (e.g. expensive printers).
- **Flexibility:** As the system is very flexible, it is very easy to install, implement and debug new services.
- **Speed: A** distributed computing system can have more computing power and its speed makes it different than other systems.
- **Open system:** As it is open system, every service is equally accessible to every client i.e. local or remote.
- **Performance:** The collection of processors in the system can provide higher performance (and better price/performance ratio) than a centralized computer.

B. DISTRIBUTED COMPUTING SYSTEM HAS FOLLOWING DISADVANTAGES [4]:

- **TROUBLESHOOTING:** Troubleshooting and diagnosing problems**.**
- **Software:** Less software support is the main disadvantage of distributed computing system.
- **Networking:** The network infrastructure can create several problems such as transmission problem, overloading, loss of messages.
- **Security:** Easy access in distributed computing system increases the risk of security and sharing of data generates the problem of data security.

V. WHAT IS NoSQL?

NoSQL is a non-relational database management system, different from traditional relational database management systems in some significant ways. NoSQL means Not Only SQL, implying that when designing a software solution or product, there are more than one storage mechanism that could be used based on the needs. NoSQL was a hashtag choosen for a meetup to discuss these new databases. It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users). These types of data storing may not require fixed schema, avoid join operations and typically scale horizontally.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 11, November 2016

VI. WHY NoSQL?

In today's time data is becoming easier to access and capture through third parties such as Facebook, Google+ and others. Personal user information, social graphs, geo location data, user-generated content and machine logging data are just a few examples where the data has been increasing exponentially. To avail the above service properly, it is required to process huge amount of data. Which SQL databases were never designed?

The evolution of NoSql databases is to handle these huge data properly. Application developers have been frustrated with the impedance mismatch between the relational data structures and the in-memory data structures of the application. Using NoSQL databases allows developers to develop without having to convert in-memory structures to relational structures.

The rise of the web as a platform also created a vital factor change in data storage as the need to support large volumes of data by running on clusters. Relational databases were not designed to run efficiently on clusters. Following fig.1 shows web applications driving data growth.

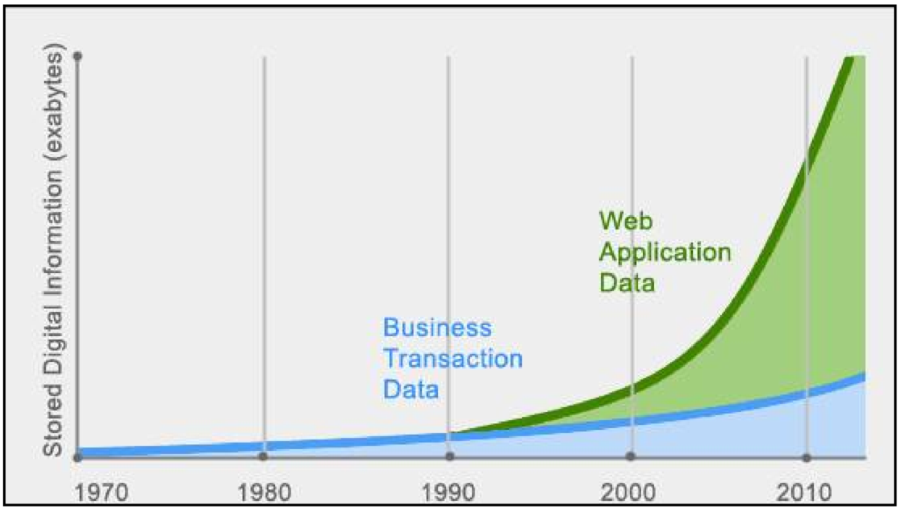

Fig. 1 Web Applications Driving Data Growth

VII. BRIEF HISTORY OF NoSQL

- The term NoSQL was coined by Carlo Strozzi in the year 1998. He used this term to name his Open Source, Light Weight, DataBase which did not have an SQL interface [5].
- In the early 2009, when last.fm wanted to organize an event on open-source distributed databases, Eric Evans, a Rackspace employee, reused the term to refer databases which are non-relational, distributed, and does not conform to atomicity, consistency, isolation, durability - four obvious features of traditional relational database systems.
- In the same year, the "no:sql(east)" conference held in Atlanta, USA, NoSQL was discussed and debated a lot.
- Graph database Neo4j is started in 2000.
- Google BigTable is started in 2004.
- CouchDB is started in 2005.
- The document database MongoDB is started in 2007 as a part of a open source cloud computing stack and first standalone release in 2009.
- Facebooks open sources the Cassandra project in 2008 [5].

And then, discussion and practice of NoSQL got a momentum, and NoSQL saw an unprecedented growth.

