International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

A Review of Hadoop Ecosystems

Aparna Bali[1]

[1]*Student, Dept of Computer Science Engineering, Chandigarh University, Gharuan, Punjab*

Abstract - *The data being generated by the users is increasing at alarming rates. For storing and processing enormous data, big data frameworks are used. Scalability is the major concern when data to be analyzed is increased to terabytes or petabytes making it difficult to analyze, because a large data set necessitates high computing and storage resources. In this paper we will discuss about hadoop and ecosystem surrounding it. A basic overview about MapReduce, hdfs, hive, pig and mahout is provided.*

Key Words: **Bid data, Hadoop, HDFS, Hive, Pig, Mahout.**

1. INTRODUCTION

Big Data is a group of huge dataset who's processing using traditional computing techniques is not possible. Such large datasets are not easily captured using state of the art tools available. The size of the data being generated is increasing rapidly from few terabytes to multi petabytes. Thus new techniques are developed to analyse and process such large amount of data. There are 3 V's associated with big data, volume, velocity and veracity where volume indicates the increase in the generation of data, velocity indicates the pace with which the data is being generated and veracity means different type of data being generated, from text files to audio/video files all are covered.

1.1 Hadoop

It is a java based open source framework, where large datasets are processed across clusters of computers using basic programming structure. It works in an environment where distributed storage computation of different clusters is provided. Hadoop is capable of scaling up from single server to multiple machines each possessing its own storage and computation capability. There are two layers in the Hadoop architecture, processing layer which is called MapReduce and storage layer which is called Hadoop distributed file system.

Hadoop Working

It is rather exorbitant to build bigger servers with hefty configurations that are capable of handling large scale processing but alternatively, one can use many computers with single cpu making it a single distributed system where cluster machine can read data in parallel resulting in increased throughput. It is cheaper than running a single server with high computing requirement. Hadoop accomplishes the succeeding jobs while running code on computer clusters. Initially data is scattered into files and directories where they are divided into even sized blocks. For further processing, these files are distributed across cluster nodes where HDFS manages the processing as it is built on top of local file system. For controlling hardware failures, block replication is done. The file is sorted after map phase and before reduce phase. The sorted data is sent to a particular computer node at last debug logs for each job are written.

Hadoop Architecture

Hadoop architecture consists of two layers namely, MapReduce which is also called processing layer and Hadoop Distributed File System also called Storage layer.

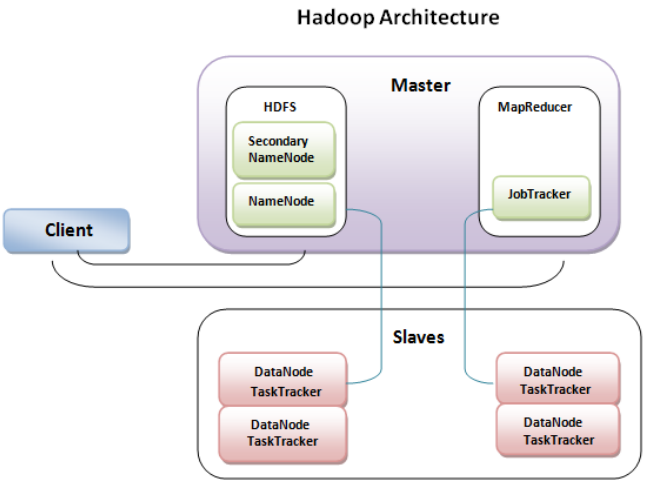


Figure-1: Hadoop Architecture

MapReduce

MapReduce is a model which is used for parallel processing over distributed computing clusters on large amount of data. MapReduce programs run on top of Hadoop framework and are composed of map () method and reduce method. In map method sorting, splitting and filtering of data is performed. In reduce method, summary operations of data shuffling and data reduction are performed. Thus the task are processed in parallel on multiple clusters in a redundant and fault tolerant way.

