Volume: 7 Issue: IV Month of publication: April 2019
DOI: https://doi.org/10.22214/ijraset.2019.4075

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 7 Issue IV, Apr 2019- Available at www.ijraset.com

Implementation Analysis of MapReduce, Pig and Hive

Anushree Raj[1], Rio D'Souza[2]
[1] *MSc Big Data Analytics Department, St Agnes College Manglore*
[2] *Computer Science Department, St Joseph Engineering College Mangalore*

Abstract: In this era of information age, a huge amount of data generates every moment through various sources. This enormous data is beyond the processing capability of traditional data management system to manage and analyse the data in a specified time span. This huge amount of data refers to Big Data. Big Data faces numerous challenges in various operations on data such as capturing data, data analysis, data searching, data sharing, data filtering etc. HADOOP has showed a big way of various enterprises for big data management. Big data hadoop deals with the implementation of various industry use cases. Hadoop framework has been emerged as the most effective and widely adopted framework for Big Data processing. In this paper we discuss the implementation analysis of MapReduce, Pig and Hive approaches.
Keywords: Big Data, Hadoop architecture, MapReduce, Pig, Hive

I. INTRODUCTION

Huge amount of unstructured data requires an effective way to draw meaning full insights from it. The process involves filtering and aggregating the data to leverage it for business analytics. Big Data and Hadoop efficiently processes large volumes of data on a cluster of commodity hardware [1]. There are various coding approaches like using the Hadoop MapReduce or components like Apache Pig and Hive. Hadoop Mapreduce is a framework or a programming model in the Hadoop ecosystem to process large unstructured data sets in distributed manner by using large number of nodes. Pig and Hive are components that sit on top of Hadoop framework for processing large data sets without writing the Java based MapReduce code. Hadoop is for processing huge volume of data. Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models [2]. It is designed to scale up from single server to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high availability, the library itself is designed to depict and handle failures at the application layer, so delivering a highly available service on the top of a cluster of computers, each of which may be prone to failures. In the proposed work the second sections gives you a detailed review on Hadoop Architecture, third section speaks of MapReduce, Pig and Hive briefly and the fourth section shows the implementation analysis of three different programming approach and the fifth section includes a brief discussion and the sixth section concludes the paper by summarizing few drawings.

II. HADOOP ARCHITECTURE

The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop consists of four types of nodes—a NameNode, DataNodes, a JobTracker, and TaskTracker. HDFS nodes (NameNode and DataNodes) provide a distributed filesystem where the JobTracker manages the jobs and TaskTrackers run tasks that perform parts of the job [3]. Users submit MapReduce jobs to the JobTracker, which runs each of the Map and Reduce parts of the initial job in TaskTrackers, collects the final results. HDFS is a distributed file system in Hadoop. HDFS is used by MapReduce to read and write data [4]. HDFS supports to store massive amount of data and provides high –throughput access to data.

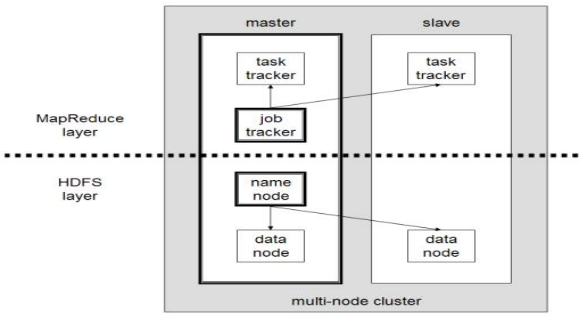

Figure 1. Hadoop Architecture

Hadoop framework is composed of four components.

A. MapReduce

MapReduce is a parallel programming model for writing distributed applications devised at Google for efficient processing of large amounts of data (multi-terabyte data-sets), on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner [3].

