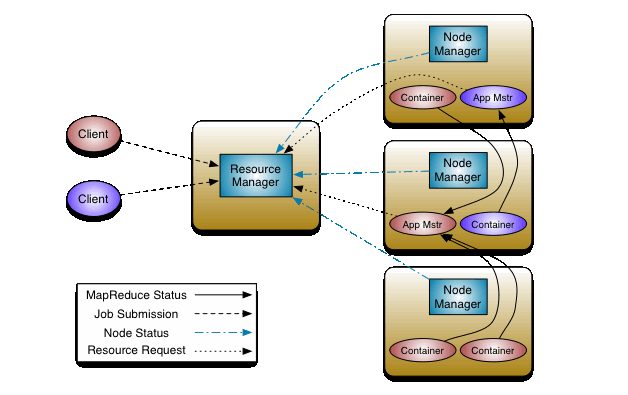
**Explain in brief the architecture of Apache Hadoop Yarn.**

**Ans :** 

The basic work of YARN is to seperate the functionalities of resource management and job scheduling into separate daemons. The concept is to have a global Resource Manager (*RM*) and per-application Application Master (*AM*). An application is either a single job or a DAG of jobs.

**YARN Infrastructure** (Yet Another Resource Negotiator) is the framework responsible for providing the computational resources (e.g., CPUs, memory, etc.) needed for application executions.

Two important elements are:

The **Resource Manager** (one per cluster) is the master. It knows where the slaves are located (Rack Awareness) and how many resources they have.

It runs several services, the most important is the **Resource Scheduler** which decides how to assign the resources.

The **Node Manager** (many per cluster) is the slave of the infrastructure. When it starts, it announces himself to the Resource Manager. Periodically, it sends an heartbeat to the Resource Manager. Each Node Manager offers some resources to the cluster. Its resource capacity is the amount of memory and the number of vcores. At run-time, the Resource Scheduler will decide how to use this capacity: a **Container** is a fraction of the NM capacity and it is used by the client for running a program.

In YARN, there are at least three actors:

* the **Job Submitter** (the client)
* the **Resource Manager** (the master)
* the **Node Manager** (the slave)

The application startup process is the following:

1. a client submits an application to the Resource Manager
2. the Resource Manager allocates a container
3. the Resource Manager contacts the related Node Manager
4. the Node Manager launches the container
5. the Container executes the **Application Master**

The Application Master is responsible for the execution of a single application. It asks for containers to the Resource Scheduler (Resource Manager) and executes specific programs (e.g., the main of a Java class) on the obtained containers. The Application Master knows the application logic and thus it is framework-specific. The MapReduce framework provides its own implementation of an Application Master.

The Resource Manager is a single point of failure in YARN. Using Application Masters, YARN is spreading over the cluster the metadata related to running applications. This reduces the load of the Resource Manager and makes it fast recoverable.