VIII. DIFFERENCE BETWEEN RDBMS AND NoSQL

A. ***RDBMS*****:** RDBMS database has following characteristics

- It contains Structured and organized data.
- Structured query language (SQL).
- Data and its relationships are stored in separate tables.
- Data Manipulation Language, Data Definition Language.
- Tight Consistency.
- BASE Transaction.

B. ***NoSQL*****:** NoSQL database has following characteristics [6]:

- NoSQL Stands for Not Only SQL.
- It has no declarative query language.
- It has no predefined schema.
- It uses databases like Key-Value pair storage, Column Store, Document Store, Graph databases [7].
- Eventual consistency rather ACID property.
- It contains Unstructured and unpredictable data.
- Uses CAP Theorem.
- Prioritizes high performance, high availability and scalability.

IX. CAP THEOREM (BREWER'S THEOREM)

We must understand the CAP theorem when we talk about NoSQL databases or in fact when designing any distributed system. CAP theorem states that there are three basic requirements which exist in a special relation when designing applications for a distributed architecture [8].

Consistency - This means that the data in the database remains consistent after the execution of an operation. For example after an update operation all clients see the same data.

Availability - This means that the system is always on (service guarantee availability), no downtime.

Partition Tolerance - This means that the system continues to function even the communication among the servers is unreliable, i.e. the servers may be partitioned into multiple groups that cannot communicate with one another.

In theoretically it is impossible to fulfil all 3 requirements [9]. CAP provides the basic requirements for a distributed system to follow 2 of the 3 requirements. Therefore all the current NoSQL database follow the different combinations of the C, A, P from the CAP theorem. Here is the brief description of three combinations CA, CP, AP [10]:

CA - Single site cluster, therefore all nodes are always in contact. When a partition occurs, the system blocks.
CP - Some data may not be accessible, but the rest is still consistent/accurate.
AP - System is still available under partitioning, but some of the data returned may be inaccurate. Fig. 2 shows CAP Theorem.

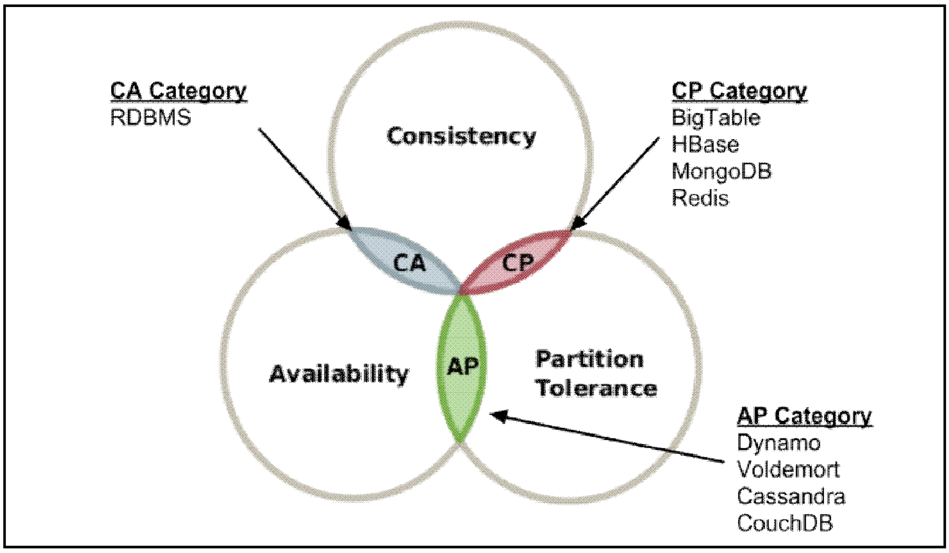

Fig. 2 CAP Theorem

The BASE Properties:

The BASE acronym was defined by Eric Brewer, who is also known for formulating the CAP theorem. The CAP theorem states that a distributed computer system cannot guarantee all of the following three properties at the same time:

- Consistency
- Availability
- Partition tolerance

A BASE system gives up on consistency [11].

- **B**asically **A**vailable indicates that the system *does* guarantee availability, in terms of the CAP theorem.
- **S**oft state indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model.

Eventual consistency indicates that the system will become consistent over time, given that the system doesn't receive input during that time.

X. NoSQL PROS/CONS

A. ***ADVANTAGES*****:** Some of the advantages of NoSQL database are as follows [12]:

- High scalability.
- Distributed Computing.
- Lower cost.
- Schema flexibility, semi-structured data.
- No complicated Relationships.

