International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal | www.ijtsrd.com
ISSN No: 2456 - 6470 | Volume - 2 | Issue – 6 | Sep – Oct 2018

NOSQL Database Engines for Big Data Management

Mrs. Yasmeen
Assistant Professor, Department of Computer Science and Engineering,
SSM College of Engineering and Technology, Baramulla, Jammu and Kashmir, India

ABSTRACT

We are living in the digital world and last two decades have seen significant expansion in the information on internet technology. In present digital world the IOT is most popular term means computers, mobile phones and physical devices like sensors are connected to internet. With the rapid outreach of internet it is very important to focus on technological advancements for managing huge amount of data with easy access.

Keywords: *Sensor, IOT, NOSQL*

I. INTRODUCTION

A database is a collection of data items that provides an organizational structure for information storage. Database also provides a mechanism for querying, creating, modifying and deleting data. A list can also be used to store data but in a list, redundancy is a major issue. A database can store relationships and data that are more complicated than a simple list with lesser or no redundancy. A relational database stores data in tables. Normally a table is based on one information theme. For example, an employee list can be divided into manager table, intern table, and junior staff table. A table is a two dimensional grid of data that contains columns and rows. The convention in relational database world is that columns represent different attributes of an entity and each row represents the instance of the entity.

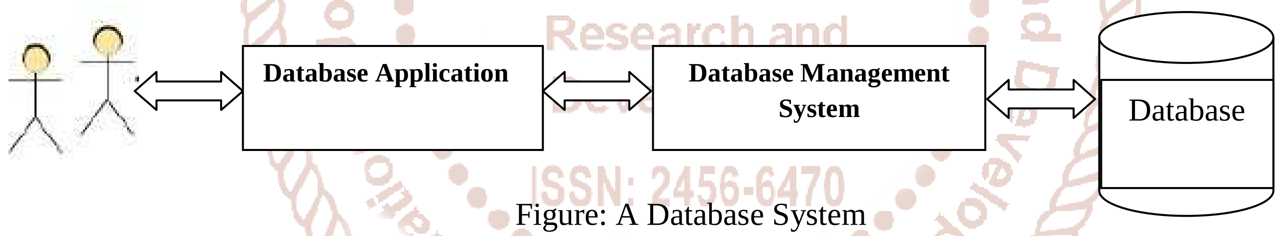

Figure: A Database System

Conceptually, database is a component of database system. Besides database, database system consists of database users, database applications and Database Management Systems (DBMS). Database users need not to be always human. It is possible, for example, for other software programs to be users of the database. Users interact with database application and application further depends on the DBMS to extract and store data in the database. The DBMS acts as a gatekeeper. All the information owing in or out of database must pass through the DBMS. It is a critical mechanism for maintaining quality of data and database. Users and database applications are not allowed directly to interact with database. A Database Management System is an intermediary between database applications and database. The DBMS creates and manages the database. DBMS can be categorized based on its data model. Relational Database Management Systems (RDBMS) [50] use relational data model given by Dr. E.F. Codd. RDBMS maintain data in tables and relationships which are created among data and tables. Database is divided into tables and they are connected through a "key field". RDBMS is the most famous and used database model.

Over last four decades, RDBMS remain a key technology to store structured data. But with growing size of data, companies do need modern technologies to maintain and process data. RDBMS are not that good for large data volumes with varying datatypes. They also have scalability problem and often result into failure while performing distributed sharding. Oracle Real Application Clusters (RAC) is a relational database cluster that provides high availability, reliability and performance. Also, MySQL cluster is another example where relational databases scale on large cluster. RDBMS satisfy ACID (Atomicity, Consistency, Isolation and Durability) properties defined by Jim Gray in the late 1970s. Consistency is bottleneck for scalability of relational databases. RDBMS follow strict data model and can not violate ACID properties. That is why NoSQL data stores were developed to address the challenges of traditional databases.

