



Session 03: Exploring MapReduce

Assignment 01 Question

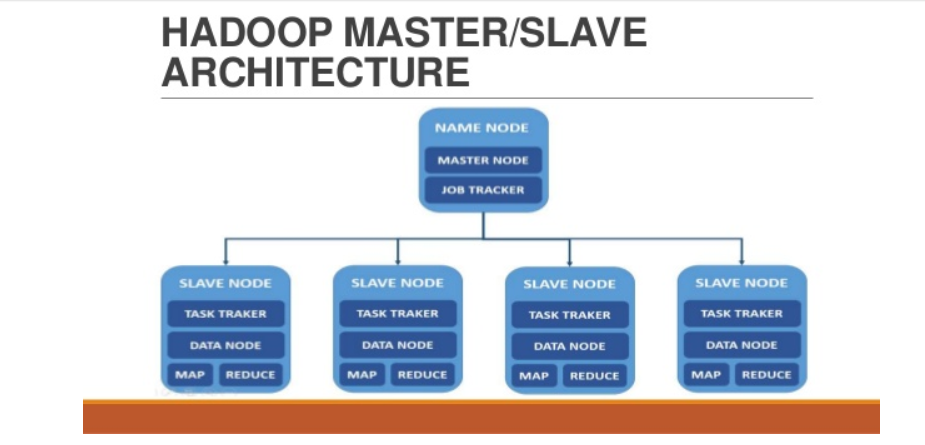
*List the Components of Hadoop 2.x and explain each component in detail*.

### **1) Hadoop Common-**

Apache Foundation has pre-defined set of utilities and libraries that can be used by other modules within the Hadoop ecosystem. For example, if HBase and Hive want to access HDFS they need to make of Java archives (JAR files) that are stored in Hadoop Common.

### **2) Hadoop Distributed File System (HDFS) -**

The default big data storage layer for Apache Hadoop is [HDFS](https://www.dezyre.com/hadoop-course/hdfs). HDFS is the “Secret Sauce” of Apache Hadoop components as users can dump huge datasets into HDFS and the data will sit there nicely until the user wants to leverage it for analysis. HDFS component creates several replicas of the data block to be distributed across different clusters for reliable and quick data access. HDFS comprises of 3 important components-NameNode, DataNode and Secondary Name Node. HDFS operates on a Master-Slave architecture model where the NameNode acts as the master node for keeping a track of the storage cluster and the DataNode acts as a slave node summing up to the various systems within a Hadoop cluster.



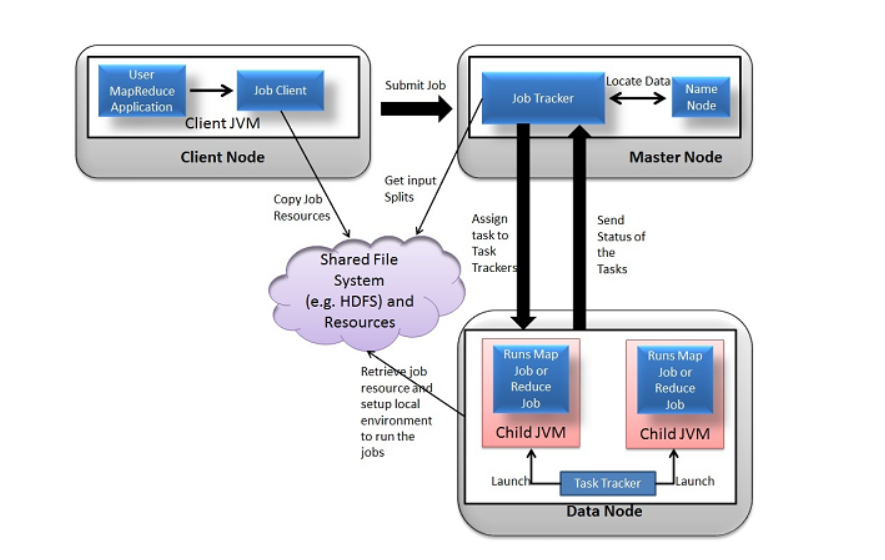
## **3) MapReduce- Distributed Data Processing Framework of Apache Hadoop**

