## **Hadoop Architecture**

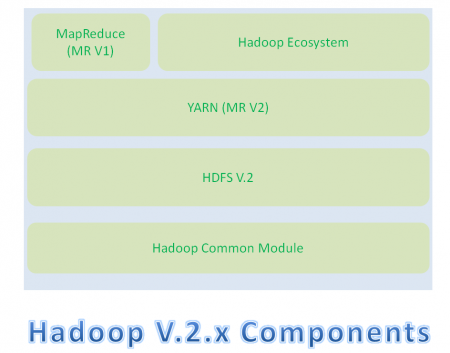
Following is the Apache Hadoop 2.x Architecture and How it’s components work in detail.

### **Contents**

* Hadoop 2.x Architecture
* Hadoop 2.x Major Components
* How Hadoop 2.x Major Components Works

### **Hadoop 2.x Architecture**

Apache Hadoop 2.x or later versions are using the following Hadoop Architecture. It is a Hadoop 2.x High-level Architecture. We will discuss in-detailed Low-level Architecture in coming sections.

[](https://cdn.journaldev.com/wp-content/uploads/2015/08/hadoop2.x-components.png)

* Hadoop Common Module is a Hadoop Base API (A Jar file) for all Hadoop Components. All other components works on top of this module.
* HDFS stands for Hadoop Distributed File System. It is also know as HDFS V2 as it is part of Hadoop 2.x with some enhanced features. It is used as a Distributed Storage System in Hadoop Architecture.
* YARN stands for Yet Another Resource Negotiator. It is new Component in Hadoop 2.x Architecture. It is also know as “MR V2”.
* MapReduce is a Batch Processing or Distributed Data Processing Module. It is also know as “MR V1” as it is part of Hadoop 1.x with some updated features.
* Remaining all Hadoop Ecosystem components work on top of these three major components: HDFS, YARN and MapReduce. We will discuss all Hadoop Ecosystem components in-detail in my coming posts.

When compared to Hadoop 1.x, 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 2.x Major Components**

Hadoop 2.x has the following three Major Components:

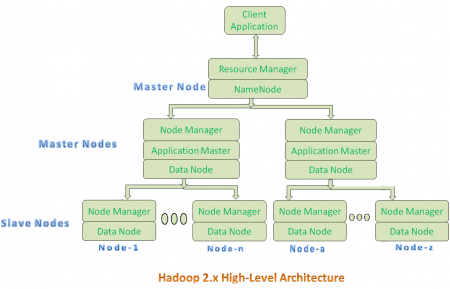
* HDFS
* YARN
* MapReduce

These three are also known as Three Pillars of Hadoop 2. Here major key component change is YARN. It is really game changing component in BigData Hadoop System.

### **How Hadoop 2.x Major Components Works**

Hadoop 2.x components follow this architecture to interact each other and to work parallel in a reliable, highly available and fault-tolerant manner.

****Hadoop 2.x Components High-Level Architecture****

[](https://cdn.journaldev.com/wp-content/uploads/2015/08/hadoop2.x-highlevel-architecture.png)

* All Master Nodes and Slave Nodes contains both MapReduce and HDFS Components.
* One Master Node has two components:
  1. Resource Manager(YARN or MapReduce v2)
  2. HDFS

It’s HDFS component is also knows as NameNode. It’s NameNode is used to store Meta Data.

* In Hadoop 2.x, some more Nodes acts as Master Nodes as shown in the above diagram. Each this 2nd level Master Node has 3 components:
  1. Node Manager
  2. Application Master
  3. Data Node
* Each this 2nd level Master Node again contains one or more Slave Nodes as shown in the above diagram.
* These Slave Nodes have two components:
  1. Node Manager
  2. HDFS

It’s HDFS component is also knows as Data Node. It’s Data Node component is used to store actual our application Big Data. These nodes does not contain Application Master component.

****Hadoop 2.x Components In-detail Architecture****

****Hadoop 2.x Architecture Description****

****Resource Manager:****

* Resource Manager is a Per-Cluster Level Component.
* Resource Manager is again divided into two components:
  1. Scheduler
  2. Application Manager
* Resource Manager’s Scheduler is :
  1. Responsible to schedule required resources to Applications (that is Per-Application Master).
  2. It does only scheduling.
  3. It does care about monitoring or tracking of those Applications.

****Application Master:****

* Application Master is a per-application level component. It is responsible for:
  1. Managing assigned Application Life cycle.
  2. It interacts with both Resource Manager’s Scheduler and Node Manager
  3. It interacts with Scheduler to acquire required resources.
  4. It interacts with Node Manager to execute assigned tasks and monitor those task’s status.

****Node Manager:****

* Node Manager is a Per-Node Level component.
* It is responsible for:
  1. Managing the life-cycle of the Container.
  2. Monitoring each Container’s Resources utilization.

****Container:****

* Each Master Node or Slave Node contains set of Containers. In this diagram, Main Node’s Name Node is not showing the Containers. However, it also contains a set of Containers.
* Container is a portion of Memory in HDFS (Either Name Node or Data Node).
* In Hadoop 2.x, Container is similar to Data Slots in Hadoop 1.x.