CLOUD BASED DATABASES- A CHANGING TREND

Mrs. Anuradha S. Kanade[*]

Dr. Arpita Gopal[**]

Mr. Shantanu Kanade[***]

Abstract

Today is the era of web scale applications which are growing drastically. Internet is becoming the most essential and popular entity in everyday's life. There are more than 2 billion internet users. The size of data being generated regularly through facebook, Mobile apps, RFID is in zettabytes. To handle a large pool of data there is a high need for technology. The advance web browsers should follow the properties like user-friendliness, usability and availability. Changing need of applications and databases proved that the traditional RDBMS are not effective for distributed environment where as The requirements of cloud computing are such as high availability, high throughput, maximum and proven scalability, disaster recovery. NoSQL databases provide elasticity and scalability along with the capability to store huge data. This provides opportunity to work with the cloud computing systems. This makes the NoSQL system extremely popular. The paper discusses the effectiveness of NoSQL databases over the relational databases, being schema-free and following the BASE properties.

Index Terms— ***Zeetabytes, NoSQL, CRM, SAN, ACID, BASE, CURD***

[*] SIMCA Pune, India

[**] SIBAR Pune, India

[***] SIOM Pune, India

I. INTRODUCTION

The relational database technologies have dominated the IT industry since 1980s. Due to big and complex data, growing number of users these systems started showing their weaknesses. Web-scale or internet scale applications demanded for alternative for the relational database system. The companies like Google, Amazon and Web 2.0 invented the NoSQL database for handling their applications according to the changing need of the environment. Due to flexible schema, power to handle the large amount of complex data such as semi-structured data or unstructured data and simple database design the NoSQL database systems became very popular. Present paper discusses about this upcoming database technology and its effectiveness.

II. CHANGE IN DATABASE MANAGEMENT SYSTEM TREND

Internet is becoming the most essential and popular entity in everyday's life of a layman. There are more than 2 billion internet users. The age group between 14-19 years is the major internet users among the entire population. As the number of internet users is increasing, the use of web applications for communication, data sharing becomes extremely normal. Hence the web applications also have to cope with larger number of simultaneous users. Similarly the mobile users are also increasing with a great speed. There are more than 4.6 billion mobile users. The applications and services provided by the mobile devices are becoming more and more sophisticated. The mobile phone is now called a smart phone. The laptops are now very common mobile devices. It is seen that most of the items carry RFID tags these days. There are about 30 billion RFID tags. The capital market is growing very fast. The social area network sites such as facebook and twitter process near about 10 terabytes of data daily. Looking at the present scenario huge volume of data is being generated every day. To handle a large pool of data there is a high need for technology. The goal of such system is high availability and the control over the one's own data. It also meets the privacy standards expected from modern web applications, by the users. The advance web browsers should follow the properties like user-friendliness, usability and availability.

The current changes in the circumstances and raised expectations from applications have been pointed out in the above discussion. More people are using the ever growing systems on ever growing mobile devices. They demand high availability and usability. The evolution in the technology is ongoing that helps to satisfy these needs. The backbone of software industry is database management system. With the increase in the speed and capabilities of computer system, many general purpose database systems emerged in 1960s. The main focus was on the application programs to extract and assimilate large amount of business data. The calculations involved were relatively simple. In 1970, E. F. Codd presented the relational model for the large data. Data was separated into individual tables and related by keys. Oracles, Sybase, DB2, and Informix etc. are relational database systems. In late 1970s, the standard query language SQL was introduced in database. With the invention of microcomputers, individual users could create and manage their own database system. After the evolution of object oriented programming, object oriented database system was emerged in 1980s. In late 1980s, the client/server database system came into existence. Applications run on client attached with server over the LAN. With the evolution of World Wide Web all types of databases were made available to all people connected to internet.

Now in 21st century the NoSQL databases came into existence. They enable any type of data to be stored in the database without having fixed structure.