II. NOSQL DATABASES

In a computing system, huge amount of data 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 named "A relational model of data for large shared data banks" which made data modelling and application programming much easier. Beyond the benefits, the relational model is also well-suited for client-server programming and today it is a predominant technology for storing structured data in web and business applications .Applications also grow with time and pose challenging demands for the data management. As stated by Jim Gray, the most challenging part is to understand the data and find patterns, trends, anomalies and extract the relevant information. With the advent of Web 2.0 applications, the data stores needed to scale to OLTP/OLAP-style application loads where millions of users read and update the information, in contrast to the traditional data stores. These data stores provide good horizontal scalability for the simple read/write operations distributed over many servers. The relational database systems have little capability to horizontally scale to these levels. So, this paved the way to seek alternative solutions for scenarios where relational database systems proved to be not the right choice. NOSQL database are growing fast and are best choice for handling the big world problem popularly known as Big Data and supporting Business Intelligence in organisations. Today we need rich mobile apps highly available, very responsive and not affected by network availability. To develop such modern mobile apps NOSQL (mobile databases) are the best solution for modern mobile app development. NOSQL use wide variety of different DB technologies that came into existence in response to the demands present in building modern applications. Mobile world is one of the most dynamic areas of Information Technology today. Smart phones and tablets have created a huge market for mobile applications. Consequently there is an increasing demand for mobile application developers. Almost all of the mobile applications require a persistent data layer, including options for queries. So the interest of database professionals, academics and researchers for mobile technologies is increasing. NOSQL approach is a strong competitor to the relational model because it supports high scalability. The famous CAP theorem describes that not any database system supports all the three attributes but only two of three is possible. Relational databases support only consistency and partition tolerance properties and the NOSQL databases support the last two means availability and partition tolerance for high availability and partitioning of data.

Types:

There are three main types of NoSQL data stores: Key-Value Data stores, Extensible Record Data stores and Document Data stores.

- In the Key-Value data stores, the values are indexed with keys and its data model follows a famous memcached distributed in-memory cache. Examples include Project Voldemort, Riak and Redis.
- Document data stores retrieve, manage and store semi structured data. They provide support for multiple forms of documents (object). The values are stored in documents as lists or nested documents. Few examples are MongoDB, SimpleDB, and CouchDB.
- Extensible Record data stores are motivated from Google's Big Table. It has flexible data model with rows and columns. Rows and columns can split over multiple nodes. HBase, Hyper Table, and PNUTS are its few examples.

III. DOCUMENT DATA STORES

Document oriented data stores are design to store, retrieve and manage semi structured data. They support multiple types of documents (objects) per data stores. "Documents" save values as nested documents or lists. These documents are of any type ranging from PDF, Word document, XML, HTML, etc. SimpleDB, CouchDB, and MongoDB are few examples of Document Oriented datastore.

MongoDB: MongoDB is an open source, document oriented datastore that is written in C++. It is developed by 10gen (Now MongoDB Inc.) for a wide variety of real time applications. It also provides full index support for collection of documents. MongoDB has a well structured document query mechanism. Next few subsections discuss different aspects of MongoDB design.

NoSQL data stores are quite handy to deal with much large velocity and volume of data. MongoDB is a scalable and high performance NoSQL datastore. It is an agile datastore that allows schemas to change quickly as applications evolve. It is provided with the rich querying capabilities. MongoDB is a real time datastore usually used for online data but also find applicability in wide variety of industries. The MongoDB package has different tools. Depending on operating system, the MongoDB package has different package components. Mongod, mongo and mongos are the core processes of MongoDB package. Mongod is responsible for database whereas mongos is for sharded cluster. Mongo is the interactive shell or the client. For the Windows environment, there are specific services like mongod.exe and mongos.exe. Different tools for data and binary import/export functionalities are the part of MongoDB package. Different MongoDB tools are depicted in the following figure. Mongod is the primary daemon process for the MongoDB system. It takes care of data requests, manages data format and executes background management operations. Datastore is a physical container for collections. Each datastore gets its own set of files on the file system. A single MongoDB server typically has multiple data stores. Unlike Extensible Record store datastore like HBase, MongoDB does not require a file system to run. Collection is a group of MongoDB documenet and is equivalent to a RDBMS table. Collections do not enforce any type of schema. Documents within a collection can have different fields. Normally, all documents in a collection are of similar or related purpose. Inside one collection, user can have "n" number of documents. Document has a JavaScript Object Notation (JSON) structure that stores a set of key/value pairs. Normally, all documents in a collection are of similar purpose.

