IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 57-62

Big Data: The Distinct Between Hadoop and Hive

Aishwarya Rajagopalan, Kausika.S, Mrs.W.Rose Varuna
Bharathiar University, Coimbatore.
Assistant Professor Department of Information Technology Bharathiar University

Abstract: *In the modern era, abundant data have become available on hand to get voluminous information. Big data points out to datasets that are not only big, but also high in variety and velocity, which makes them difficult to handle using traditional tools and techniques. It can be structured, unstructured or semi-structured, resulting in incapability of conventional data management methods. The rate of data generation is so startling, that it has induced a pressing need to implement easy and cost-effective data storage and retrieval mechanisms. Technologies such as MapReduce & Hadoop are used to essence value from Big Data. Hadoop is wellendorse, standard-based, open source software framework build on the foundation of Google's MapReduce. This new data storage technology is HDFS. This file system is meant to support excessive amount of structured as well as unstructured data. Hive is a data warehouse infrastructure tool to operate structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.In this paper it is concluded that though both Hadoop and Hive are used in the different areas it is proved that Hadoop is one of the best tool.*

Keywords: *Big Data, Hadoop, Hive, Hadoop vs. Hive*

I. INTRODUCTION

Imagine a world without data storage; a place where every detail about a person or organization, or every aspect which can be documented is lost directly after use. Organizations would thus lose the ability to extract valuable information and knowledge, perform detailed analyses, as well as provide new opportunities and advantages. Data is the building block upon which any organization thrives. Now think of the extent of details and the surge of data and information provided nowadays through the advancements in technologies and the internet.

Big Data

Big data is a term that refers to data sets or combinations of data sets whose size (volume), complexity (variability), and rate of growth (velocity) make them difficult to be captured, managed, processed or analyzed by conventional technologies and tools, such as relational databases and desktop statistics or visualization packages, within the time necessary to make them useful[1].

A. Challenges of Big Data:

- *Volume of data:* Research studies have found that data volumes are doubling every year.
  Additionally a significant percentage of organizations are also storing three or more years of historic data. While the size used to determine whether a particular data set is considered big data is not firmly defined and continues to change over time, most analysts and practitioners currently refer to data sets from 30-50 terabytes(10 12 or 1000 gigabytes per terabyte) to multiple petabytes (1015 or 1000 terabytes per petabyte) as big data.

- *Variety of data:* Studies also indicate that 80 percent of data is unstructured (such as images, audio, tweets, text messages, and so on). The majority of enterprises have been unable to take full advantage of all this unstructured information. Big data processing mechanisms must know how to deal with eclectic data.

- *Velocity of data:* Velocity refers to the speed of data processing. For time-sensitive processes such as catching fraud, big data must be used as it streams into your enterprise in order to maximize its value. Sometimes, data may arrive at unprecedented speeds, and thus it must be dealt with in a timely manner. It should be processed in such a speed that is compatible for real time applications.

- *Variability:* Data flow can be inconsistent which can be challenging to manage.

- *Complexity:* The relationships between various attributes in a dataset, hierarchies and data linkages add to the complexity of data.

These are the challenges faced in Big Data.

II. BIG DATA TOOLS

Here are the top tools used to store and analyze Big Data. It can be categorized into two: Storage and Querying or Analysis.

A. HPCC

HPCC is a big data tool developed by LexisNexis Risk Solution. It delivers on a single platform, a single architecture and a single programming language for data processing.

Features:

- Highly efficient accomplish big data tasks with far less code.
- Offers high redundancy and availability
- It can be used both for complex data processing on a Thor cluster
- Graphical IDE for simplifies development, testing and debugging
- It automatically optimizes code for parallel processing.

B. STORM:

Storm is a free and open source big data computation system. It offers distributed real-time, fault-tolerant processing system. With real-time computation capabilities.

Features:

- It benchmarked as processing one million 100 byte messages per second per node
- It uses parallel calculations that run across a cluster of machines
- It will automatically restart in case a node dies. The worker will be restarted on another node
- Storm guarantees that each unit of data will be processed at least once or exactly once
- Once deployed Storm is surely easiest tool for Bigdata analysis

C. COUCH DB:

Couch DB stores data in JSON documents that can be accessed web or query using JavaScript. It offers distributed scaling with fault-tolerant storage. It allows accessing data by defining the Couch Replication Protocol.