[MapReduce](https://www.dezyre.com/hadoop-course/mapreduce) is a Java-based system created by Google where the actual data from the HDFS store gets processed efficiently. MapReduce breaks down a big data processing job into smaller tasks. [MapReduce](https://www.dezyre.com/article/hadoop-mapreduce-vs-apache-spark-who-wins-the-battle/83) is responsible for the analysing large datasets in parallel before reducing it to find the results. In the Hadoop ecosystem, Hadoop MapReduce is a framework based on YARN architecture. YARN based Hadoop architecture, supports parallel processing of huge data sets and MapReduce provides the framework for easily writing applications on thousands of nodes, considering fault and failure management.

The basic principle of operation behind MapReduce is that the “Map” job sends a query for processing to various nodes in a Hadoop cluster and the “Reduce” job collects all the results to output into a single value. Map Task in the Hadoop ecosystem takes input data and splits into independent chunks and output of this task will be the input for Reduce Task. In The same Hadoop ecosystem Reduce task combines Mapped data tuples into smaller set of tuples. Meanwhile, both input and output of tasks are stored in a file system. MapReduce takes care of scheduling jobs, monitoring jobs and re-executes the failed task.

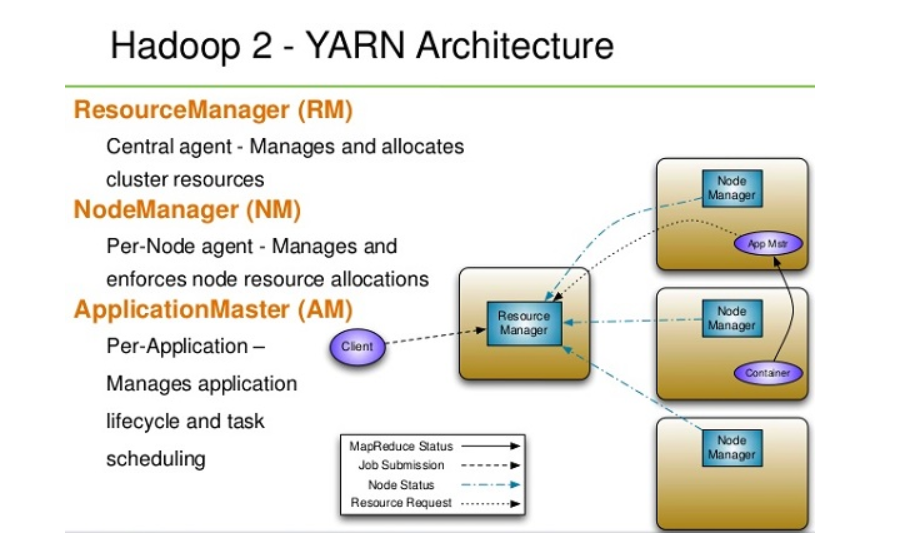
MapReduce framework forms the compute node while the HDFS file system forms the data node. Typically in the Hadoop ecosystem architecture both data node and compute node are considered to be the same.

The delegation tasks of the MapReduce component are tackled by two daemons- Job Tracker and Task Tracker as shown in the image below –



## **4)YARN**

[YARN](https://www.dezyre.com/article/hadoop-2-0-yarn-framework-the-gateway-to-easier-programming-for-hadoop-users/84) forms an integral part of Hadoop 2.0. YARN is great enabler for dynamic resource utilization on Hadoop framework as users can run various Hadoop applications without having to bother about increasing workloads.



### **Key Benefits of Hadoop 2.0 YARN Component-**

* It offers improved cluster utilization
* Highly scalable
* Beyond Java
* Novel programming models and services
* Agility