**Session 1**

**Assignment 1.5**

Student Name: Karthik K

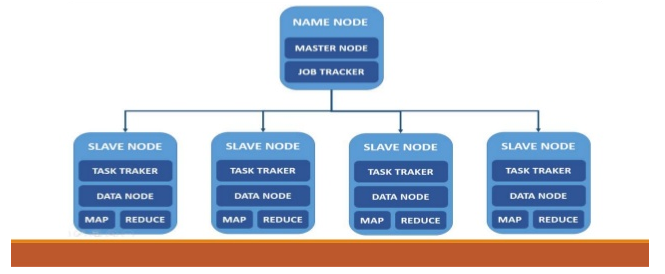
Course: Big Data Hadoop & Spark Training

**Problem Statement**: List the Components of Hadoop 2.x and explain each component in detail.

**Solution:**

# Hadoop 2.x Architecture:

Hadoop 2.x Architecture is designed completely different. It has added one new component: YARN and also updated HDFS and MapReduce component’s Responsibilities.



Hadoop Master/Slave Architecture

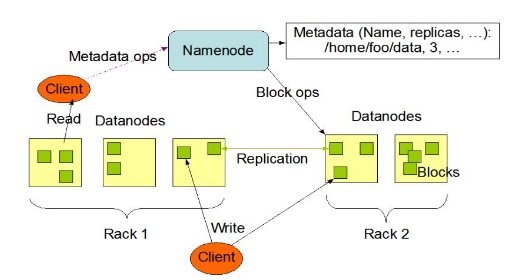
1. Hadoop 2.x Major Components:

Hadoop 2.x has the following three Major Components:

1. HDFS
2. YARN
3. MapReduce

HDFS:

The Hadoop Distributed File System (HDFS) is the primary storage system used by Hadoop applications. HDFS is a distributed file system that provides high-performance access to data across Hadoop clusters.



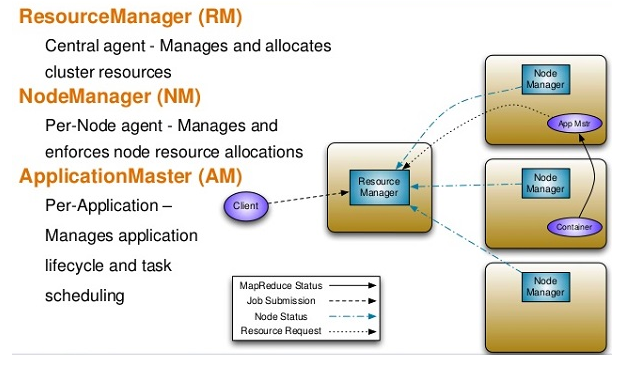
HDFS Architecture

* The file store in HDFS provides scalable, fault-tolerant storage at low cost.
* The HDFS software detects and compensates for hardware issues, including disk problems and server failure.
* HDFS stores file across the collection of servers in a cluster.
* Files are decomposed into the blocks and each block is written to more than one of the servers.

### YARN (Yet Another Resource Negotiator)

Apache **YARN** (**Yet another Resource Negotiator**) is Hadoop’s cluster resource management System. YARN was introduced in Hadoop 2 to improve the MapReduce implementation, but it is general enough to support other distributed computing paradigms as well.

YARN provides APIs for requesting and working with cluster resources, but these APIs are not typically used directly by user code. Instead, users write to higher-level APIs provided by distributed computing frameworks, which themselves are built on YARN and hide the resource management details from the user. The illustration shows some distributed computing frameworks (MapReduce, Spark, and so on) running as YARN applications on the cluster compute layer (YARN) and the cluster storage layer (HDFS and HBase).



Yarn Architecture

### MapReduce

**MapReduce** is a programming model for data processing. The model is simple, yet not too simple to express useful programs in. Hadoop can run MapReduce programs written in various languages; in this chapter, we look at the same program expressed in **Java, Ruby, and Python**. Most importantly, MapReduce programs are inherently parallel, thus putting very large-scale data analysis into the hands of anyone with enough machines at their disposal.

