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

Hadoop 2.x contains 3 components mainly,

1. HDFS
2. YARN
3. MapReduce

1. **HDFS (Hadoop Distributed File System):** This is the default big data storage layer for Hadoop. HDFS is one of the key Hadoop components as users can dump huge datasets into HDFS and the data will reside there 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**: An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. Also it contains Hadoop File System Tree and other metadata information about files and directories.

Further a NameNode contains 2 components,

* + **fsimage (file system image):** It will have the entire directory structure of the HDFS and configuration of the replication level. It also contains the information of Modification and access time of the files, access permission of files and directories block size of the file.
  + **Edits:** When any write operation takes place in HDFS, the directory structure gets modified. These modifications are stored in memory as well as in edits files (edits files are stored on hard disk). If existing fsimage file gets merged with edits, we’ll get updated fsimage file. This process is called checkpointing and is carried out by Secondary NameNode. It takes fsimage and edits files from NameNode and returns updated fsimage file after merging.
* **DataNodes**: This Stores actual data blocks of file in HDFS on its own local disk and sends signals to NameNode periodically to verify whether it is active or not. Sends block reporting to the NameNode on cluster start up as well as periodically at every 10th Heartbeat. The data node is the workhorse of the system. They perform all the block operation including periodic checksum. They receive instructions from the name node of where to put the blocks and how to put the blocks.
* **Secondary NameNode**: Performs house-keeping activities for NameNode, like periodic merging of namespace and edits.

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.

2. **YARN (MRv2 – MapReduce Version 2):** This new architecture introduced in hadoop-0.23, divides the two major functions of the JobTracker: resource management and job life-cycle management into separate components. The new Resource Manager manages the global assignment of compute resources to applications and the per-application Application Master manages the application’s scheduling and coordination. An application is either a single job in the sense of classic MapReduce jobs or a DAG of such jobs. The Resource Manager and per-machine Node Manager Daemon, which manages the user processes on that machine, form the computation fabric. The per-application Application Master is, in effect, a framework specific library and is tasked with negotiating resources from the Resource Manager and working with the Node Manager(s) to execute and monitor the tasks.

3. **MapReduce:** This 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 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.