A Survey Paper on Hadoop Framework

R. Krishnaveni Jeenath
Department of Information Technology
Assistant Professor, Dr.Sivanthi Aditanar College of Engineering, TAMILNADU

Abstract- *The Apache Hadoop is an open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. Apache Hadoop is an open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster.*

Keywords- Hadoop, HDFS

I. INTRODUCTION

Hadoop is an open-source data platform or framework developed in java, dedicated to store and analyse the large sets of unstructured data. With the data exploding from digital mediums, the world is getting flooded with cutting-edge big data technologies.

II. HADOOP FRAMEWORK

Hadoop framework is composed of the following modules:

- Hadoop Common - It contains the Java libraries and utilities needed by other Hadoop modules.
- HDFS - Hadoop Distributed File System – Hadoop Distributed File System is a part of Hadoop framework, used to store and process the datasets. It provides a fault-tolerant file system to run on commodity hardware.
- MapReduce - It is a parallel programming model for processing large amounts of structured, semi-structured, and unstructured data on large clusters of commodity hardware.
- Hadoop YARN - A framework for cluster resource management.

2.1. Components in Hadoop Framework

CHUKWA:

Chukwa is an open source data collection system for monitoring large distributed systems. Chukwa is built on top of the Hadoop Distributed File System (HDFS) and Map/Reduce framework It inherits Hadoop’s scalability and robustness. Chukwa also includes a flexible and powerful toolkit for displaying, monitoring and analysing results to make the best use of the collected data

AVRO:

Avro is a remote procedure call and data serialization framework developed within Apache's Hadoop project. It uses JSON (JavaScript Object Notation) for defining data types and protocols, and serializes data in a compact binary format. Its primary use is in Apache Hadoop, where it can provide both a serialization format for persistent data, and a wire format for communication between Hadoop nodes, and from client programs to the Hadoop services. Using Avro, big data can be exchanged between programs written in any language Eg: C, C++, C#, Python. Ruby

ZOOKEEPER:

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. All of these kinds of services are used in some form or another by distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. Even when done correctly, different implementations of these services lead to management complexity when the applications are deployed.

HIVE:

Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.

PIG:

Pig is a platform for analysing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure supports parallelization, which in turns enables them to handle very large data sets. Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs, for which large-scale parallel implementations already exist. Pig's language layer currently consists of a textual language called Pig Latin.

SQOOP:

Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases.

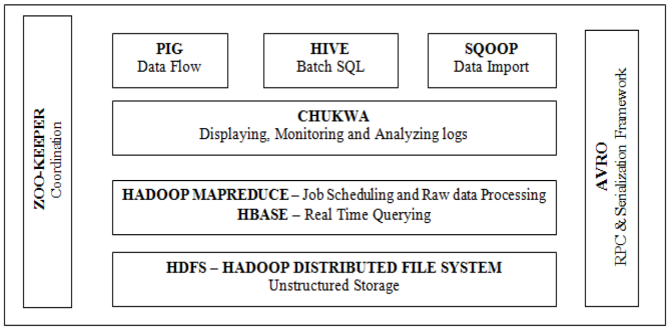
ZOO-KEEPER
Coordination
PIG
Data Flow
HIVE
Batch SQL
SQOOP
Data Import
CHUKWA
Displaying, Monitoring and Analyzing logs
HADOOP MAPREDUCE – Job Scheduling and Raw data Processing
HBASE – Real Time Querying
HDFS – HADOOP DISTRIBUTED FILE SYSTEM
Unstructured Storage
AVRO
RPC & Serialization Framework

Fig 1 Hadoop Framework

III. RUNNING A JOB IN HADOOP

There are three components in running a job:

- a user node,
- a JobTracker, and
- Several TaskTrackers.

The data flow starts by calling the runJob(conf) function inside a user program running on the user node, in which conf is an object containing some tuning parameters for the MapReduce framework and HDFS.

3.1 Job Submission:

Each job is submitted from a user node to the JobTracker node that might be situated in a different node within the cluster through the following procedure: A user node asks for a new job ID from the JobTracker and computes input file splits. The user node copies some resources, such as the job’s JAR file, configuration file, and computed input splits, to the JobTracker’s file system. The user node submits the job to the JobTracker by calling the submitJob() function.

