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

The 3 most important components of Hadoop 2.x are as explained below:

1. **Hadoop File Distributed System (HDFS)** is a primary storage system used by Hadoop applications. HDFS is a distributed file system that provides high performance access to data across Hadoop clusters. HDFS is built to support applications with large data sets, including individual files that reach into the terabytes. It uses a *master/slave* architecture, with each cluster consisting of a single *NameNode* that manages file system operations and supporting *DataNodes* that manage data storage on individual compute nodes.

Because HDFS typically is deployed on low-cost commodity Hardware, server failures are common. The file system is designed to be highly *fault-tolerant*, however, but facilitating the rapid transfer of data between compute nodes and enabling Hadoop systems to continue running if a node fails.

When HDFS takes in data, it breaks the information down into separate pieces and distributes them to different nodes in a cluster, allowing for parallel processing. The file system also copies each piece of data multiple times and distributes the copies to individual nodes, placing at least one copy on a different server rack than the others. As a result, the data on nodes that crash can be found elsewhere within a cluster, which allows processing to continue while the failure is resolved.

1. **MapReduce** is a software framework for easily writing applications which process vast amounts of data in parallel on large clusters of commodity hardware in a reliable, fault-tolerant manner. Though MapReduce is supposed to be replaced with YARN/MapReduce 2, it still is an important component of Hadoop.

The term MapReduce refers to two separate and distinct tasks that Hadoop programs perform. The first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs).

The reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples. As the sequence of name MapReduce implies, the reduce job is always performed after the map job

1. **Yet Another Resource Negotiator (YARN)** is a cluster management technology. Sometimes called MapReduce 2.0, YARN is a software rewrite that decouples MapReduce’s resource management and scheduling capabilities from the data processing component, enabling Hadoop to support more varied processing approaches and a broader array of applications.

For example, Hadoop clusters can now run interactive querying and streaming data applications simultaneously with MapReduce batch jobs. The original incarnation of Hadoop closely paired the HDFS with the batch oriented MapReduce programming framework, which handles resource management and job scheduling on Hadoop systems and supports the parsing and condensing of data sets in parallel.

YARN combines a central resource manager that reconciles the way applications use Hadoop system resources with node manager agents that monitor the processing operations of individual cluster nodes. Running on commodity hardware clusters, Hadoop has attracted particular interest as a staging area and data store for large volumes of structured and unstructured data intended for use in analytics applications.

Separating HDFS from MapReduce with YARN makes the Hadoop environment more suitable for operational applications that can’t wait for batch jobs to finish.