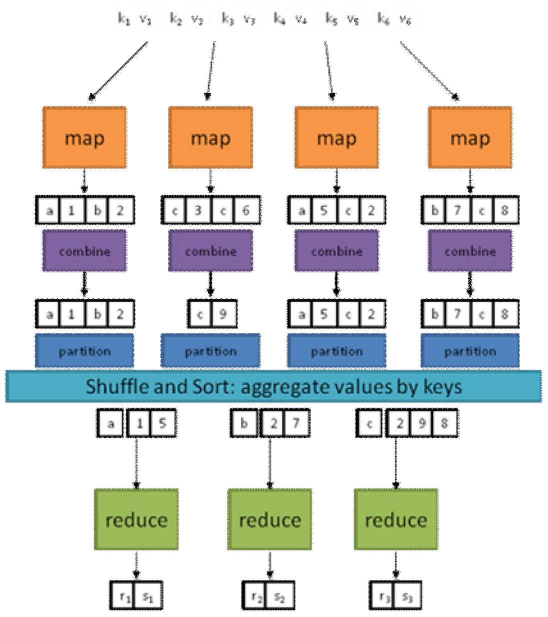

Figure 2. Map Reduce framework

MapReduce is generally defined as a programming model for processing and generating large sets of data. It is comprised of two unique functions:

map – the function takes key/value pairs as input and generates an intermediate set of key/value pairs.

reduce – the function which merges all the intermediate values associated with the same intermediate key.

A Map Reduce framework is based on a master-slave architecture where one master node handles a number of slave nodes [7]. Map Reduce works by first dividing the input data set into even-sized data blocks for equal load distribution. Each data block is then assigned to one slave node and is processed by a map task and result is generated. The slave node interrupts the master hub when it is sit without moving. The scheduler then assigns new assignments to the slave hub. The scheduler takes data location and resources into consideration when it distributes data blocks.

B. Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware.

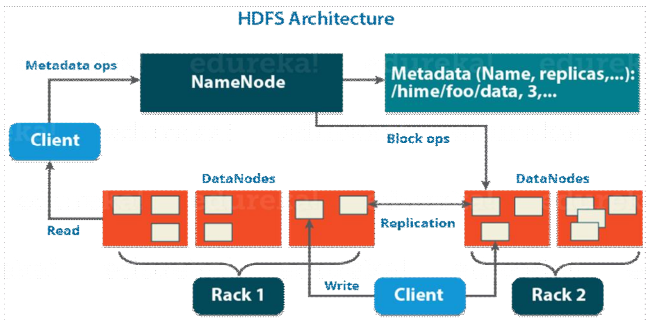

Figure 3. HDFS Architecture

Hadoop Distributed File System is a block-structured file system where each file is divided into blocks of a pre-determined size. These blocks are stored across a cluster of one or several machines [3]. Apache Hadoop HDFS Architecture follows a *Master/Slave Architecture*, where a cluster comprises of a single NameNode (Master node) and all the other nodes are DataNodes (Slave nodes). Name node is the master node, that manages the File System Namespace and controls access to files by clients. It maintains and manages the DataNodes. It records the metadata of all the files stored in the cluster [8]. It keeps a record of all the blocks in HDFS and in which nodes these blocks are located. DataNodes are the slave nodes in HDFS. DataNodesare the slave daemons or process which runs on each slave. The actual data is stored on DataNode. Secondary NameNode. It works concurrently with the primary NameNode. The Secondary NameNode is one which constantly reads all the file systems and metadata from the RAM of the NameNode and writes it into the hard disk or the file system. Blocks are the continuous location on hard drive where data is stored. HDSF stores each file as blocks which are scattered throughout the Hadoop cluster. Replication management: HDFS provides a reliable way to store huge data in a distributed environment as data blocks. By default it replicates into 3 blocks to provide fault tolerance. Therefore, if you are storing a file of 128 MB in HDFS using the default configuration, the space of 384 MB (3*128 MB) blocks will be replicated three times and each replica will be residing on a different DataNode. Rack Awareness: An in-built Rack Awareness Algorithm is used to ensure that all replicas are nt stroed on the same rack or a single rack.

