DOI: http://dx.doi.org/10.26483/ijarcs.v8i9.5177

ISSN No. 0976-5697

Volume 8, No. 9, November-December 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

PERFORMANCE COMPARISON OF HADOOP MAPREDUCE AND APACHE PIG

Chandra Kala Kuruba
Assistant Professor of CSE
Vignan Nirula Institute of Technology & Science for Women
Guntur, A.P. India

***Abstract:*:** In the recent era data has been growing at an exponential rate. This data can be either Structured or Semi-Structured or Unstructured and needs to be processed and analyzed carefully to get new insights. There are various tools or frameworks available for this purpose; few of them are Apache Hadoop MapReduce, Apache Pig,Apache Hive, Apache Spark,Tez etc. These tools are widely adopted for managing and processing BigData. Hadoop MapReduce provides low level of abstraction whereas Pig provides high level of abstraction. In this paper, we discuss the major architectural component differences between MapReduce and Pig and conduct detailed experiments to compare their performances with different inputs.

Keywords: Pig, Hadoop, MapReduce,HDFS, Big Data Analytics

I. INTRODUCTION

Data is growing at a fast pace in terms of both volume and velocity. This semi-structured data or unstructured data [1] could, however, provide new insights if analyzed carefully .This large amount of data can be processed in parallel by using MapReduce [2] framework and Apache Pig. Queries are divided and distributed across many nodes to be processed in parallel which is known as Map stage. Then the results are combined by Reduce stage and output is produced. This idea led to the building of the open source framework Hadoop for the distributed computing across multiple nodes[3]. Its major drawback was processing through repetitive datasets, which used to consume a significant amount of time. Apache Pig provide simple APIs, and hide the complexity of parallel task execution and fault-tolerance from the user. Pig [4] analyzes large data sets in a high-level language and run on top of Hadoop [2].It doesnot have storage capability so it has to depend on Hadoop HDFS or other storage systems. Apache Pig is an abstraction over MapReduce[5] that provides Pig Latin, which is a high-level language to write data analysis programs. Pig is a tool which can process both structured/unstructured data representing them as data flows, but also, those Pig Latin scripts are internally converted to Map and Reduce tasks. Pig can process data at petabyte scale [6]. When it comes to low-scale data, they consume more time.

II. DIFFERENCES

In this section, we discuss the underlying technical differences among Hadoop Mapreduce and Apache Pig.

Hadoop Mapreduce	**Apache Pig**
Compiled language	Scripted Language
Lower level of abstraction	Higher level of abstraction
More lines of code and based on JAVA	Computationally less number of lines than Mapreduce and pig is a data flow language
More development effort is involved	Development effort is less.
Code efficiency is high when compared to pig	Code efficiency is less

III. HADOOP

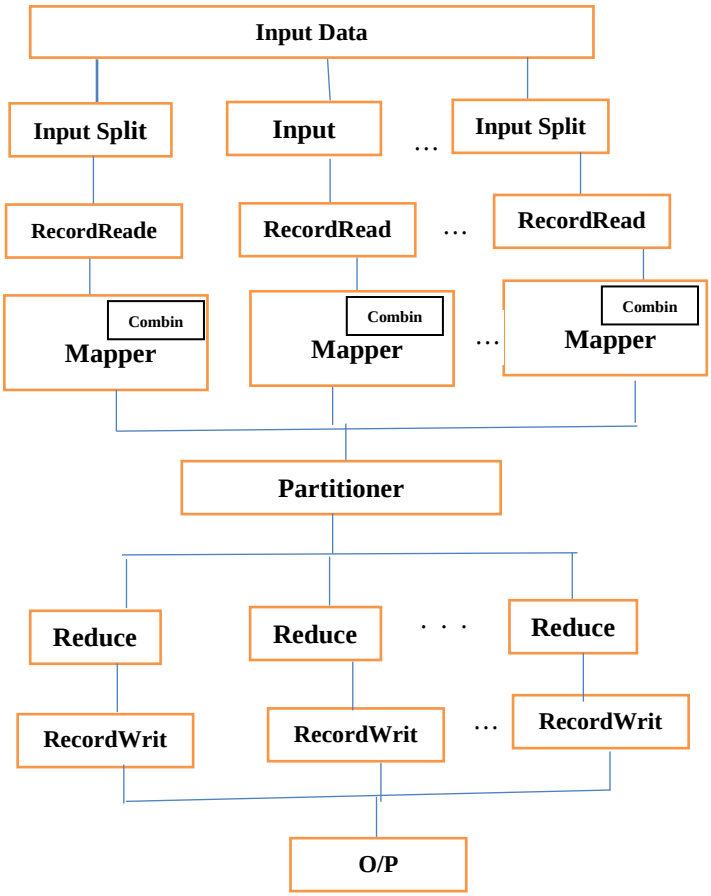
Hadoop has Apache Hadoop [6] has 2 components.