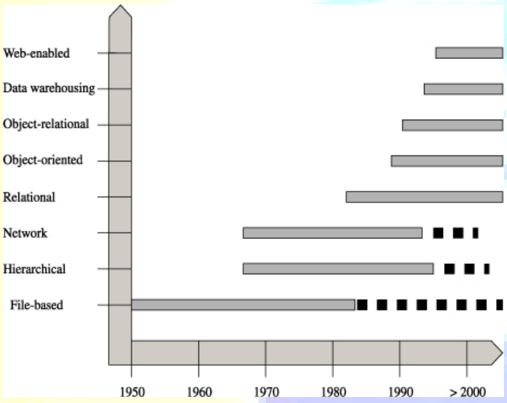


Figure 1. Changing Trend in DBMS

A. *Changing scenario in Interactive software*

There is huge difference in the trend in 1975 and today. The evolution in the technology has lead to the drastic change in the type of software used. Following are the main components of the interactive software.

1. Users

In 1975, there were about 2000 interactive software systems. Few organizations such as American Airlines System, branch automation system of Bank of America deployed and supported such software. But today due to social networking sites, mobile usages, E-commerce applications there are more than two billions users. The web applications can serve the users 24 hours a day, 365 days a year. The applications can grow from no user to millions of users.

2. Applications

In early days the interactive software was used to automate the complex business process and reduce the paper work. The systems like online reservation, payroll, stock maintenance sales management etc were developed. But today the interactive software systems are changing the nature of communication, shopping, advertising, entertainment and relationship management. Hence database system should also be flexible with the changing requirement.

3. Infrastructure

In 1970 the infrastructure was mainly centralized. The computing environment included the mainframe, minicomputers with shared CPU, disk, memory and so on. The computer networking was in its infancy. The memory was very expensive and scarce resource. Today this norm is changed to distributed environment. The servers and virtual machines are interconnected via high speed data networks.

	In 1975 Online Applications	**In 2015 Interactive Web Applications**
Users	Max. 2000 online users	Min. 2000 online users
	Static Users	Dynamic Users
Applications	Business Process Automation	Business Process Innovation
	Highly Structured Data	Structured, Semi-structured and Unstructured Data
Infrastructure	Beginning of Networking	Sophisticated Networking with very high speed

	Centralized Computing	Distributed Computing
	Limited and expensive Memory	Readily available and cheap memory

Table 1. Changing Scenario in Interactive Software

4. *Requirements of Cloud Computing*

Cloud computing is expected to reduce the cost and improve the flexibility along with agility. These benefits cannot be achieved without providing massive scalability at incremental cost. This enables the need for the fault-tolerant data stores and alternative to the fixed structured RDBMS. The applications like business intelligence, enterprise analytics, CRM, document processing, SAN, web 2.0 applications have varying needs for data, query and index types. The concepts like normalization, ACID properties of relational databases are found to be inadequate in distributed processing.

Dwight Merriman of 10 gen (the company which invented the MongoDB) stated two major requirements of data stores in cloud computing environment. [3]

1. High until almost ultimate scalability in horizontal direction
2. Low administration overhead

According to him the following classes of databases work very well in the cloud.

1. Data warehousing specific databases for batch data processing and map/reduce operations.
2. Databases containing a richer feature set than key/value stores fitting the RDBMS
3. Simple fast and scalable key/value-stores
4. Databases that contain richer key/value stores. These databases fill the gap between traditional databases and offers good performance and scalability such as document stores.

The cloud computing also has following common requirements.

i. Security- There should be world-class security provided at every level.
ii. Transparency- There should be accurate, transparent and real-time performance of service
iii. Multitenancy- It should follow multitenant architecture

iv. Horizontally scalable- The system should be highly scalable and support millions of users

v. High performance – The performance of system should be high and delivery should be consistent

vi. Disaster recovery –The data should be highly protected from failure at any time

vii. High availability- High availability of infrastructure and software should be present.

III. NoSQL Databases

Traditional databases handled more predictable and structured data. Relational databases may require vertical and sometimes horizontal expansion of servers to expand as data or processing requirements grow. More cloud friendly approach to employ NoSQL database provide an alternative to this. A NoSQL database is the type of database that can handle the structured, semi-structured and unstructured data. NoSQL databases are commonly referred to "Not-Only-SQL" as they provide SQL support as well.

A. *Characteristics of NoSQL Databases*

The popularity of NoSQL databases is due to their beautiful features. [3]

1. No schema required.

   Data can be inserted in NoSQL database without defining the database schema. Also the format of the data can be changed at any time without disturbing the application. It provides tremendous flexibility on business.