1) *Hadoop Common:* These are Java libraries and utilities required by other Hadoop modules.
2) *Hadoop YARN*: This is a framework for job scheduling and cluster resource management.

Hive is a Hadoop-based data warehousing-like framework developed by Facebook. It allows users to fire queries in SQL, with languages like HiveQL, which are highly abstracted to Hadoop MapReduce [4]. This allows SQL programmers with no MapReduce experience to use the warehouse and makes it easier to integrate with business intelligence and visualization tools for real-time query processing.

The features of Hive are it is Hibernate Query Language (HQL), it supports UDF, it has metadata storage, t provides data indexing, it provides different storage type and providescHadoop integration.

III. MAPREDUCE, PIG AND HIVE

A. MapReduce

It is the core component of processing in a Hadoop Ecosystem as it provides the logic of processing. In other words, MapReduce is a software framework which helps in writing applications that processes large data sets using distributed and parallel algorithms inside Hadoop environment. In a MapReduce program, **Map() and Reduce()** are two functions. The **Map function** performs actions like filtering, grouping and sorting. While **Reduce function** aggregates and summarizes the result produced by map function. The result generated by the Map function is a key value pair (K, V) which acts as the input for Reduce function. As MapReduce is a programming paradigm, so you can write MapReduce programs in Java, Python etc.

B. PIG

Pig's programming language referred to as Pig Latin is a coding approach that provides high degree of abstraction for MapReduce programming but is a procedural code not declarative. At the backend of a Pig programme a mapreduce works. PIG has two parts: **Pig Latin**, the language and **the pig runtime,** for the execution environment. Pig has tools for data storage, data execution and data manipulation. . It gives you a platform for building data flow for ETL (Extract, Transform and Load), processing and analyzing huge data sets.

C. HIVE

Hive runs on workstation and converts SQL query into a series of MapReduce jobs for execution on a Hadoop cluster. Hive organizes data into tables, which provide a means for attaching structure to data stored in HDFS. Hive, makes it convenient to run the metastore on your local machine [2].

Apache Hive is similar to an SQL engine that has its own metastore on HDFS and the tables can be queried through a SQL like query language known as HQL (Hive Query Language). Hive queries are converted to MapReduce programs in the background by the hive compiler for the jobs to be executed parallel across the Hadoop cluster. The query language of Hive is called Hive Query Language (HQL), which is very similar like SQL. it can serve both the purposes, i.e. large data set processing (i.e. Batch query processing) and real time processing (i.e. Interactive query processing) and hence it is highly scalable.

IV. IMPLEMENTATION

We use of VMWare workstation to set up virtual machine on the physical machine. It serves as a cross platform. We implement on Hortonworks Sandbox environment and explore on Hortonworks Data Platform (HDP). It is software which has all hadoop components in it.

A. HADOOP

1) Installation of Server: To install the hadoop components using Hortonworks Sandbox, we first need to enable the virtualization in our system configuration during the booting process. The status is checked at MSinfo32 if the VM Virtualization is enabled. Next, install the VMWare workstation which helps to store and run any softwares. The minimum requirement for the mother operating system to run is 500MB.

2) Configuration: Open the Virtual Machine and Import the Hortanworks Sandbox. Ensure the VMWare software and the Hortanworks Sandbox is on the same folder.

3) Environment Set up: VMWare settings allow you to allocate memory for each section. But it physically doesn't allocate, since it reserves the minimum for motherboard. Using the play option we install the Hadoop Eco System components. Identify the IP address of hadoop so as to map with Windows. We use WINSCP Windows Secure Copy and lay down a connection to hadoop using its IP address as hostname. We login into hadoop through a secure username and password.

4) Working of hadoop environment

a) hadoop fs -mkdir /bingo

b) hadoop fs -ls /

c) hadoop fs -ls /bingo

d) hadoop fs copyFromLocal /root/test/sample.txt /bingo

e) hadoop fs –ls /bingo

We manually create a folder in hadoop. Then drag and drop a file into the hadoop folder from the system. Use mkdir to create a directory in HDFS. Then copy the file from the root to HDFS folder.

