Comparisons on different Storage of Big Data Tools : Comparative Study

Prof. Hetal Nimit Shah [*], Prof. Jaideep Raulji [**]

[*] Assistant Professor, M.C.A. Department, D.D. University, Nadiad
[**] Ph D. , Navrachna University, Vadodara

DOI: 10.29322/IJSRP.12.04.2022.p12413
http://dx.doi.org/10.29322/IJSRP.12.04.2022.p12413

Paper Received Date: 13th March 2022
Paper Acceptance Date: 29th March 2022
Paper Publication Date: 6th April 2022

***Abstract*-** Big Data is emerging technology era implicates to handle huge amount of data to store, retrieve, manage, analyzed and processing**.** Big Data handles data like structured , semi structured and unstructured data for different applications like E-commerce, Hospital, HealthCare, Social Media, Cloud Computing, IOT based and many more. for storage, management and processing different tools are required to handle such peta and tera bytes amount of data. Traditional tools was not unable to perform management and analysis of complex, semi-structured ,and unstructured data. Big Data has different tools to handle, managed, querying different categories of mainly unstructured and semi-structured data using like Cassandra, Spark, Hadoop, Map-Reduce, Couch DB, Mongo DB and many more tools helping out developers to perform different operations on it.

***Index Terms*-** Big Data, Big Data Tools, Relational Data bases , Operations on Big Data, Run-Lenth Encoding

I. INTRODUCTION

Data which is huge and large amount of massive volume of data which are difficult to manage and thousands of operations required to perform on details to get valuable information is known as Big Data".

On Daily basis users can generate tera bytes and peta bytes data cannot manage easily. To storage and management of such data developers require different tools like Hadoop, Cassandra, Map Reduce, Spark, Kafka, CouchDB, Neo4J, Mongo DB, Sqlite and many more.

Table 1. Comparison between Big data and traditional data [2]

	Big Data	**Traditional**
Type of data	Semi and Unstructured Structured	Structured
Data storage	No SQL, Hadoop Distributed File System	RDBMS
Volume of data	Peta and Zetta bytes, Eexa Bytes, Tera bytes	Mega & Giga byte
Rate of data generation	Rapid	More time
Sources of data	Multiple sources	Centralized

On daily basis zeta or peta bytes of data generated by different organizations , government agenesis, social media, health care, finance, ecommerce websites to store different massive data various tools are required to process , manage , operating and generate valuable information from data. Earlier transaction and master data can easily managed by relational data bases.

The main objective of paper is to provide insights into processing, storing, retrieval, and management of different complex data applications with help of tools like No-SQL, New SQL, and SQL tools like Couch DB, Mongo DB , Cassandra, Hadoop and Map Reduce, Spark, Neo4J etc. Developers are required to use multiple tools for storage and analysis. As per our authors search Big Data Contains 10V's (volume, velocity, variety, variability, value, veracity, validity, vulnerability, volatility and visualization). Our Authors main focus is to store peta or tera or zeta bytes and processing using multiple Big Data Tools.

II. IDENTIFY, RESEARCH AND COLLECT IDEA

Table 2: Identifying following Data Which have different categories of examples with categories of Characteristics of Data

Category	**Big data (Distributed Data)**	**Traditional small data (Centralized Data)**
Type of data source	It includes data generated from real- time analysis (distributed data) Genomic Data Streaming data	Traditional enterprise data Student data

	Web log Data	Customer relationship Financial Data
Data Velocity	Ample of Data Generation on hourly and daily Basis, It requires for User to get faster response	Data generated on Batch mode, or near real time, more rapidly than big data . It does not require immediate response
Data Storage	NOSQL , HDFS	RDBMS
Data Integration	Difficult	Easy
Data Access	Interactive and faster	Batch mode and near real time

Our as a team main focus on data Storage and Comparisons of Different tools of Big Data. There are lots of tools for Big Data Storage for Semi-Structured and Unstructured as well as Structured Data.

Where to store the data and how long to keep them? Due to the variety of data, today's data may be stored in various databases (relational or NoSQL), data warehouses, Hadoop, etc. Today, database management is way beyond relational database administration. Because big data is also fast data, it is impractical to keep all of the data forever. Careful thoughts are needed to determine the lifespan of data.

Big data unavoidably needs distributed parallel computing on a cluster of computers. Therefore, we need a distributed data operating system to manage a variety of resources, data, and computing tasks.

Today, Apache Hadoop [10] is the de facto distributed data operating system. Apache Hadoop is an open-source software framework for distributed storage and distributed processing of big data on clusters of commodity hardware. Essentially, Hadoop consists of three parts:

- HDFS is a distributed high-throughput file system
- MapReduce for job framework of parallel data processing
- YARN for job scheduling and cluster resource management

III. WRITE DOWN YOUR STUDIES AND FINDINGS

"NoSQL" was coined in 1998 by Carlo Strozzi as the name for his then-new NoSQL Database, chosen simply because it doesn't use SQL for managing data.