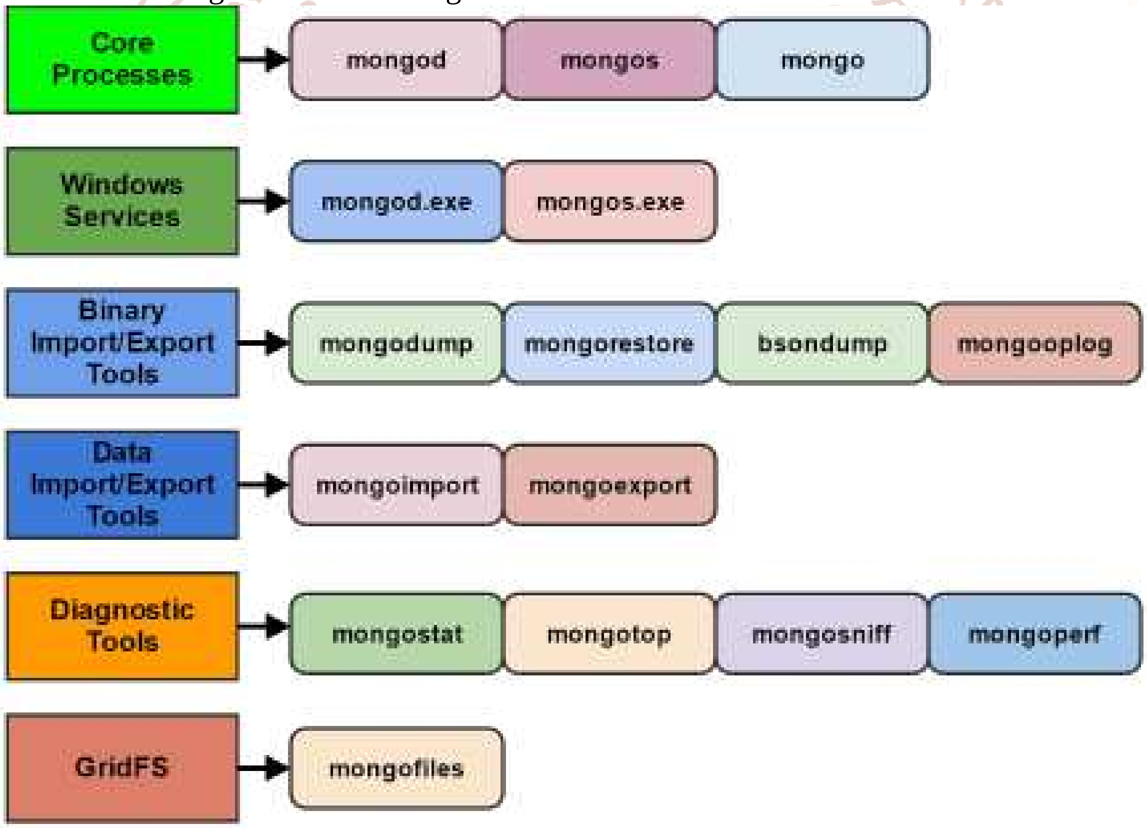

Figure: MongoDB Package Components

IV. EXTENSIBLE RECORD DATA STORES

Google's Big Table is the motivation for extensible record database engines. It has a flexible data model with rows and columns which can be extended any time. Apache HBase, Apache Accumulo, and HyperTable are few of the famous Extensible Record stores. Extensible record stores are scalable and both rows and columns can split over multiple nodes. Extensible Record stores are often term as Column Oriented data stores.