B. PIG Implementation

A Pig Latin program consists of a series of operations or transformations which are applied to the input data to produce output. These operations describe a data flow which is translated into an executable representation, by Pig execution environment. Underneath, results of these transformations are series of MapReduce jobs which a programmer is unaware of [10]. Pig allows the programmer to focus on data rather than the nature of execution.

1) Program to count number of words.

a) grunt> A = LOAD '/play/temp.txt' USING TextLoader() AS (Words:Chararray) ;

b) grunt> DUMP A ;

c) grunt> B = FOREACH A GENERATE FLATTEN (TOKENIZE(*)) ;

d) grunt> C = GROUP B BY $0 ;

e) grunt> D = FOREACH C GENERATE group, COUNT(B) ;

f) grunt> STORE D INTO '/output' ;

2) Program to output the selected fields from a given file using PIG

a) grunt> batting = LOAD '/play/cricket.csv' USING PigStorage(',') ;

b) grunt> DUMP batting ;

c) grunt> runs = FOREACH batting GENERATE $0 as playerID , $1 as year , $8 as runs ;

d) grunt> runs = filter runs by runs>0 ;

e) grunt> grp_data = GROUP runs by (year) ;

f) grunt> max_runs = FOREACH grp_data GENERATE group as grp_data, MAX (runs.runs) as max_runs ;

g) grunt> join_max_runs = JOIN max_runs by ($0, max_runs), runs by (year,runs) ;

h) grunt> join_data = FOREACH join_max_runs GENERATE $0 as year, $2 as playerID, $1 as runs ;

i) grunt> DUMP join_data ;

C. HIVE implementation

1) *Installation of HIVE*
2) *Running HIVE:* Apache Hive is similar to an SQL engine that has its own metastore on HDFS and the tables can be queried through a SQL like query language known as HQL (Hive Query Language). Hive queries are converted to MapReduce programs in the background by the hive compiler for the jobs to be executed parallel across the Hadoop cluster.
3) *Programming HIVE:* For hadoop developers who know SQL, Hive is MORE like SQL with clauses like FROM, WHERE, GROUP BY ORDER BY. In Hive the hadoop developers have to compromise on optimizing the queries as it depends on the Hive optimizer and hadoop developers to train the Hive optimizer on efficient optimization of queries [10]. Hive is generally used for processing structured data in the form of tables.

a) Program to demonstrate the word Count Using HIVE

```
hive>  drop table if exists doc;
hive> create table doc (text string) row format delimited fields terminated by '\n' stored as textfile;
hive> load data inpath '/play/temp.txt' overwrite into table doc;
hive> SELECT word , COUNT(*) FROM doc LATERAL VIEW explode(split(text,' ')) xTable as word GROUP BY word;
```

b) Program to output the selected fields from a given file using HIVE

```
hive>   Create table temp_batting (col_value STRING);
hive> LOAD DATA INPATH '/play/cricket.csv' OVERWRITE INTO TABLE temp_batting ;
hive>  Select * from temp_batting ;
hive>  create table batting (player_id STRING, year INT, runs INT)  ;
hive>  insert overwrite table batting SELECT regexp_extract (col_value, '^(?:([^,]*)\,?){1}',1) player_id, regexp_extract (col_value,
'^(?:([^,]*)\,?){2}',1) year, regexp_extract (col_value, '^(?:([^,]*)\,?){9}',1) runs from temp_batting ;
hive> SELECT year, max(runs) FROM batting GROUP BY year;
hive> SELECT a.year, a.player_id, a.runs from batting a JOIN (SELECT year, max(runs) runs FROM batting   GROUP BY year) b
ON (a.year = b.year AND a.runs = b.runs);
```