2. Auto-shading

   This is sometimes called as elasticity. A NoSQL database automatically spreads the data across the multiple servers without requiring applications to participate. The servers can be added or removed from the data layer without application downtime. The data is spread automatically over the servers. Many NoSQL databases support the cross data centers, data replication, storing multiple copies of data across the cluster. All this ensures the high availability and support s disaster recovery.

3. Distributed query support

   In contrast with RDMBS system, NoSQL database system retain the full query expressive power even after distributing across hundreds or thousands of servers.

4. Integrated caching

   The NoSQL database technologies cache the data in system memory to reduce latency and increase sustained data throughput. This is transparent to the application developer and operations team while in RDBMS the caching tier is separate developed on separate servers and managed by operations team.

B. ACID Vs BASE

The relational databases provide tight structure and very strict consistency. They also provide a very large feature set and follow ACID (Availability, Consistency, Isolation and Durability) property. The acronym for ACID is as follows.

- Atomic: Everything in a transaction succeeds or the entire transaction is rolled back.
- Consistent: A transaction cannot leave the database in an inconsistent state.
- Isolated: Transactions cannot interfere with each other.
- Durable: Completed transactions persist, even when servers restart etc.

But in data warehouses and business intelligent applications all this may not be necessary. The age of internet along with the social area networks, blogs, mobile applications, wikis etc has created the need for processing, analysing and delivering the constantly growing enormous data. The organizations, companies and individuals who offer these services and applications have to determine their requirements regarding performance, availability, consistency and durability.

NoSQL databases in contrast with RDBMS follow the BASE (Basically Available, Soft-State and Eventually Consistent). According to Ippolito BASE properties can be summerized in following way.

An application works normally all the time i.e. basically available. It does not have to be consistent all the time but in some known state i.e. eventually consistent.The decision criteria to select whether to choose ACID or BASE properties is suggested by Brewer. According to him if a system or parts of a system have to be a consistent and partition-tolerant, ACID properties are required and if availability and partition-tolerance are important then BASE properties can be followed. For growing number of applications and use-cases the availability and partition are more important than strict consistency.

C. CAP theorem

Eric Brewer in the symposium at ACM in 2000 came up with the CAP-theorem [5] which is now widely accepted by large web companies such as Amazon as well as NoSQL community. There are three properties of a system viz. consistency (all copies have same value), availability (system can run even if parts have failed) and Partitions (network can break into two or more parts, each with active systems that cannot influence other parts). According to CAP theorem, the system can have at most two of these three properties for any shared-data system. To scale out, system should have partition. That leaves either consistency or availability to choose from. In almost all cases, availability is chosen over consistency.

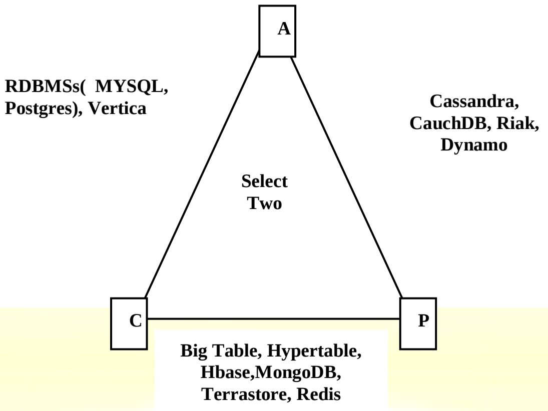

A
RDBMSs(MYSQL, Postgres), Vertica
Cassandra, CauchDB, Riak, Dynamo
Select Two
C
P
Big Table, Hypertable, Hbase,MongoDB, Terrastore, Redis

Figure 2. CAP and associated NoSQL databases

D. *NoSQL architecture*

The NoSQL architecture has four components .

1. Modelling Language -The structure of database and schema is described by the modelling language.
2. Database Structure - Each database uses its own data structures and stores the data using permanent storage device.
3. Database Query Language - The operations like create, update, read and delete can be performed on database.
4. Transactions- The operations like Create, Update, Read and Delete (CURD) can be applied on the database.