HBase: HBase is a Column Oriented data store that runs on top of HDFS. HBase is an open source Apache project which can be summarized as distributed, fault tolerant scalable data store. It is good in managing sparse data sets. Unlike a relational database management system (RDBMS), it does not support structured query language like SQL. In fact, HBase is not at all a relational database. HBase is written in Java much like a typical Hadoop application but it does not use MapReduce. HBase applications can also be written using AVRO, REST and THRIFT API. A HBase system is made up of set of tables. These tables are stored in HDFS. Each table contains rows and columns much like a traditional database. Each table has a column defines as it primary key and all calls to access the table must use the primary key. HBase architecture has three layers namely: the client layer, the server layer and the storage layer. The client layer provides an interface to the user. It has client library which is used to communicate with the HBase installation. The storage layer has a coordination system and a file system. HDFS is the most commonly used file system for HBase. Apache ZooKeeper is used as the coordination service for HBase. A master server and the region servers are two component of server layer. The following figure describes the architecture overview of HBase.

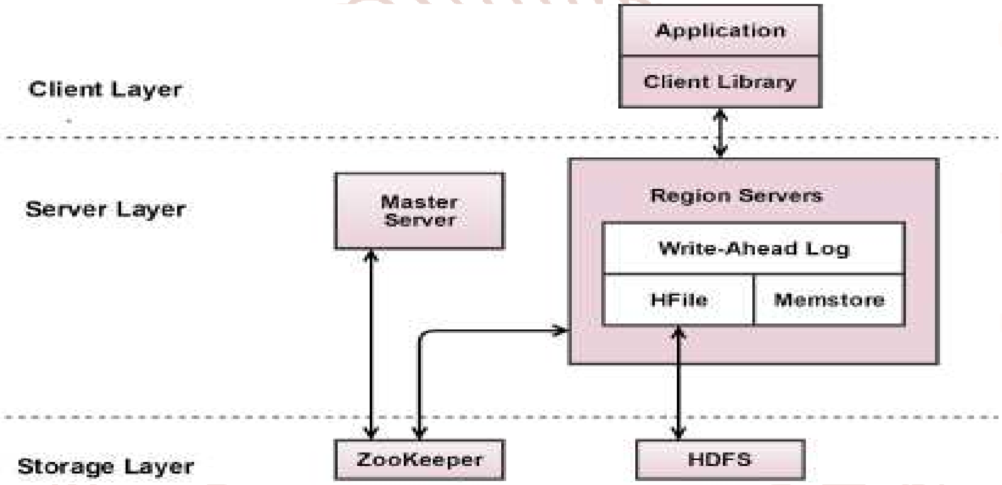

Figure: HBase Architectur

V. KEY-VALUE DATA STORES

Key-value: Key-value data stores are primarily a big hash tables with unique primary key and a pointer to a particular data item. Its data model has identical design to the memcached in memory cache. The keys can be primitive types or objects and values are accessed only by keys. These data stores provide support for much functionality like replication, partition, locking, versioning, transactions and/or other features. They are extremely useful in building specialized application with super fast query capabilities. Cassandra, Redis, Riak, Scalaris, and Project Voldemort are few examples of key-value data stores.

Cassandra: Cassandra is a distributed, highly scalable and fault tolerant NoSQL datastore. It is a structured store with decentralized architecture. It was developed by Facebook Inc. and its first release came out in 2008. The main aim to develop Cassandra is to meet storage requirements of the Index Search Problem. For this purpose, Facebook needed a datastore with very high write throughput. Apache Cassandra is an open source project under the Apache license 2.0.

In traditional databases that can be deployed over multiple nodes and even in data stores like HBase, Google's Bigtable etc, master slave relationship exist between the nodes. The master is authority for distributing and managing data. Slaves on other hand synchronize their data to the master. All writes pass via master and it is the single point of failure. The architectures that have master/slave setup sometime have adverse effect if master node fails.