Map stage : The job of the mapper is to process the input data. Usually the input data may be a file or directory which is stock up in the Hadoop file system HDFS. That input file is passed line be line to the mapper function. The mapper

function processes the data and result in small chunks of files.

Reduce stage : In this stage, the data coming from the mapper is first Shuffled and then Reduced. The Reducer's task is to process the data coming from the map stage. The data after being processed, produces a fresh output, which is hoarded in the HDFS. Hadoop transmit the Map and Reduce jobs to the suitable servers available in the cluster. All the particulars of data-passing like issuing and verifying task conclusion, copying of data from the cluster amid the nodes is performed in this stage. when the tasks are completed, the data to be processed is collected from the clusters and it is reduced to form suitable results. These results are sent to hadoop server.

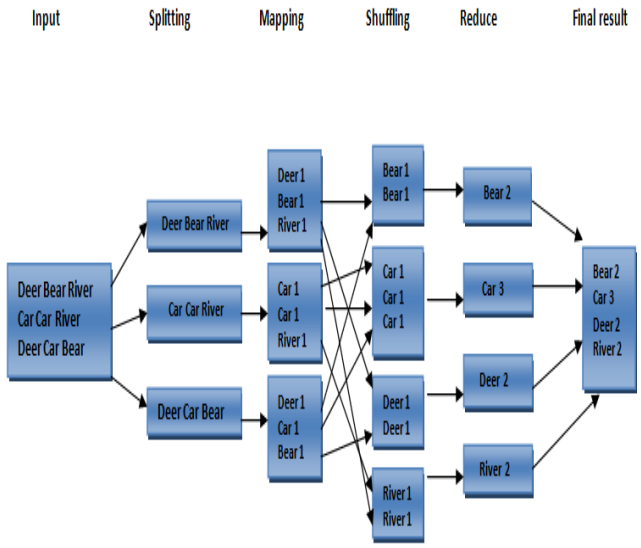


Figure-2: MapReduce Example

1.2 The Hadoop Distributed File system

When the storage capability of a single physical machine exceeds its limit as the dataset outgrows there arises a necessity to partition into number of different machines. File systems which supervise the storage space on a network of systems are distributed file systems. One of a major issue in local file system is that it is incapable of handling failures and result in data loss. HDFS is a distributed file system for Hadoop, which is extremely fault tolerant and prevents data loss unlike local distributed systems. HDFS is built on low cost hardware and it is capable of holding enormous quantity of data with easy access. To stock up such massive data, the files are scattered over multiple machines. To prevent data loss from system failure, files are redundantly stored which makes parallel processing possible.

HDFS Architecture

HDFS pursue the famous master-slave architecture consisting of following essentials elements.

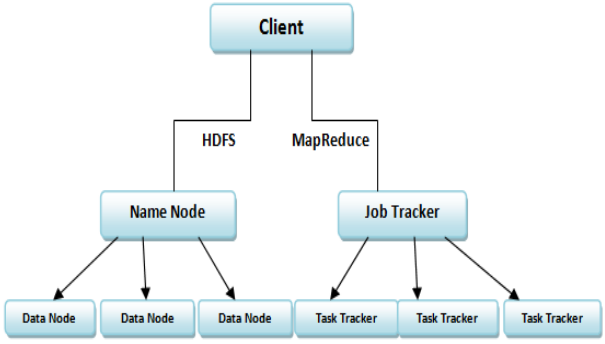


Figure-3: HDFS architecture

Namenode

The namenodes basically works on commodity hardware. The system comprising namenode operate as the master server. It administer the directory namespace and control users access to directory. Directory actions such as opening renaming and closing directories are also executed.

Datanode

The system comprising datanode operate as the slave server. The data storage of the system is managed by these nodes.As per client request, the Datanodes execute read and write operations on the directory. According to the directions of the namenode, operations such as block deletion, creation, deletion, and replication are also performed.

Block

The data is stored in files and directories of HDFS. The file stored in HDFS will be broken into segments and are stocked up in individual data nodes. The files broken into segments are called blocks. Also the least data that HDFS is capable of reading or writing is called a Block. According to HDFS configuration the default bock size being 64 MB can be increased.