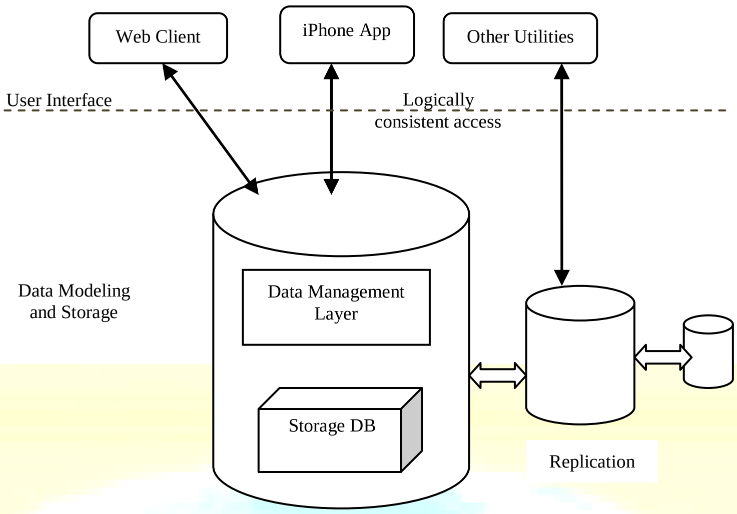


Figure 3. NoSQL Architecture

E. *Categories of NoSQL databases*

The taxonomy of NoSQL databases is discussed as follows.

1. Key-value stores

Key-value stores allow the application developer to store a schema-free data. This type of database is the backbone of all other NoSQL databases. This data is usually a string and represented by the key and its value pair. This helps to define the data with flexibility by avoiding fixed data model. The key of the item is unique in nature. Tokyo Cabinet, Redis, Cassandra are the key-value databases. The storage mechanism for them is easy to understand and complex SQL queries are not required.

Major operations performed are,

get(key), returning a list of objects and a context

put(key, context, object), with no return value

To ensure the availability and durability of the system when machine is crashed the replication is used. Replication in this case is done as shown on the diagram. The nodes are connected in circular fashion to each other and the keys between A and B are stored by the other nodes B, C and D.

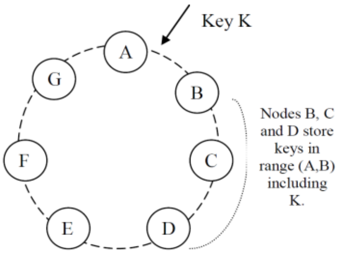
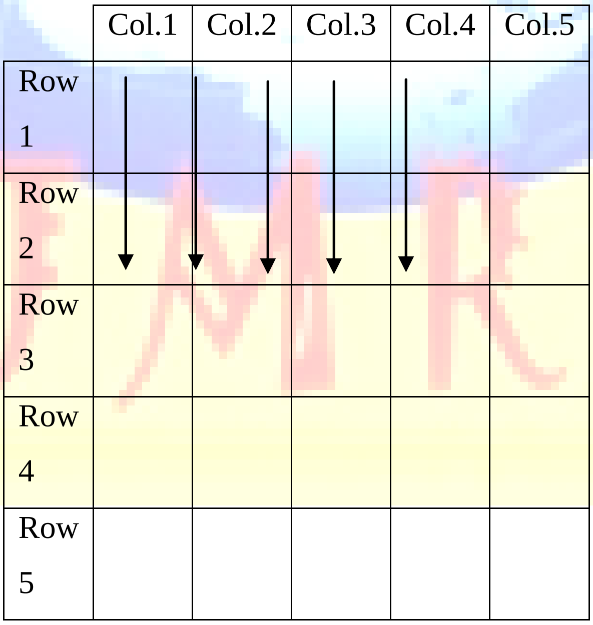


Figure 4. Replication

2. Column NoSQL databases

A column NosQL databases store data in a columnar manner and each attribute is stored in a separate table and successive values of that attribute are stored consecutively. Columns are essentially keys that can be used to search the related values in rows. Null values do not exist in the table. Any number of columns can be added any time. This gives advantage for data warehouses and analytical applications. They provide aggregation functions with great speed and handle vast volume of data. Unused columns do not occupy the storage. Hence they use smaller disk space. SybaseIQ, Vertica, C-Store, BigTable, Cassandra are the column databases.