By contrast, Cassandra was designed with the understanding that failures can and do occur. It has a peer-to-peer distribution model. The data is divided among all nodes in the cluster. All nodes are structurally identical. Therefore, there is no master node. Equality among nodes due to peer-to-peer network improves general datastore ability. It also makes scaling up and scaling down much easier because a new node will not be treated differently. The following figure describes the Cassandra Read Repair.

VI. CONCLUSION

Big Data is a very popular term today represented by 3V i.e volume, variety and velocity of data. The research paper focus on new breed of databases commonly known as NOSQL Databases and how they are beneficial for managing huge volumes of data.

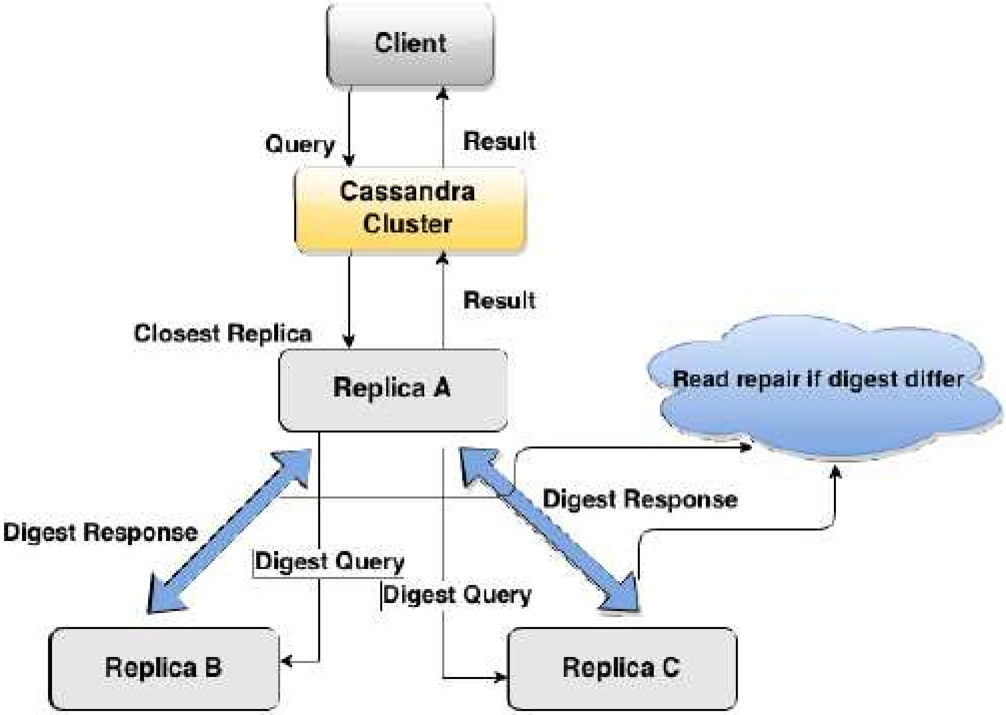

Figure: Read Repair of Cassandra

REFRENCES:

