**COMP-440** **TERM PAPER** HARISH PINDI

**A – no comment, all good.**

**COMPARISON OF SQL RELATIONAL DATABASES WITH HADOOP**

**1. Abstract:**

Since we live in the era where information is the ultimate power, it is a big challenge to store huge volumes of data electronically.   As there is a vast growth in data over by years, it is so hard to store this data and retrieve them back. In order to overcome such issues, there are several data storage technologies that are introduced to the market. Among them SQL relational databases and Hadoop are the prominent ones. So in this paper I will introduce about these data storage methods and make a comparison between these methods to find the most appropriate one to be used.

**2. Summary**

**2.1 Introduction of Big data:**

The epoch of data is everywhere in our day to day life. A market research analyst like IDC [2] (International Data Corporation) have estimated the size of the digital universe in terms of 4.4 zettabytes [3] in 2013 and forecasts the growth of the data to be 44 zettabytes in 2020 [1]. For example, like in social media sites, digital pictures, videos, web search engines, Institutional data records, etc. Additionally, the amount of information that is engendered by the machines as a part of the internet of things is still more outstanding than the record of data that is generated by the people [1]. All these vast amounts of data are so termed as Big data.

Big data also refer to as Data Intensive Technologies, that are becoming a new technology trend in science, business and industry. It often categorizes the data by means of 5V properties [4], that includes Volume, Variety, Velocity, Value and veracity. Where Volume defines the size of the data. While Velocity tells about the gait at which information is generated, Variety and Variability tell us about the complexity and social system of information and different ways of understanding it. IBM [5] conducted a survey on Big data and analytics and states that every day we create 2.5 quintillion bytes of information. This implies that 90 percent of the world's data created today is equivalent to the sum of the data generated for the last two years.

Big data not only defines about data, but also includes techniques, software, architecture, tools and technologies to deal with the data. After collecting these data from different sources, it helps in making smart decisions for the upcoming future product development, helps to know about the root cause of product failures, cost reduction, calculating the risk factors to avoid in the near future.

**2.2 Data Storage Methods:**

There are two kinds of data storage methods: SQL relational databases and Apache Hadoop that is well known and widely used in the present modern world.

The relational database is defined by Codd in 1970, as a digital database where the data is structured based on the relational model of information. It is required to accomplish a standard format used in the digital preservation of the relational databases data and structure [6].

Hadoop is an open-source framework built up by Doug Cutting in the year 2006 and is now managed by Apache Software Foundation. It is written in Java for distributed memory and distributed processing of very large information sets on computer clusters built from commodity hardware [7].

**3. Argument:**

Relational databases and Hadoop are two different types of storage methods. Each of them has its own unique design and architecture. Also, each of them has unique properties while dealing and handling the data, which can be further explained in this paper.

**4. Comparison of RDBMS and Hadoop in terms of ACID and CAP theory:**

**4.1ACID [8]**  
 A-Atomic, C-Consistency, I-Isolation, D-durability

**Atomicity:** Either all or none of the updates are executed or committed during in a transaction.

**Consistency:** If the database is consistent during the beginning and also at the end of the transaction, then the database is in a consistent state.

**Isolation:** Each of the updates of a transaction cannot be determined by the other concurrent transactions until the respective transaction is either committed or executed.

**Durability**: whenever the transaction is committed or executed the updates are never lost.

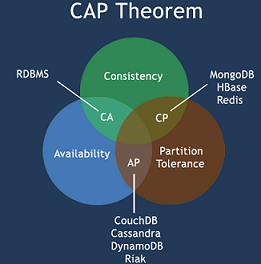
Relational databases generally follow these ACID properties for a safe sharing of potentially accurate data when the data is simple enough, but in the case of big data where the data is huge and unstructured, the ACID theory fails to implement all of its properties. So it is highly difficult for RDBMS to store the unstructured data and find a relationship between them since it has predefined schema for its tables. Also Hadoop fails to follow the ACID concepts when dealing with the big data. However, NoSQL has a dynamic schema for the unstructured data which is more compatible when dealing with big data.

**4.2 CAP Theorem [9]:**

**Consistency:** All of the nodes should see the same data at the same time. These imply data should be up to date and be consistent.

**Availability:** for every request there should be a response, whether it succeeded or failed. Even if there are node failures the survivors should continue to operate on.

**Partition tolerance**: Despite of the network partition the system continues to operate the process.



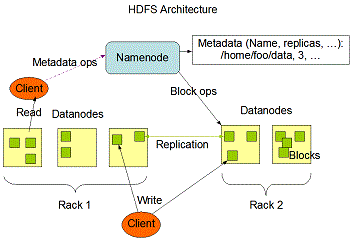
**Fig 4.2.1 CAP Theorem**

According to Brewer’s theorem [10] a distributed system can satisfy only two out three requirements at the same time. Cap theorem describes about the future trade-offs. A future of the CAP theorem always helps in making smart decisions. RDBMS is not partitioned tolerant and always meets the consistency and availability (CA). There are different use cases in which consistency is said to be ok. For instance, in a travel booking site Availability and partition tolerance are much important. So Cassandra, Couch dB and Dynamo dB always fall in this (AP) category. Whereas MongoDB, HBase, Redis falls into Consistency Partition tolerance (CP) category

**4.3 Architecture of Relational databases and Hadoop**

RDBMS architecture [11] is divided into three levels. The Internal level (Storage level) the process on how the data is stored, the external level (logical level), How the data are actually seen by the user. The Conceptual level (community logical level), actually deals on how the above two levels are interconnected. Here all the tables are normalized to interconnect by means of establishing relationships between the tables. It often expresses this connection by means of relationship between the real world entities.

