Assignment 02

Apache Hadoop and Distributions

Name: Dean D'souza

H.U. ID: 168424

Apache Hadoop

Apache Hadoop is an open-source software library which aims at providing a framework for distributed file storage and distributed processing of large data sets residing on commodity grade hardware in the form of clusters of computers. This framework was developed as a result of two papers published by Google, the Google File System paper and the MapReduce: Simplified Data Processing on Large Clusters paper, and was mostly developed in Java programming language, with native code and command line utilities written in C and as shell scripts, respectively.

In order to process large data sets, Hadoop makes use of packaged code (usually in JAR format) [2][3] which is distributed to each node in the Hadoop cluster, which themselves store a large block of a file or data set. This code is then run on these large chunks in parallel in order to process the data set faster and more efficiently as compared to a conventional supercomputer architecture. [2]

The most important components of the **Apache Hadoop framework** are:

1. **The Hadoop Distributed File System (HDFS):** It is responsible for storage and high-throughput access of large data sets. [2] The data to be stored is essentially split into large blocks (usually of the order of 128MB) and distributed across the nodes of the Hadoop cluster.
2. **MapReduce:** It is a YARN (Yet Another Resource Negotiator) based system for the distributed and parallel processing of such data sets. This system utilizes key/value pairs in order to accomplish the processing tasks. The key part of this pair gives an idea of the kind of data, while the value part gives the actual instance of data associated with that key. The MapReduce architecture itself consists of two parts:
    a. Map phase: In this phase the individual records form the data source forms the input for a map() function which produces one or more intermediate values along with an output key. This forms the input for the reduce() function.
    b. Reduce phase: In this phase the output of the map() function is used as an input, such that the intermediate values are combined into one or final values for the same key.

The Apache Hadoop framework also has libraries and utilities required by the different modules stored in the Hadoop Common Module and a job scheduling framework that manages cluster resources, i.e., the computing resources in each cluster, and their utilization for user applications, known as Hadoop YARN. [1][2]

The **Hadoop architecture** itself is considered to have three basic layers having a logical hierarchy as follows:

1. Application Layer/End user access Layer: This layer acts as a point of contact for applications (which themselves could be personalized solutions or third party tools such as business intelligence, etc.) with the Hadoop environment.
2. MapReduce workload management layer: This layer is also known as the Job Tracker layer and is responsible for providing a job execution engine which co-ordinates all the aspects of the Hadoop environment, which includes scheduling and launching jobs, balancing workload, etc. [4]
3. Distributed parallel file system/Data Layer: This layer takes care of information storage, usually through HDFS, and may also consist of commercial or Third-party implementations.

When we talk about **cluster components**, following are the important components:

1. Master Node: This node acts as the primary controller of other nodes and itself consists of a Job Tracker, Task Tracker and Name Node. There are usually a number of these nodes to eliminate the risk of a single point of failure.
2. Data Nodes: This node is responsible for storing data in HDFS, replicating the data across clusters (usually the data is replicated at least three times) and interacting with client applications.
3. Worker Node: This node includes a data node and a task tracker and is responsible for providing processing power for analyzing the data.

The main advantages of Apache Hadoop include the ability to distribute data and computation over a network of consumer grade hardware, with mostly independent tasks executed at each node, while providing a simple programming model where the end-user programmer needs to design the MapReduce tasks. Hadoop programs also have a flat scalability curve allowing for addition of more nodes easily and with minimal re-work for applications. It also provides high reliability of data with built-in fault tolerance and high-throughput, allowing for fast processing, access and efficiency of data used, while also allowing multiple users at a time. Additionally, this system gets rid of the over-head of caching data by directly reading from the source every time. Also, from a programming perspective, the modular nature of Hadoop also makes it easier to modify and shape to an end-user needs.

However, Apache Hadoop also has the disadvantages of difficult Cluster and Task management, joins of multiple data sets being slow and complicated and the optimal configurations for different deployments not being obvious enough for easier implementation. Additionally, some of the main concerns revolve around Hadoop not being suitable for small data sets, low security due to improper implementation in java and missing encryption methodology for storage and for transmission over the network and some stability issues.

Another factor that could be viewed as either an advantage or a disadvantage is that of the Hadoop MapReduce and HDFS being under active development and is open-source, which could lead to the need for updating the consumer applications frequently to incorporate new features or changes. However, this also means that issues are generally resolved quickly due to the vast user base and contributor base.