	Col.1	Col.2	Col.3	Col.4	Col.5
Row 1					
Row 2					
Row 3					
Row 4					
Row 5					

Figure 5. Structure of Column-Oriented Database

3. Document-based

Document databases are considered by the many as it is the next step from simple key-value stores. It allows representing more meaningful data structures. They allow encapsulating the key-value pairs in the documents. There is no fixed schema for the documents. So there are no issues regarding schema migration. Documents consist of named fields that have key/value pair. The key value is unique and its value may be string, number, Boolean, date, ordered list or associative map. These databases provide powerful and dynamic queries, binary storage, scalability, great documentation and multi-language support. Due to their effectiveness they are becoming very popular in Industry. CauchDB, MongoDB are the examples of document based databases. The example of document is as below.

{

FirstName: “Radha”, Address: “Pune”, Children:({Name:“Soham”, Age:5}, {Name: “Soumya”, Age: 1})

}

4. Graph database

Graph is very powerful tool for representing understanding objects and their relationships in various application domains. Today these databases have become more in use and the volume of graph data increases the rapidly. But the performance of query processing is still adequate due to complexity of processing graph data. They provide the set of theorems for deriving equivalences and thus provide foundation for the graph traversal engine optimizes. neo4j, InfoGrid are Graph based databases. The social area network, telephone cabling, circuit diagrams etc can be shown using graph structure very effectively.

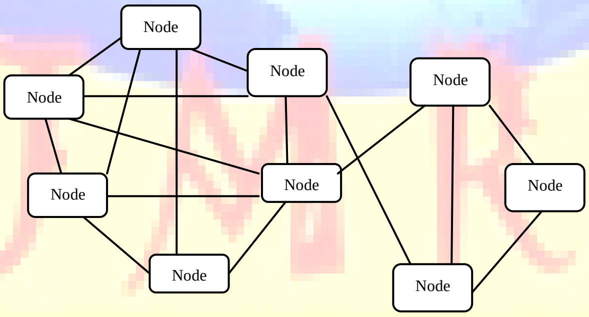


Figure 6. Social Area Network (SAN)

5. Data structure store

This type of database enables to store the data structures as the value itself. Redis is an open source advanced key/value store referred to as a data structure server. The keys can contain strings, hashes, lists, sets and sorted sets.

Mainly open source databases are widely used as they have small upfront software costs and are suitable for large scale distribution on commodity hardware.

F. Benefits of NoSQL databases over RDBMS

1. The NoSQL databases avoid such unnecessary complexity. – Relational databases provide large feature set and very strict consistency. They also follow the ACID (Availability, Consistency, Isolation and Durability) property. These things may be more than necessary to develop particular applications.
2. High throughput –NoSQL databases have a simple API, serve huge amounts of data and provide significantly high throughput than RDBMS.
3. Horizontal Scalability and Running on Commodity Hardware – NoSQL databases are designed to scale better in horizontal direction and also they do not rely on the highly available hardware. Some NoSQL databases provide auto sharding.
4. No need of expensive object-relational mapping – Many NoSQL databases use more simple or similar objects used in object oriented programming language. They avoid the use of expensive object-relational mapping.
5. Ease of setting up database clusters – The NoSQL databases allow to set up the clusters very easily and the cost of setting up clusters is also very low as compared to RDBMS.
6. No more One-Size-Fits-All concept – Relational database systems believe in rigid structure of database. The data is forced to fit into that structure. But the NoSQL database gives flexibility in storing the data as some of these databases are schema free.

G. NoSQL applications

From the simplicity of the columnar approach accrue many benefits, especially for those seeking a high-performance environment to meet the growing needs of extremely large analytic databases. These key factors are seamlessly engineered into a column-oriented database, which enable reasonably-priced, benchmark-busting performance to meet an organization’s business intelligence needs.

H. Challenges with NoSQL databases

Though NoSQL systems are just in developing stage, they have become very much popular. But presently they are facing following challenges.

1. Maturity – The RDBMS systems are stable and mature enough as they are around since long time. At the other side the NoSQL databases are still emerging and many features are yet to be implemented.
2. Support- Any enterprise wants to get timely support at the time of system failure. All RDBMS vendors provide high level enterprise support. But mostly all NoSQL systems are open source systems. There is very less support resources from Oracle, Microsoft or IBM.
3. Administration- The NoSQL systems mainly are designed to provide no admin solution, but the today users require skill and effort to maintain the system.
4. Modeling- The data model may suffer from duplication of data objects (non-normalized model). This can happen due to the different object model used by different developers and their mapping to the persistency model.