A. HDFS
B. MapReduce

A. HDFS (Hadoop Distributed FileSystem) : HDFS is a storage system which can store large volumes of different types of data. Hadoop cluster contains single master node called Name Node and many worker/slave nodes called Data Nodes for managing storage related issues. The data is stored in DataNodes and NameNode maintains the metadata and controls all data nodes. The computations are performed in data nodes.The backup (replication) of data for fault tolerant[4] system is also maintained in DataNodes. The default replication factor is 3. The DataNodes serves client's requests for reading and writing the blocks [7]. The architecture involves storing blocks of 64 MB each and this data can scale up to Gigabytes and Terabytes in the data nodes.

B. **MapReduce Engine**: It can process the large volumes of data parallel which is usually stored in HDFS. The Job Tracker receives the job from the user and split them into tasks and assign to Task Trackers to perform computations.

MapReduce Flow Chart



RecordReader: Reads input record one at a time and converts it into appropriate key and value pairs.
Mapper: Input (key,value) pairs are taken from RecordReader and performs necessary processing on those and a set of intermediate (key,value) pairs are generated.[7]
Number of Mappers = Number of Input Splits
Combiner:Combiner is an optional element. It is known as mini-reducer.It performs all operations like Reducer but at mapper site.
Partitioner: Partitioners are optional and are mainly responsible for partitioning/segregating intermediate key/values to the different Reducers.
Reducer: Performs shuffling (combines data having same key), sorting of key/value pairs and send output to RecordWriter.
RecordWriter: This element sends the output key/value pairs to output directory.

IV. APACHE PIG

Apache Pig is a tool used to process large volumes amounts of different types of data by representing them as data flows. Using pig Latin scripting language operations like ETL(Extract,Transform and load), adhoc data analysis and iterative processing can be easily achieved[3]

Few Pig Data Access / Data Transformation operators:

1. LOAD - To load data from HDFS to a relation/bag
2. DUMP – to dump the data from relation on to output device.
3. STORE - to store the results into a directory after processing
4. SPLIT - to split a relation into 2 or more sub relations
5. JOIN - to join two or more relations
6. UNION - To combine more than 1 sub relation to one relation
7. FOREACH – to generate specified data transformations based on column data
8. GROUP BY - to group data from one or more relations having same key.
9. SORT – to sort the given relation either in ascending or descending order.

Pig is an abstraction over MapReduce. All pig scripts internally are converted into map and Reduce tasks to get the task done. Pig was built to make programming MapReduce applications easier. Before Pig, Java was the only way to process the data stored in HDFS.

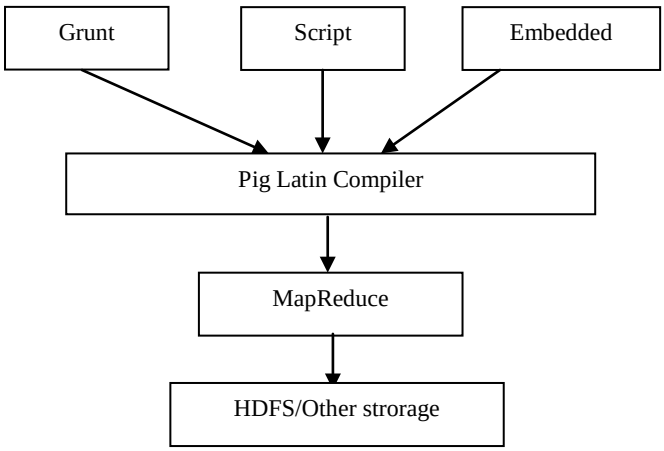


Fig: PIG Architecture

V. EXPERIMENT AND PERFORMANCE EVALUATION

Dataset: We have taken the dataset from github for performance analysis on wordcount program.
Experimental Setup:Cluster: 3 node cluster.
Node Configuration: 4GB RAM, 500GB Hard drive, Linux OS,Hadoop 2.7.1 version,pig 2.4.3 version.