Features:

- CouchDB is a single-node database that works like any other database
- It allows running a single logical database server on any number of servers
- It makes use of the ubiquitous HTTP protocol and JSON data format
- Easy replication of a database across multiple server instances
- Easy interface for document insertion, updates, retrieval and deletion

D. CLOUDERA:

Cloudera is the fastest, easiest and highly secure modern big data platform. It allows anyone to get any data across any environment within single, scalable platform.

Features:

- High-performance analytics
- It offers provision for multi-cloud
- Deploy and manage Cloudera Enterprise across AWS, Microsoft Azure and Google Cloud Platform
- Spin up and terminate clusters, and only pay for what is needed when need it
- Developing and training data models[5].

III. HADOOP

Hadoop is a Programming framework used to support the processing of large data sets in a distributed computing environment. Hadoop was developed by Google's MapReduce that is a software framework where an application break down into various parts.

The following Fig.1 describes about the High Level Architecture of Hadoop.

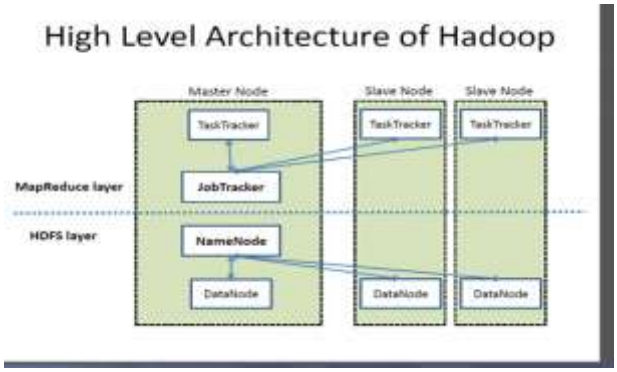

Fig.1 *Hadoop Architecture.*

A.HDFS Architecture

Hadoop includes a fault- tolerant storage system called the Hadoop Distributed File System, or HDFS. HDFS stores huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without losing data. Hadoop creates clusters of machines and coordinates work among them. Clusters can be built with inexpensive computers. If one fails, Hadoop continues to operate the cluster without losing data or interrupting work, by shifting work to the remaining machines in the cluster. HDFS manages storage on the cluster by breaking incoming files into pieces, called "blocks," and storing each of the blocks redundantly across the pool of servers. Usually, HDFS stores three complete copies of each file by copying each piece to three different servers. HDFS follows the Master- Slave Architecture. It has the following components.

The following Fig.2 describes about the HDFS Architecture.

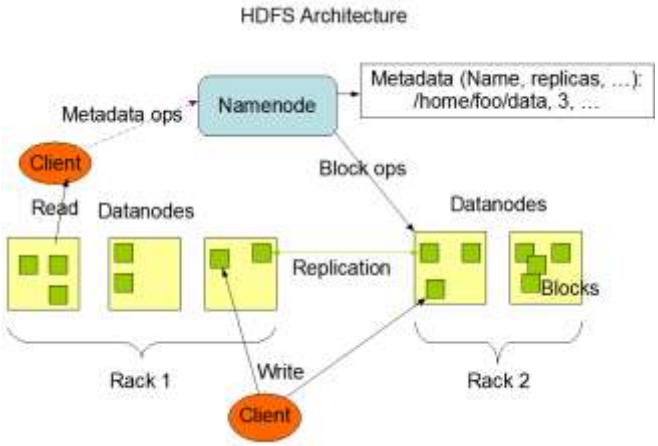

Fig.2 *HDFS ARCHITECTURE*

1) Name node: The HDFS consists of a single name node,the master node. It controls and manages the file system namespace. A file system namespace has a hierarchy of files and directories, where users can create, remove or move files based on their privilege. A file is split into one or more blocks and each block is stored in a Data node. HDFS consists of more than one Data Nodes. The roles of the name node are as follows:

a. Mapping blocks to their data nodes.

b. Managing of file system namespace

c. Executing file system operations- opening, closing and renaming of files[4].