Cloudera

Cloudera was one of the first distributions of Apache Hadoop based software which still as a large user base. It is an open-source Apache Hadoop distribution which is aimed at being an enterprise level deployment of Hadoop as a data hub. However, it does require commercial license purchases for actual

use (training Virtual Machines and 60-day trials are available for free) and is hence not completely open source as it also comes packaged with proprietary tools like Cloudera Management Suite for automated installation and the company Cloudera Inc. provides a number of software, services and support. One such software is the Cloudera Navigator Optimizer (in beta), which is a SaaS based tool for instant insights of the workload and for providing recommendation on optimization strategies.

The most recent Cloudera Hadoop distribution (CDH), has also incorporated several open-source projects such as Impala, which allows for execution of interactive SQL queries directly against the data stored in HDFS at faster speeds, and Apache Solr, which is used to index and search the data more efficiently, which also helps in faster execution of SQL queries. CDH can also be run on a windows server. Cloudera offers the option of using HDFS, Apache HBase and Amazon Simple Storage service as for data storage and also provides Cloudera Manager for simple administration of clusters. It also uses HttpFS for web access of files. It also supports a number of programming languages for end-user application development.

The Cloudera distribution also follows a master-slave architecture, with a shared-nothing computing framework which supports both MapReduce and YARN.

Hortonworks

Hortonworks is an open source distribution based on Apache Hadoop which is aimed at analyzing, storing and managing Big Data. It is also one of the few distributions which does not come packaged with additional proprietary software (can be downloaded for free) and can be run on windows servers natively. It uses HDFS or Network File Storage (NFS) for file storage and access and uses WebHDFS for web access. Recent release of Hortonworks incorporates Apache Tez (Stinger) which provides a developer API and framework for writing native YARN applications. This allows for native integration with the Apache Hadoop YARN for better performance under mixed workloads. It also incorporates Apache Solr for better data search.

Hortonworks also utilizes Apache Ambari for installation administration, and Ganglia or Nagios for monitoring. Hortonworks has also made advances for better integration with Apache Spark for faster processing and data access and also supports a number of programming languages for application development. Hortonworks provides the closest resemblance to native Apache Hadoop system along with the philosophy of being completely open source. However, it does try to make improvements over the traditional Apache Hadoop and many contributions in Apache Hadoop's development are made through Hortonworks.

The Hortonworks distribution also follows a master-slave architecture, with a shared-nothing computing framework which supports both MapReduce and YARN.

MapR

MapR is a licensed distribution of Apache Hadoop which makes a number of improvements such as providing full data protection, no single point of failure, improved performance and ease of use for programmers and administrators through simplified interfaces. [8] It is also written in C/C++ which executes much faster as compared to Java based distributions. It provides three levels of licenses with M3 offering a free distribution (but without much improvements). M3 is also supposed to be integrated with Ubuntu soon. The highest level license of M7 provides rewrite of HBase which implements the

HBase API directly in the file-system layer. It can also come packaged with MapR-DB if required, which allows for close to zero administration requirements.

It utilizes a MapR-FS file system (a proprietary file system for enterprise grade features) or Direct Access NFS for file access and uses NFS for web access. MapR also uses a MapR Control System for administration. Overall MapR is considered to be a more production ready distribution of Apache Hadoop. However, when it comes to writing custom applications, MapR can be a bit tedious as it deviates from the traditional Apache Hadoop distribution implementation on a number of cases. Additionally, many consider it quite expensive to obtain and use efficiently.

References:

[1] Apache Hadoop homepage, The Apache Software Foundation, (http://hadoop.apache.org)

[2] "Apache Hadoop", Wikipedia, (https://en.wikipedia.org/wiki/Apache_Hadoop)

[3] "JAR (file format)", Wikipedia, (https://en.wikipedia.org/wiki/JAR_(file_format))

[4] "Hadoop for Dummies", Robert D. Schneider, published by John Wiley & Sons. Inc., 2012

[5] "Hadoop-Advantages and Disadvantages", J2EEBrain, (http://www.j2eebrain.com/java-J2ee-hadoop-advantages-and-disadvantages.html/1)

[6] "Cloudera vs Hortonworks vs MapR: Comparing Hadoop Distributions", Experfy Editor, (https://www.experfy.com/blog/cloudera-vs-hortonworks-comparing-hadoop-distributions/)

[7] "Cloudera", Wikipedia, (https://en.wikipedia.org/wiki/Cloudera)

[8] "MapR", Wikipedia, (https://en.wikipedia.org/wiki/MapR)

[9] "Comparing the Top Hadoop Distributions", Kirill Grigorchuk (Director of R&D at Altoros), Network World, (https://www.netowrkworld.com/article/2369327/software/comparing-the-top-hadoop-distributions.html)