V. DISCUSSION

A serious of execution is performed on data set using Pig and Hive. Apache Pig and Hive are layered on top of hadoop, which uses hadoop MapReduce libraries. This paper shows the execution correlation of Pig and Hive. Flumes is used to secure information from online networking. The data acquired can be structured using the document framework GFS or HFS. Hadoop MapReduce, consists of several lines of basic java code requiring extra effort and time for code review and quality assurance. Thus, to simplify this Apache offers Pig Latin and Hive SQL languages which helps in constructing MapReduce programs easily. By using Hadoop MapReduce as the coding approach - it is hard to achieve join functionality making it difficult and time consuming to implement complex business logic [9]. The advantage is that MapReduce provides more control for writing complex business logic when compared to Pig and Hive [11].

VI. CONCLUSION

We summarise few drawings through the usage of the three programming approaches MapReduce, Pig and Hive. Hadoop MapReduce is a compiled language whereas Apache Pig is a scripting language and Hive is a SQL like query language. Pig and Hive provide higher level of abstraction whereas Hadoop MapReduce provides low level of abstraction. Hadoop MapReduce requires more lines of code when compared to Pig and Hive. Hive requires very few lines of code when compared to Pig and Hadoop MapReduce because of its SQL like resemblance. Pig and Hive coding approaches are slower than a fully tuned Hadoop MapReduce program. When using Pig and Hive for executing jobs, Hadoop developers need not worry about any version mismatch. There is very limited possibility for the developer to write java level bugs when coding in Pig or Hive. Pig has problems in dealing with unstructured data like images, videos, audio, text that is ambiguously delimited, log data, etc. Most of the jobs can be run using Pig and Hive but to make use of the advanced application programming interfaces, hadoop developers must make use of MapReduce coding approach. If there are any large data sets that Pig and Hive cannot handle for instance key distribution then Hadoop MapReduce comes to the rescue. Based on certain circumstances the choice is made to use Hadoop MapReduce over Pig and Hive.

REFERENCES

[1] Sachin Bende, Rajashree Shedge, “Dealing with Small Files Problem in Hadoop Distributed File System,” 7th International Conference on Communication, Computing and Virtualization 2016, Procedia Computer Science 79 (2016) 1001 – 1012

[2] Apache Hadoop. http://hadoop.apache.org/

[3] B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. Saleem Basha, P. Dhavachelvan, “Big Data and Hadoop-A Study in Security Perspective,” 2nd International Symposium on Big Data and Cloud Computing (ISBCC’15), Procedia Computer Science 50 (2015) 596 – 601

[4] “Hadoop, MapReduce and HDFS: A developer perspective,”(Procedia Computer Science, Volume 48, 2015,Pages 45-50)

[5] Kala Karun. A , Chitharanjan. K,” A Review on Hadoop – HDFS Infrastructure Extensions”, Conference on Information and Communication Technologies, 2013, IEEE

[6] DT Editorial Services, “Big Data(covers Hadoop2, Map Reduce, Hive, Yarn, Pig, R and Data Visualization)” by Dreamtech Press

[7] Tharso Ferreira, Antonio Espinosa, Juan Carlos Moure, PorfidioHern’ andez, “An Optimization for MapReduce Frameworks in Multi-core,” International Conference on Computational Science, ICCS 2013, Procedia Computer Science 18 (2013) 2587 – 2590

[8] Can Uzunkaya, Tolga Ensari, Yusuf Kavurucu, “Hadoop Ecosystem and Its Analysis on Tweets,” World Conference on Technology, Innovation and Entrepreneurship, Procedia - Social and Behavioral Sciences 195 (2015) 1890 – 1897

[9] PekkaPääkkönen, DanielPakkala1, “Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems,”

[10] https://intellipaat.com/tutorial/hadooptutorial/introductio n- hadoop/

[11] Naveen Garg , Dr. Sanjay Singla, Dr. Surender Jangra, “Challenges and Techniques for Testing of Big Data,” Procedia Computer Science 85 (2016) 940 – 948.