1.3 Apache Hive

In the present day, Hive is a booming Apache project which is being used by numerous organizations as scalable and multi-purpose data processing platform which makes it possible for analysts having good SQL skills to run any query on enormous stocks of data stored in HDFS. Hive [18] is built on top of hadoop as a open source data warehousing tool. It is basically used for adhoc querying and data summarization. Hive supports queries written in a SQL like language called HiveQL. These queries are converted into map-reduce jobs implemented on Hadoop. It is extensive for users to ask diverse questions on the traffic data through the query interface. In common use, Hive converts SQL query into a sequence of MapReduce tasks for executing on a cluster of Hadoop nodes. In Hive data is organized into tables thereby providing a way for connecting structure to data which is stored in the HDFS where as Metadata like table schema, is loaded in a database which is called the metastore.

Hive Architecture

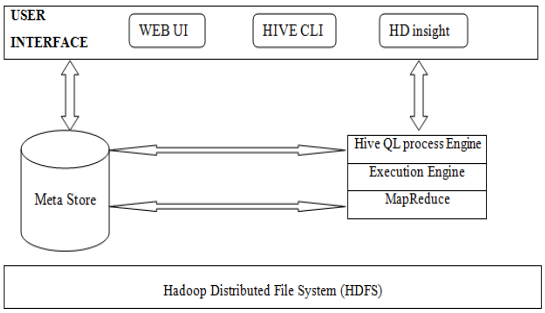


Figure-4: Hive Architecture

The various components of hive architecture are as follows

- User interface - Hive user interface is responsible for user and HDFS interactions. The interfaces supported by hive are Hive command line, web user interface.
- Meta Store - The instances or metadata of the tables created in hive are stored in a schema called Metadata. The metadata consists of columns, data types, tables etc.
- HiveQL process engine - It serves as a substitute of conventional MapReduce programs, where instead of writing programs in java, the purpose of map reduce job can be fulfilled by running simple queries on hive.
- Execution Engine - The combination of MapReduce and HiveQL process engine is Execution Engine. The queries are processed by execution engine and same results as that of MapReduce are generated.
- HDFS – HDFS stands for Hadoop distributed file system where data is stored in distributed nodes offering fault tolerance.

1.4 Apache Pig

Apache pig [17] is a platform for analysis of enormous sets of data demonstrating them as data flows. In Hadoop using Apache Pig, we can carry out every bit of the data manipulation operations. A high-level language pig latin is used for scripting data analysis programs. All the Pig Latin scripts are internally transformed to Map and Reduce tasks. The Pig Engine processes the Pig Latin scripts as input and transforms these scripts to MapReduce jobs. For carrying out mapreduce jobs in hadoop, it was not easy for Programmers not good in java to write programs. Pig acts as an advantage to such programmers. Without typing the long and complicated code, Mapreduce jobs can be easily performed using Pig Latin. The multi query method is used by pig which reduces the code length. For instance, a task which requires more than 300 lines of code in java can be easily replaced by less than 8 lines of code in pig leading to reduction in the development time to manifold. Being similar to sql, pig is simple to grasp and saves learning time. Many built in operators are available in pig for providing support to filter, sort, join etc operations. Moreover additional data types like bags, maps and tuples are provided by pig which is not present in MapReduce.

Apache Pig Execution Modes

Apache Pig is capable of running in two modes, HDFS mode and Local Mode.

Local Mode- In local mode, the entire files are run from local file system. There is no requirement of HDFS. Basically for testing function this mode is usually used.

MapReduce Mode- The data stored in hdfs is processed using pig MapReduce mode. The MapReduce jobs are called upon for processing data, when pig scripts are executed in this mode.