B. ***DISADVANTAGE:*** NoSQL databases are not without their faults and limitations. Some common issue with these databases are as follows [12]:

- Lack of encryption support for data files.
- No standardization.
- Limited query capabilities (so far).
- Eventual consistent is not intuitive to program for.

XI. CONCLUSION

The aim of this paper is to give a thorough overview and introduction to the NoSQL database movement which appeared in the recent years to provide alternatives to the predominant relational database management systems. The SQL vs. NoSQL debate will continue. Facebook, 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. The demand for SQL will not go away anytime soon, nor will the reality of today's more distributed, virtualized, and commodity-based IT infrastructure. NoSQL is increasingly considered a viable alternative to relational databases, and should be considered particularly for interactive web and mobile applications.

REFERENCES

1. E. F. CODD," A Relational Model of Data for Large Shared Data Banks", Communications of the ACM, Volume 13/ Number 6/ June, 1970. Available at: http://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf.
2. Michal Valenta," ACID implementation in RDBMS", Available at: https://users.fit.cvut.cz/valenta/doku/lib/exe/fetch.php/bivs/dbs-2/rdbms-architecutre-hadnout.pdf.
3. George Coulouris, Jean Dollimore, Tim Kindberg ," DISTRIBUTED SYSTEMS Concepts and Design Fifth Edition",Available at: https://azmuri.files.wordpress.com/2013/09/george-coulouris-distributed-systems-concepts-and-design-5th-edition.pdf.
4. Insup Lee," Introduction to Distributed Systems", Department of Computer and Information Science University of Pennsylvania, CIS 505, Spring 2007, Available at : http://www.cis.upenn.edu/~lee/07cis505/Lec/lec-ch1-DistSys-v4.pdf.
5. Arto Salminen," Introduction to NoSQL", NoSQL Seminar 2012 @ TUT,Available at: http://www.cs.tut.fi/~tjm/seminars/nosql2012/NoSQL-Intro.pdf.
6. From Wikipedia ,Available at: https://en.wikipedia.org/wiki/NoSQL.
7. Christoforos Hadjigeorgiou," RDBMS vs NoSQL: Performance and Scaling Comparison", The University of Edinburgh ,Year of Presentation: 2013,Available at: https://static.ph.ed.ac.uk/dissertations/hpc-msc/2012-2013/RDBMS%20vs%20NoSQL%20-%20Performance%20and%20Scaling%20Comparison.pdf.
8. Seth Gilbert, Nancy A. Lynch," Perspectives on the CAP Theorem", Available at: https://pdfs.semanticscholar.org/0b0a/af71707a8247b35822f91a95319f1c97476c.pdf.
9. Salomé Simon," Brewer's CAP Theorem", CS341 Distributed Information Systems University of Basel, HS2012, Available at: http://informatik.unibas.ch/fileadmin/Lectures/HS2012/CS341/workshops/reportsAndSlides/ReportSalomeSimon.pdf.
10. Martin Kleppmann, " A Critique of the CAP Theorem", Available at: https://www.cl.cam.ac.uk/research/dtg/www/files/publications/public/mk428/cap-critique.pdf.
11. Chao Xie, Chunzhi Su, Manos Kapritsos, Yang Wang, Navid Yaghmazadeh, Lorenzo Alvisi, Prince Mahajan, "Salt: Combining ACID and BASE in a Distributed Database", Available at : file:///C:/Users/Dell/Downloads/Xie14Salt.pdf.
12. Cory Nance, Travis Losser, Reenu Iype, Gary Harmon," Nosql Vs Rdbms - Why There Is Room For Both", Proceedings of the Southern Association for Information Systems Conference, Savannah, GA, USA March 8th–9th, 2013,page No-111-116,Available at: http://sais.aisnet.org/2013/Nance.pdf.

BIOGRAPHY

Nusrat Jahan is an active researcher and Assistant Professor in the computer engineering Department at S.R.C.O.E, Pune has 1.5 years of teaching experience. She received her M.Tech degree (2013) in Computer Science and Engineering from Banasthali University, Rajasthan. And received B.Tech. degree in Computer Science and Engineering from R.N. Modi Engineering College, Kota, Rajasthan in 2010. Her subject of interests includes Data mining, Big data, Natural Language Processing and Information retrieval etc.

Kusum Munde is an active researcher and Assistant Professor in the computer engineering Department at S.R.C.O.E, Pune has 11 years of teaching experience. She received her M.E. in Computer Science and Engineering from B.A.M.U., Aurangabad, Maharashtra, India And received B.E. degree in Computer Science and Engineering from S.R.T.M.U., Nanded, Maharashtra, India. Her subject of interests includes Data mining, Big data, Natural Language Processing and Information retrieval etc.