Sample Code:

WordCount MapReduce program	WordCount Pig Script
public class SampleMapper extends Mapper<LongWritable,Text,Text,IntWritable> { protected void map(LongWritable key,Text record,Context ctxt) throws IOException,InterruptedException { String line= record.toString(); for(String word:line.split(" ") { ctxt.write(new Text(word),new IntWritable(1); } } } public class SampleReducer extends Reducer<Text,IntWritable,Text, IntWritable> { protected void reduce(LongWritable key,Iterator<IntWritable> list_of_values,Context ctxt) throws IOException,InterruptedException { int wcount=0; for(IntWritable total_value : list_of_values) { wcount+=total_value.get(); } ctxt.write(key,new IntWritable(wcount)); } }	Records = LOAD 'inputdata' AS (record:chararray); Words=FOREACH Records GENERATE FLATTEN(TOKENIZE(record) as word; Group_data = GROUP words BY word; WCount = FOREACH Group_data GENERATE group,COUNT(words); DUMP WCount;

TABLE.I . RESULTS:

Number of words	Apache Pig(in minutes)	Hadoop Map Reduce (in minutes)
1000	1.4	0.89
10000	2.1	1.22
100000	2.3	1.43
1000000	2.7	1.69
10000000	3.0	1.91

Table 1: Execution time comparision for wordcount program

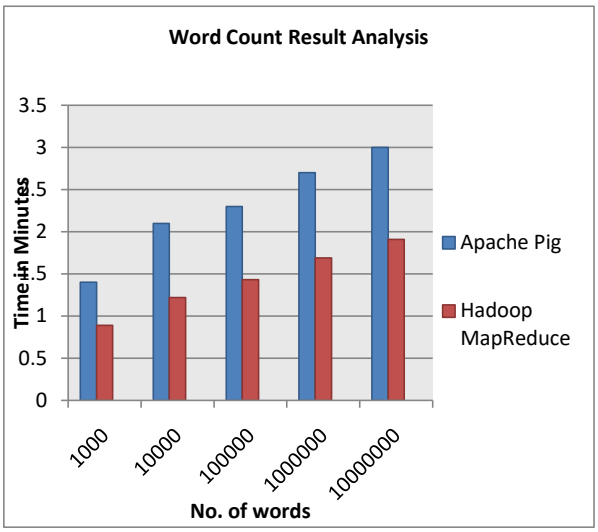


Figure 2: Result analysis of MapReduce & Pig

VI. CONCLUSION & FUTURE WORK

Both Hadoop MapReduce and Pig are found suitable for efficient processing of data of various applications. This paper focused on two issues, one is performance and other is development (lines of code). MapReduce takes lesser execution time as compared with Pig. But, developing a Mapreduce program involves more lines of code and one should know JAVA completely whereas writing a Pig Latin script is easier and doesn't require to know JAVA. The key aim of this paper is to check the execution time and difference between Mapreduce and Pig. It is found that MapReduce is more suitable and faster as compared to Pig. A huge dataset could be chosen to test other machine learning algorithms in both of the frameworks. We could further venture into other computation frameworks like Apache hadoop and Apache Spark or Tez and compare their performances in both single and multi node clusters.

VII. REFERENCES

[1] R. Blumberg and S. Atre, "The Problem with Unstructured Data," DM Review, pp. 42-46, 2003.

[2] Sanjeev Dhawan, Sanjay Rathee, "Big Data Analytics using Hadoop Components like Pig and Hive", American International Journal of Research in Science, Technology, Engineering & Mathematics, vol: 2(1), March-May, 2013, pp. 88-93.

[3] Jimmy Lin and Chris Dyer, "Data- Intensive Text Processing with Map Reduce", pp. 18-38, Morgan and Claypool publishers.

[4] Anshu Choudhary and C.S. Satsangi "Query Execution Performance Analysis of Big Data Using Hive and Pig of Hadoop", vol. Su-9 no.3, pp.91-93, Sep 2015.

[5] Akaash Vishal Hazarika and G Jagadeesh Sai Raghu Ram "Performance Comparision of Hadoop and Spark Engine" International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud).

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," USENIX Association, OSDI' 04: 6th Symposium on Operating System Design and Implementation, 2004, pp. 137 – 149.

[7] Jimmy Lin and Chris Dyer, "Data- Intensive Text Processing with Map Reduce", pp. 18-38, Morgan and Claypool publishers.