2) Data node: The HDFS consists of more than one data node that stores the file blocks that are mapped onto it by the Name node. The data nodes are responsible for performing read and write operations from file systems as per client request. They also perform block creation and replication. The minimum amount of data that the system can write or read is called a block. This value is not fixed, and can be increased.

B. MapReduce Architecture

The processing pillar in the Hadoop ecosystem is the MapReduce framework. A distributed programming model based on the Java Programming language. The framework allows the specification of an

operation to be applied to a huge data set, divide the problem and data, and run it in parallel. The data processing frameworks are called mappers and reducers. In Hadoop, operations are written as MapReduce jobs in Java. Higher level languages like Hive and Pig that make writing these programs easier. The outputs of these jobs can be written back to either HDFS or placed in a traditional data warehouse. There are two functions in MapReduce as follows: It consists of two important tasks : Map and Reduce

1) Map stage: The map function takes in a set of data as the input, and returns a key-value pair as the output. The output of the map stage serves as input to the reduce stage.

2) Reduce stage: The function which merges all the intermediate values associated with the same intermediate key. The output of reduce stage is stored in the HDFS

The following Fig.3 describes the mechanism of Map Architecture.

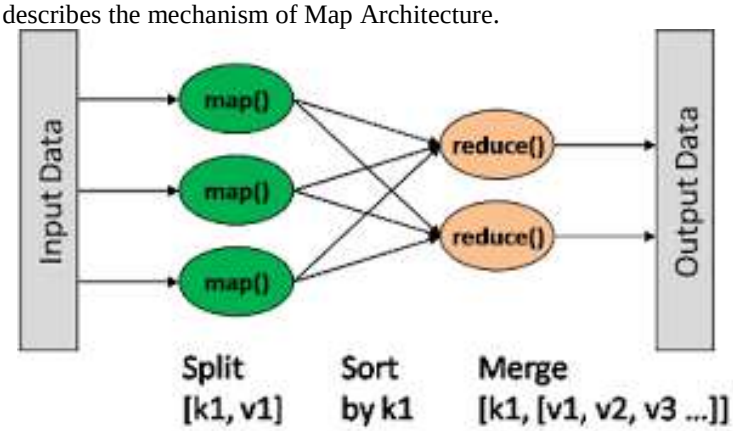

Fig.3 *MAP REDUCE ARCHITECTURE.*

The MapReduce framework is attractive due to its scalability.

IV. HIVE

Hive is evolved on top of Hadoop as its data warehouse framework for querying and analyzing of data that is kept in HDFS. Hive is an open source product that lets programmers analyze large data sets on Hadoop. Hive makes the job simple for accomplishing operations like analysis of large datasets. Map Reduce doesn't provide the features of optimization and usability, but Hive framework provides those features. Hive's SQL-inspired language segregates the user from the complication of Map Reduce programming. It reuses known notions from the relational database paradigm, such as tables, rows, columns and schema, to ease learning.

The hive can utilize directory structures to partition data to make better performance on specific queries. Metastore is used for keeping schema information. This metastore typically presents in a relational database. The user can communicate with hive using various ways; those are Web GUI and Java Database Connectivity (JDBC) interface. Most communications incline to take place over a Command Line Interface (CLI). Hive extends a CLI to write hive queries using Hive Query Language (HQL). In general, HQL syntax is identical to the SQL syntax. Hive supports four file formats such as TEXTFILE, SEQUENCEFILE, ORC and RCFILE (Record Columnar File). In a single user scenario, hive utilizes derby database for metadata storage and for multi user scenario, hive utilizes MYSQL to store metadata. Important difference between HQL and SQL is that, hive query processes on the Hadoop infrastructure rather than conventional database. Hadoop is a distributed storage, so when we submit hive query, it will appeal on huge data sets. The data sets are so large that high-end, expensive, conventional databases would fail to accomplish operations. Hive supports partition and buckets concepts for simple access of data when client executes the query. Hive supports custom specific User Defined Functions for data cleansing and filtering.

A. Hive Architecture:

The Fig.4 describes about the Hive architecture in Hadoop tool.