IV. CONCLUSIONS

From the discussion in this paper it is clear that the volume of data is drastically increasing due to web applications such as social area networks like facebook, business intelligent applications, mobile apps etc. The size of data being generated today is in zettabytes which a very large. To handle this bulky data the traditional relations databases are not suitable. The alternatives for this problem are available through the new trend NoSQL databases. They provide a wide variety of databases and also benefits such as no complexity, horizontal scalability, schema free structure and cloud computing features. It is also true at the same time that the relational databases will not disappear from the picture as they have their own application areas in business processing applications, but the today's era proves that there are more effective alternatives available for distributed environment and cloud computing through the range of NoSQL databases. NoSQL databases fulfill Cloud computing requirements such as horizontal scalability, high throughput, handling high volume of data, flexibility in data storage, fast and availability. According to CAP theorem it is clear that only two properties among Consistency, Availability and Partition are followed by shared data systems at a time. The web applications such as Amazon have already accepted this fact. The NoSQL databases provide different flavours of databases that gives tremendous flexibility in using any type of data viz. Graph-based, Key-Value based, Document-based etc. This is the great benefit achieved by the users.

V. FUTURE SCOPE

As the NoSQL databases are widely becoming popular, there are lot of opportunities in research. Any special NoSQL database can be considered for studying the performance based on query optimization, memory consumption and scalability. The comparative study of various NoSQL databases on the basis of performance can be done.

ACKNOWLEDGMENT

The authors wish to thank the laboratory provided by SIBAR which provided us the working platform. We are grateful to Dr. Apoorva Palkar, Director-SIMCA & Dr. S. D. Mundhe, Director-MCA (SIMCA) for their suggestions and support. Also the authors are thankful to their friends who helped directly or indirectly.

REFERENCES

[1] An Oracle White Paper, 2011, “Oracle NoSQL Database”, pp. 1- 12

[2] Couchbase, “NoSQL Database Technology”, 2011, White Paper-Post-relational data management for interactive software systems, pp. 1-14

[3] Christof Strauch,2011, NoSQL Databases, A Project Report Submitted to Stuttgart Media University

[4] Denise Miura, “A practical guide to noSQL”, a blog posted on March 17, 2011 at http://blogs.marklogic.com/2011/03/17/a-practical-guide-to-nosql/

[5] Dr. Eric A. Brewer and Inktomi ,” Towards Robust. Distributed Systems”, UC Berkeley, PODC Keynote, July 19, 2000, <www.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf>

[6] Julian Browne, “Brewer's CAP Theorem”, posted by Julian Browne, January 11, 2009. http://www.julianbrowne.com/article/viewer/brewers-cap-theorem

[7] Kristina Chodorow and Michael Dirolf , 2010, “MongoDB The Definitive Guide”, O’Reilly Media Inc., Sebastopool, CA, USA

[8] Kyle Banker, 2010, "MongoDB Schema Design", MongoNYC, pp 1-66

[9] Lenk Alexander, Klems Marcus et. al., 2010, “What’s Inside the Cloud? An Architectural Map of the Cloud Landscape”, IEEE, Cloud’09 -ICSE’09 Workshop, pp. 23 -31

[10] “NoSQL -- Your Ultimate Guide to the Non -Relational Universe!” < http://nosql-database.org/links.html > accessed on 6-3-13

[11] “NoSQL (RDBMS)” , <http://en.wikipedia.org/wiki/NoSQL> accessed on 6-3-13

[12] Roberto V. Zicari, 2010, “On NoSQL technologies - Part III: document stores, nosql databases, ODBMSs”, p. 1-8

[13] Safari Books Online, “Historical Perspective of Database Systems”, article accessed on March 16, 2013, posted on <http://my.safaribooksonline.com/book/databases/9788177585674/introduction-to-database-systems/ch01lev1sec9>

[14] Vatika Sharma, Meenu Dave, 2012, “SQL and NoSQL Databases”, International Journal of Advanced Research in Computer Science and Software Engineering, pp. 20-27