On the other hand, Hadoop Framework [12] consists of two important layers, Hadoop, Distributed File System (HDFS) or so called as data storage layer and the Map-Reduces framework or the data processing layer. HDFS is generally a block structured file system managed by a central name node that is designed to run on heterogeneous environments. Several Individual files are broken into cubes of a set size and passed out across multiple Data Nodes in the cluster. The Name node maintains metadata about the size and location of Blocks and their reproductions.



**Fig 4.3.1 HDFS Architecture** **[13]**

The Map-Reduce Framework, works primarily by means of two main offices that is a Map and Reduce. The architecture is a Master Slave component. The Master here is a single job tracker and slaves or worker nodes are Task trackers. A piece of the task is split down into map tasks based on the number of data blocks that need processing and reduce jobs.

**4.4 RDBMS v's Hadoop [14]:**

1. RDBMS is generally designed for its queries or updates where the data has to be indexed to deliver the lowest latency retrieval and update times on a relatively small amount of data, whereas Hadoop includes Map-Reduce which is a good fit for problems that need to analyze the whole data set in a stack fashion, particularly for ad-hoc analysis.

2. Relational database often works on the structured data like on spreadsheets, normalized tables, etc. because it has predefined schema, whereas Hadoop works on the unstructured data like images, plain text data because it is designed to interpret the data during processing time (Schema on read).

3. In the case of updates, the traditional RDBMS suits the applications only when the data is written many times and read once. While in the case of Map-Reduce Scenario, the data is written once and read many times.

4. Relational databases include normalization property to keep its unity and remove redundancy, whereas for Hadoop, it performs high speed streaming of reads and writes in a nonlocal operation.

5. The efficiency of Hadoop can be maximized by means of scaling the size of the equipment in a linear fashionable way, whereas for RDBMs, it is nonlinear

6. Traditional RDBMS is involved in ACID transactions, while Map-Reduce has none of the transactions.

7. RDBMS is not an open source development, whereas Hadoop is an open source development.

8. The cost of maintenance for RDBMS is too high, whereas for Hadoop it is cheaper to implement.

9. The data size in traditional RDBMS is in Gigabytes, while for Map-Reduce it is Petabytes

10. Data failure in RDBMS is much more severe than the Hadoop.

**4.5 Use Cases for Relational databases and Hadoop:**

There are a lot of instances where in this big data world also we make use of Relational DB. One of the best example of use of Relational DBs is hive meta store. Hive is data warehouse software facilitates reading, writing, and managing large data sets residing in distributed storage using SQL. Here we have a data stored in the HDFS layer and we query data through hive to do analytics on top of it. But hive store all the metadata information internally in a Relational DB which is mostly MySQL.

Apache Hadoop is an open source implementation of Google’s Map-Reduce model and has become extremely standard over the years for building huge information analytics platform. Twitter Sentiment analysis of the tweets or any Social media data is a very important medium to know people’s view. The concern here is the data from social media is huge. The volume and speed they come is capable enough to crash our conventional relational databases. With Hadoop, we have a storage layer HDFS which can store such amount of data. Likewise, we can horizontally scale our storage by just adding simple commodity hardware to our HDFS cluster. Another important thing is we need to consume the data also at a certain rate so that we do not miss any tweets. This again can be handled using Big Data queuing system like KAFKA which is an industry standard queuing technology used by many technology giants like Twitter, LinkedIn itself to queue up the data flowing in the system. Also Facebook implements graph search and map reduce functionalities through Hadoop.

**5. Conclusion:**

Relational Databases and Hadoop are two different technologies that are implemented in certain scenarios. So it’s up to the computer data analyst to make a coherent decision to choose the right technology while performing operations on the type of data he is dealing with.

**6. References:**

[1]. Statistics of analyzing the data

<http://www.emc.com/leadership/digital-universe/2014iview/internet-of-things.htm>

[2]About IDC: <https://www.idc.com/about/about.jsp>

[3] Note:1 zettabyte=1021 bytes or equivalent to 1k Exabyte’s, one million petabytes or one

Billion terabytes., <https://en.wikipedia.org/wiki/Zettabyte>

[4] Defining the architecture components of the big data ecosystem.

<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6867550>

[5] Estimation analysis of data generated by IBM.

[http://www-01.ibm.com/software/data/bigdata/what-is-big- data.html](http://www-01.ibm.com/software/data/bigdata/what-is-big-%20%20data.html)

[6] A relational model of data for large shared data banks by cod. <http://dl.acm.org/ft_gateway.cfm?id=362685&ftid=49879&dwn=1&CFID=785266361&CFTOKEN=39957930>

[7] Apache Hadoop: <https://en.wikipedia.org/wiki/Apache_Hadoop>

[8] The CAP theorem versus databases with relaxed ACID properties. <https://dl.acm.org/ft_gateway.cfm?id=2557981&ftid=1443426&dwn=1&CFID=615410647&CFTOKEN=52952498>

[9] CAP Theorem Wikipedia <https://en.wikipedia.org/wiki/CAP_theorem>

[10] Brewer’s CAP theorem, twelve years later, how the rules have changed

[http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed](http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed%20)

[11] Relational database architecture Textbook: An Introduction to Database Systems, 8th

     Edition, C J Date.pdf. [13] HDFS architecture:

[12] Hadoop DB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads.[http://dl.acm.org/ft\_gateway.cfm?id=1687731&ftid=729753&dwn=1&CFID=785518120&CFTOKEN=36759838](http://dl.acm.org/ft_gateway.cfm?id=1687731&ftid=729753&dwn=1&CFID=785518120&CFTOKEN=36759838%20)

[13] HDFS Architecture :<https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html>

[14] Hadoop The definitive guide 4th edition text book: <http://javaarm.com/file/apache/Hadoop/books/HadoopThe.Definitive.Guide_4.edition_a_Tom.White_April-2015.pdf>