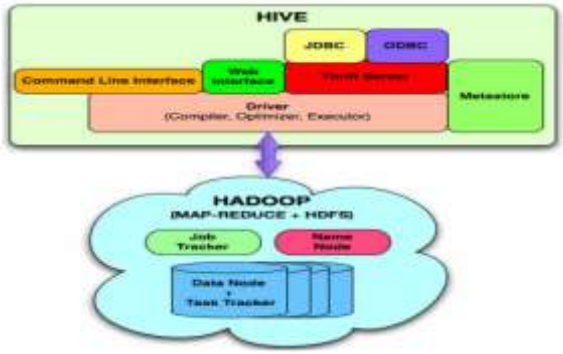

Fig.4 *Hive Architrcture*

B. Hive Functionality:

Hive Server is an API that enables the clients to process the queries on hive data warehouse and obtain the required results. Under hive server driver, compiler and execution engine communicate with each other and execute the query. The client posts the query via a GUI. The driver accepts the queries in the first instance from GUI and it will define session handlers, which will fetch required APIs that is modelled with different interfaces like JDBC or ODBC. The compiler generates the plan for the job to be processed. Compiler in turn is in touch with matter and it obtains metadata from Meta Store. Execution Engine is the vital component here to process a query by directly interacting with Job Tracker, Name Node and Data nodes. By running hive query at the backend, it will produce a series of Map Reduce Jobs. In this scenario, the execution engine acts like a bridge between hive and Hadoop to execute the query. For HDFS operations, Execution Engine communicates Name Node. At the end, it is going to fetch required results from Data Nodes. It will be having duplex communication with Metastore.

C. Modes of Hive Execution:

Hive employs in two modes. They are Interactive mode and Non Interactive mode. In Interactive mode, it directly moves to hive shell when you enter hive in command shell. The non-interactive mode is about processing commands directly in console file mode. The user can create two types of tables – Internal table and External table in both modes. The execution process of Hive with Hadoop environment is represented [5].

Hive Data Modelling:

Hive works with two types of table structures -internal and external, rely on the design of schema and how the data is getting placed into Hive,

I. *Internal Table:*

- Create tables
- Load the data.

II. *External Table:*

Data will be obtainable in HDFS; the table is going to get created with HDFS data. By dropping the table, it removed only schema, data will be obtainable in HDFS as before. External tables give a choice to create multiple schemas for the data kept in HDFS instead of dropping the data every time whenever schema upgrades[3].

V. HADOOP VS HIVE

In the given below table the Hadoop Vs Hive is discussed.

Hive	***Hadoop***
Designed and developed by Facebook	Designed and developed by Google
Data stored in external table, HBase or in HDFS	Data stored in HDFS only.
Supports HQL -Hive Query Language	Supports Multiple Program Languages such as Java, Python, Scala.[6].
Can work only on structured data.	Can process structured, Unstructured and semi-structured.

Data processed using HQL (Hive Query Language)	Data processed using MapReduce written in Java[6].
Framework computed with SQL-Like language	Framework computed with SQL and No-SQL
Database involved-Derby (default) also support MySQL, Oracle.	Database involved-HBase, Cassandra
SQL based program framework	Java based program framework

VI. CONCLUSION

Hadoop and Hive both are used to process the Big data. Hadoop is a framework which provides platform for other applications to query/process the Big Data while Hive is just an SQL based application which processes the data using HQL (Hive Query Language)Hadoop can be used without Hive to process the big data while it's not easy to use Hive without Hadoop. As a conclusion running both of the technology together can make Big Data query process much easier and comfortable for Big Data Users comparatively Hadoop is best than the Hive.

REFERENCES

[1]. http://www.ijsrp.org/research-paper-1014/ijsrp- p34125.pdf
[2]. https://pdfs.semanticscholar.org/501d/814f3f83e8ca44afbd3da13792f206a16402.pdf
[3]. http://www.ijcst.com/vol74/1/11-iqbaldeep-kaur.pdf
[4]. https://www.irjet.net/archives/V3/i1/IRJET-V3I1152.pdf
[5]. https://www.infoworld.com/article/2608271/hadoop/hadoop-review-apache-hive-brings-real-time-queries-to-hadoop.html
[6]. https://www.educba.com/hadoop-vs-hive/.