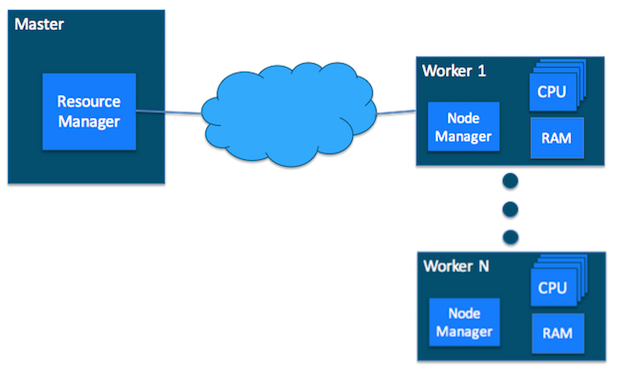
Explain in brief the architecture of Apache Hadoop Yarn.

Ans:



The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.

The idea is to have a global ResourceManager (*RM*) and per-application ApplicationMaster (*AM*). An application is either a single job or a DAG of jobs.

The ResourceManager and the NodeManager form the data-computation framework.

The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system.

 The NodeManager is the per-machine framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler.

The per-application ApplicationMaster is, in effect, a framework specific library and is tasked with negotiating resources from the ResourceManager and working with the NodeManager(s) to execute and monitor the tasks.

The ResourceManager has two main components:

1) Scheduler

2) ApplicationsManager

**Scheduler:**

-The Scheduler is responsible for allocating resources to the various running applications subject to familiar constraints of capacities, queues etc.

-The Scheduler is pure scheduler in the sense that it performs no monitoring or tracking of status for the application.

-Also, it offers no guarantees about restarting failed tasks either due to application failure or hardware failures.

-The Scheduler performs its scheduling function based the resource requirements of the applications; it does so based on the abstract notion of a resource *Container* which incorporates elements such as memory, cpu, disk, network etc.

**Application Manager:**

-The ApplicationsManager is responsible for accepting job-submissions, negotiating the first container for executing the application specific ApplicationMaster and provides the service for restarting the ApplicationMaster container on failure.

-The per-application ApplicationMaster has the responsibility of negotiating appropriate resource containers from the Scheduler, tracking their status and monitoring for progress.