3.2 Task assignment:

The JobTracker creates one map task for each computed input split by the user node and assigns the map tasks to the execution slots of the TaskTrackers. The JobTracker considers the localization of the data when assigning the map tasks to the TaskTrackers. The JobTracker also creates reduce tasks and assigns them to the TaskTrackers. The number of reduce tasks is predetermined by the user, and there is no locality consideration in assigning them.

3.3 Task execution

The control flow to execute a task (either map or reduce) starts inside the TaskTracker by copying the job JAR file to its file system. Instructions inside the job JAR file are executed after launching a Java Virtual Machine (JVM) to run its map or reduce task.

3.4 Task execution

The control flow to execute a task (either map or reduce) starts inside the TaskTracker by copying the job JAR file to its file system. Instructions inside the job JAR file are executed after launching a Java Virtual Machine (JVM) to run its map or reduce task.

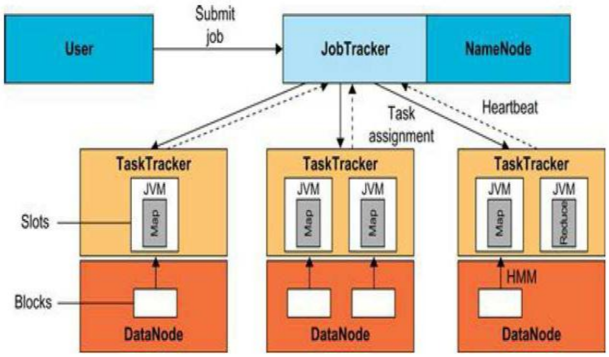

Fig 2 Running a Job in Hadoop

IV. HDFS ARCHITECTURE

Hadoop File System was developed using distributed file system design. Unlike other distributed systems, HDFS is high fault tolerant and designed using low-cost hardware.

HDFS holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing.

4.1 Namenode

The namenode is the commodity hardware that contains the GNU/Linux operating system and the namenode software. It is software that can be run on commodity hardware. The system having the namenode acts as the master server

4.2 Datanode

The datanode is a commodity hardware having the GNU/Linux operating system and datanode software. For every node (Commodity hardware/System) in a cluster, there will be a datanode. These nodes manage the data storage of their system. Datanodes perform read-write operations on the file systems, as per client request. They also perform operations such as block creation, deletion, and replication according to the instructions of the namenode.

4.3 Block

Generally the user data is stored in the files of HDFS. The file in a file system will be divided into one or more segments and/or stored in individual data nodes. These file segments are called as blocks. In other words, the minimum amount of data that HDFS can read or write is called a Block. The default block size is 64MB, but it can be increased as per the need to change in HDFS configuration.

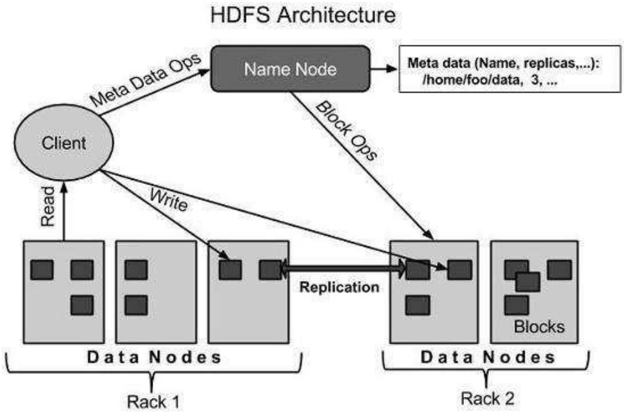

Fig 3 HDFS Architecture

V. CONCLUSION

In this paper, that describes what the Hadoop Framework components are and how to run a job and also describes the HDFS concepts. This survey is useful for study about the basic Hadoop concepts.

REFERENCES

[1] Kai Hwang, Geoffery C. Fox and Jack J. Dongarra, “Distributed and Cloud Computing: Clusters, Grids, Clouds and the Future of Internet”, First Edition, Morgan Kaufman Publisher, an Imprint of Elsevier, 2012.

[2] Tom White, “Hadoop The Definitive Guide” O’Reilly, 2009.

[3] Jason Venner, “Pro Hadoop- Build Scalable, Distributed Applications in the Cloud”, A Press, 2009

[4] http://www.tutorialspoint.com/Hadoop