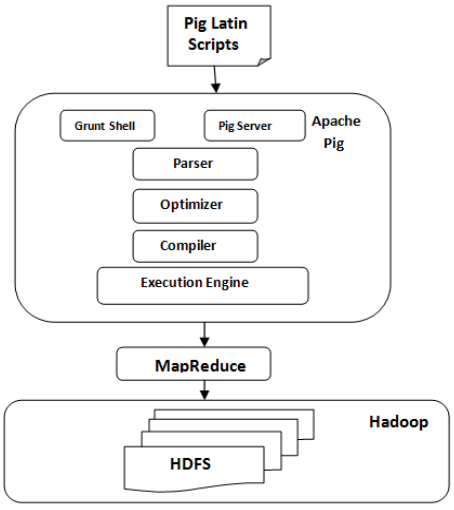
Apache Pig Execution Mechanisms

There are three modes for executing scripts in Apache Pig, interactive mode, embedded mode and batch mode.

- Grunt shell (Interactive Mode) – Using the Grunt shell we can run pig in interactive mode, where we type statements in pig Latin syntax and receive output.
- Script (Batch Mode) – by writing a file with .pig extension, we can run pig scripts in batch mode.
- UDF (Embedded Mode) – There is a provision for creating user defined functions in any other language and embedding them into pig scripts.

Apache Pig Architecture

To play out a specific undertaking Programmers utilizing Pig, software engineers need to compose a Pig script utilizing the Pig Latin dialect, and execute them utilizing any of the execution systems (Grunt Shell, UDFs, and Embedded). After execution, these scripts will experience a progression of changes connected by the Pig Framework, to deliver the fancied yield. Inside, Apache Pig changes over these scripts into a progression of MapReduce occupations, and in this way, it makes the software engineer's employment simple. The design of Apache Pig is demonstrated as follows.



Figure–5: Pig architecture

- Parser- First of all the scripts are sent to the parser where it is investigated for syntax and other assorted

checks. As an output, a directed acyclic graph is generated representing logical operators and pig Latin statements where nodes represent logical operators and edges represent data flows.

- Optimizer- Projection and pushdown are some logical optimizations which are carried out by logical optimizer on the logical plan i.e. DAG.
- Compiler- The task of the compiler is to assemble the optimized logical plan to MapReduce jobs.
- Execution engine- At last, the MapReduce jobs are sorted and presented to Hadoop. Finally, desired results are produced when these jobs are executed.

1.5 Apache Mahout

Mahout is a development of apache foundations having machine learning algorithms implementation. A person who controls and rides an elephant is called mahout. As mahout rides on an elephant in the same manner, apache mahout is built on top of hadoop so the name is justified. Mahout is capable of dealing with huge data sets as hadoop makes it scalable. With the increase in the data being used there is a considerable increase in the accuracy and performance of the model built. Mahout should be preferred if the training dataset varies from 10 lakhs to 1 crore. In traditional ML-algorithms, with the hike in the size of data, there are memory and performance constraints making algorithm slow. Whereas, algorithms in mahout are scalable and works in parallel and consumes less time. However mahout lacks any user interface, it consists of only java libraries. As a brief summary, mahout consist of implementations of various ML-algorithms in the field of clustering, classification, recommendations etc which are available in ready to use mode, there is no need to invest time in making algorithms. Presently, Mahout provides support for the subsequent use cases:

Recommendation: A complete API support is provided by mahout to build a recommendation engine based on user preference. The user data is taken as input and predicted data includes items that user might like based on its usage behaviour and previous patterns. On the basis of past user pattern, items of use can be predicted. For instance, when a bag is selected, a list of other bags also gets displayed as recommendations. Or when playing a video on YouTube, other similar videos are recommended.

Classification: The extent to which an article belongs to a particular group is decided by classification. A typical instance of classification is spam filtering for classification of Emails. A rich set of APIs are provided to build our personal classification model. As an illustration, a email or file classifier which can classify it as malicious or benign.

Clustering: In this technique, items are grouped together on the basis of certain similarity. The name of the clusters is not known before hand and on the basis of certain properties we find different item clusters. A major distinction between clustering and classification is that in latter end name of class is known. The clustering is used by goggle news for grouping news. K-means, canopy is widely used clustering algorithms.