1. A B Moniruzzaman and Syed AkhtarHossain, "**NOSQL Database: New Era of Databases for Big Data Analytics- Classification, Comparison and Characteristics",** International Journal of Database Theory and application.
2. Aaron Schram and Kenneth M. Anderson, "**MySQL to NOSQL: Data Modelling challenges In Supporting Scalability**".
3. AmeyaNayak, Anil Poriya, and DikshayPoojary, "**Types of NOSQL Databases and its comparison with the Relational Databases",** International Journal of Applied Information Systems, Volume 5 No. 4, March 2013, www.ijais.org.
4. AnkitaBhatewara and KalyaniWaghmare, "**Improving Network Scalability using NoSql Database",** International Journal of Advanced Computer Research, Volume-2, Number-4, Issue-6, December-2012.
5. Chad DeLoatch and Scott Blindt, "**NOSQL Databases: Scalable Cloud and Enterprise Solutions",** August 2, 2012.
6. Chieh Ming Wu, Yin Fu Huang, John Lee, "**Comparisons between MongoDB and MS-SQL Databases on the TWC Website**", American Journal of Software Engineering and Applications, Volume 4, No 3, April 2015.
7. Clarence J M Tauro, Aravindh S and Shreeharsha A.B, "**Comparative Study of the New Generation, Agile, Scalable, High Performance NOSQL Databases",** International Journal of Computer Applications, volume 48-No. 20, June 2012.
8. David Taniar, "**High Performance Database Processing**", 26thIEEE International Conference on Advanced Information Networking and Applications, 2012.
9. DrK.Chitra and B.Jeevarani, "**Study on Basically Available, Scalable, and Eventually Consistent NOSQL Databases",** Volume3, Issue7, July 2013, www.ijarcsse.com.
10. Felix Gessert, Wolfram Wingerath, Steffen Friedrich, Norbert Ritter, "**NoSQL database systems: a survey and decision guidance**", Springer, November 2016.
11. GuoYubin, Zhang Liankuan, Lin Fengren, Li Ximing, "**A Solution for Privacy-Preserving Data Manipulation and Query on NOSQL**

database", Journal of Computers, Vol 8, No. 6, June 2013.

12. Hanen Abbes, FaiezGargouri, **"Big Data Integration: a MongoDB Database and Modular Ontologies based Approach",** Elsevier, September 2016.
13. InduArora and andDrAnu Gupta, "**Cloud Databases: A Paradigm Shift in Databases",** International Journal of Computer Science Issues, Vol 9, Issue 4, No. 3, July 2012, www.IJCSI.com
14. IoannisKonstantinou, Evangelos Angelou, Christina Boumpouka, DimitriosTsoumakos, NectariosKoziris, "**On the Elasticity of NOSQL Databases over Cloud Management Platforms (extended version)**", Computing Systems Laboratory, School of Electrical and Computer Engineering National Technical University of Athens.
15. Joao Ricardo Lourenco, Bruno Cabral, Paulo Carreiro, Marco Vieira, Jorge Bernardino, "**Choosing the right NoSQL database for thejob: a quality attribute evaluation"** Journal of Big Data, Springer, 2015.
16. Katarina Grolinger, Wilson A Higashino1, AbhinavTiwari,Miriam AM Capretz, "**Data management in cloud environments: NoSQLand NewSQL data stores**"Journal of Cloud Computing, Springer, 2013.
17. Leonardo Rocha, Fernando Vale, Elder Cirilo, Darlinton Barbosa, FernandoMourao, "**A Framework for Migrating Relational Datasets to NoSQL**", Elsevier, Volume 51, 2015.
18. Liana Stanescu, Marius Brezovan, and Dumitru Dan Burdescu, "**An algorithm for mapping the relational databases toMongodb – a case study",**International Journal of Computer Science and Applications, January 2017,https://www.researchgate.net/publication/318599517.
19. Lior Okman, Nurit Gal-Oz, YaronGonen, Jenny Abramov, "**Security Issues in NoSQL Databases**", International Joint Conference of IEEE TrustCom, 2011.
20. Marin FOTACHE and Dragos COGEAN, "**NOSQL and SQL Databases for the Mobile Applications. Case study: MongoDBVsPostgreSQL",** Volume 17, No 2/2013.
21. Nadeem Qaisar Mehmood, Rosario Culmone, Leonardo Mostarda, "**Modeling temporal aspects of sensordata for MongoDBNoSQL database**", Journal of Big Data, Springer, 2017.
22. Naseer Ganiee, "**New Database Constraints and Modern Applications",** IJLTEMAS, Volume III, Issue II, February 2014.
23. Naseer Ganiee, "**NOSQL: The Big Data Solution",** International Journal of Advancement in Engineering Technology, Management and Applied Sciences, Volume 1, Issue 2, July 2014.