The term took on a new meaning after 2009 when Johan Oskarsson organized a meetup for developers to discuss the spread of "open source, distributed, and non relational databases" like Cassandra and Voldemort. [4]

The following table includes several such data models, but please note that this is not a comprehensive list:

Operational Database Model	**Example DBMSs**
Key-value store	Redis, Memcache DB, Riak
Columnar database	Cassandra, Apache Hbase,
Document store	Mongo DB, Couch base
Graph database	Orient DB, Neo4j, Hyper-Graph DB

Our main research findings to get acquired the knowledge of above mentioned Big Data tools to understand the purpose as well as maintain Data Storage Capacity and NoSQL follows CAP theorem to handle complex, unstructured and semi-structured data for different applications.

NoSQL Databases follows CAP Theorem: Consistency, Availability, and Partition Tolerance.

Consistency:
- All replicas contain the same version of data
- Client always has the same view of the data(no matter what node)

Availability :
System remains operational on failing nodes
- All clients can always read and write

Partition tolerance
- Multiple entry points
- System remains operational on system split (communication malfunction)
- System works well across physical network partitions

Key Value Based Database Functionality With Example:

Key-value databases, also known as key-value stores, work by storing and managing associative arrays. An associative array, also known as a dictionary or hash table, consists of a collection of key-value pairs in which a key serves as a unique identifier to retrieve an associated value. Values can be anything from simple objects, like integers or strings, to more complex objects, like JSON structures. Key Value Databases main tasks are session management, message queuing , and caching.

Examples of Key Value Pairs are like given below:

Radis: An in-memory data store used as a database, cache, or message broker, Redis supports a variety of data structures, ranging from strings to bitmaps, streams, and spatial indexes.

Riak: A distributed key-value database with advanced local and multi-cluster replication.

Columnar databases Functionality with Example: sometimes called *column-oriented databases*, are database systems that store data in columns. This may seem similar to traditional relational databases, but rather than grouping columns together into tables, each column is stored in a separate file or region in the system's storage.

The data stored in a columnar database appears in record order, meaning that the first entry in one column is related to the

first entry in other columns. This design allows queries to only read the columns they need, rather than having to read every row in a table and discard unneeded data after it's been stored in memory.

Because the data in each column is of the same type, it allows for various storage and read optimization strategies. In particular, many columnar database administrators implement a compression strategy such as **run-length encoding** to minimize the amount of space taken up by a single column. This can have the benefit of speeding up reads since queries need to go over fewer rows. One drawback with columnar databases, though, is that load performance tends to be slow since each column must be written separately and data is often kept compressed. Incremental loads in particular, as well as reads of individual records, can be costly in terms of performance.

columnar databases used mainly for performing aggregate functions as well as beneficial to perform query processing faster.

Some Examples of Columnar DataBases are given Below:

Apache Cassandra: A column store designed to maximize scalability, availability, and performance.

Apache Hbase: A distributed database that supports structured storage for large amounts of data and is designed to work with the Hadoop software library.

Document-oriented Databases Functionality with Example: *Document-oriented databases*, or *document stores*, are NoSQL databases that store data in the form of documents. Document stores are a type of key-value store: each document has a unique identifier — its key — and the document itself serves as the value.

Differently from Relational databases models Document Oriented Data bases can store all the data not as object of tables or databases but it stores all the data of given object in a single document. Document stores typically data in JSON, BSON, XML or YAML as well as can store binary data as PDF Document.

Document-oriented databases have seen an enormous growth in popularity in recent years. Thanks to their flexible schema, they've found regular use in e-commerce, blogging, and analytics platforms, as well as content management systems. Document stores are considered highly scalable, with sharding being a common horizontal scaling strategy. They are also excellent for keeping large amounts of unrelated, complex information that varies in structure.

Some of the popular Examples from Document Databases are given below:

Mongo DB	A general purpose, distributed document store, MongoDB is the world's most widely used document-oriented database at the time of this writing.
Couch base	Originally known as Membase, a JSON-based, Memcached-compatible document-based data store. A *multi-model* database, Couchbase can also function as a key-value store.
Apache Couch DB	A project of the Apache Software Foundation, CouchDB stores data as JSON documents and uses JavaScript as its query language.

IV. GET PEER REVIEWED

V. IMPROVEMENT AS PER REVIEWER COMMENTS

VI. CONCLUSION

With Comparing different Big Data Storage tools all have their own Characteristics for storage of data, but Database Administrator or Developer need to select Data Base depends on

REFERENCES

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in Plastics, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] 1Satish Chandra Reddy Nandipati, 2Chew XinYing*, 3Mohd Adib Omar 1,2,3School of Computer Sciences, 11800, Universiti Sains Malaysia, Pulau Pinang, Malaysia

[3] Kalyan Nagaraj, G.S. Sharvani* and Amulyashree Sridhar, Department of Computer Science, RV College of Engineering, Mysore Road, R V Vidyanikethan, Bengaluru,

[4] www.digitalocean.com

AUTHORS

First Author – Prof. Hetal Nimit Shah, M.C.A. Department, Dharmnsinh Desai University and hnshah.mca@ddu.ac.in
Second Author – Dr. Jaideep Raulji, M.C.A., Ph.D. , Navrachna UNiversity and jaideepraulji@gmail.com .