Dimensional reduction: As it is known features are termed as dimensions. The decrease in the amount of random variable under consideration is termed as feature reduction. Algorithms are provided for feature reduction in mahout. For instance mahout provides single value decomposition algorithm.

2. LITERATURE REVIEW

Ibrahim, Lena T et.al (2015) [1] expressed concern over data explosion issues. Many organizations have to face the challenge to handle, capture, and monitor the data due to the ever-increasing volume of data sets, varying from quite a few terabytes to manifold petabytes. The author proposed a method to resolve the issue by introducing a traffic supervising system based on hadoop which performs analysis of internet traffic at a large scale.

Anjali P.P et.al (2013) [2] did a survey on broad scope of Apache Pig framework. The key features of the network flows were extracted for data analysis using available tools for packet capture. The flow analyzer based on Pig was implemented and traffic flow analysis was done easily without much programming skills. It reduced the time consumption greatly by reducing the complex programming constructs required for a MapReduce concept using Java programs. Programming with Pig has the advantages of Map/Reduce jobs because of the layered design and built in compilers.

Sean Owen et.al (2012) [3] described about recommendations, classifications and clustering in apache mahout. In machine learning systems, the accuracy of the system built depends on the size of data used. As the amount of training set is enlarged, so is the Mahout's performance. The classifier model was built and trained by teaching an algorithm with a set of examples. Then evaluation and tuning of classifier model was done to get better answers.

Ashish Thusoo et.al (2010) [4] gave a detailed discription about hive which is an open source data warehousing tool built on top of hadoop. It is basically used for adhoc querying and data summarization.It supports queries written in a SQL like language referred as HiveQL. These queries are converted into map-reduce jobs implemented on Hadoop. Users ask diverse questions about the network traffic via the query interface. As the queries will be transformed into MapReduce programs, their performance relies on query optimization.

3. CONCLUSIONS

It is difficult to analyze enormous data generated using traditional tools and there is lot of big data challenges faced by organizations these days. In this paper an overview about ecosystem surrounding hadoop is provided. Hadoop is used for large scale analysis and performs parallel processing by using map reduce algorithms, hive is a data warehousing framework for providing data summaries whereas pig is used to create map reduce programs using pig Latin scripts and mahout is a machine learning framework that is used for classification and training of large datasets.

REFERENCES

[1] Ibrahim, Lena T., Rosilah Hassan, Kamsuriah Ahmad, and Asrul Nizam Asat. "A study on improvement of internet traffic measurement and analysis using Hadoop system." In Electrical Engineering and Informatics (ICEEI), 2015 International Conference on, pp. 462-466. IEEE, 2015.

[2] Anjali PP, Binu A. Network Traffic Analysis: Hadoop Pig vs Typical MapReduce. arXiv preprint arXiv. 2013 Dec 19; 1312-5469

[3] Owen, Sean, Robin Anil, Ted Dunning, and Ellen Friedman. "Mahout in action." (2012).

[4] Thusoo, Ashish, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Ning Zhang, Suresh Antony, Hao Liu, and Raghotham Murthy. "Hive-a petabyte scale data warehouse using hadoop." In 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pp. 996-1005. IEEE, 2010.

[5] Mohanty, Hrushikesha. "Big Data: An Introduction." In *Big Data*, pp. 1-28. Springer India, 2015.

[6] McAfee, Andrew, Erik Brynjolfsson, Thomas H. Davenport, D. J. Patil, and Dominic Barton. "Big data." *The management revolution. Harvard Bus Rev*90, no. 10 (2012): 61-67.

[7] Swan, Melanie. "The quantified self: Fundamental disruption in big data science and biological discovery." *Big Data* 1, no. 2 (2013): 85-99.

[8] Marz, Nathan, and James Warren. *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co., 2015.

[9] Olston, Christopher, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. "Pig latin: a not-so-foreign language for data processing." In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1099-1110. ACM, 2008.

[10] About big data frameworks http://www.tutorialspoint.com/big_data_tutorials.htm/ date accesed- 1 july 2016.