Hadoop 2.x components

**Components of YARN:**

1. Global Resource Manager

• Assigns resources among applications for optimal resource utilization.

• One cluster has one instance of Resource Manager.

2. Node Manager

• Runs on each node and communicates with Resource Manager about resource

usage on the machine.

• It receives requests from resource manager about resource allocation to jobs

and maintains life cycle of containers.

3. Application-specific Application Master

• It is the actual instance which does processing.

• It requests Resource Manager for resources and works with NodeManager to get

those resources for task execution. Application Master could be MapReduce or any

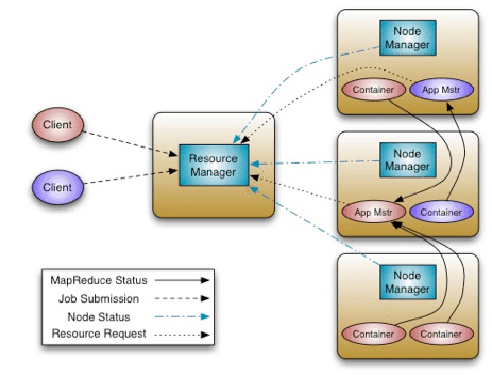
other processing framework.

4. Scheduler

⦁ It is plugged with Resource Manager to help in resource allocation. Different schedulers allocate resources using different algorithms.

5. Container

⦁ Resource is handled by Resource Manager and Node Manager.

⦁ Processing is handled by Application Master (MapReduce is one of the many possible types of Application Master). So, processing other than MapReduce is also possible.

**Job submission and flow:**

Step 1: Job/Application (which can be MapReduce, Java/Scala Application, DAG jobs like Apache Spark etc.) is submitted by the YARN client application to the ResourceManager daemon along with the command to start the ApplicationMaster on any container at NodeManager.

Step 2: Application Manager process on Master Node validates the job submission request and hands it over to Scheduler process for resource allocation.

Step 3: Scheduler process assigns a container for ApplicationMaster on one slave node.

Step 4: NodeManager daemon starts the ApplicationMaster service within one of its container using the command mentioned in Step 1, hence ApplicationMaster is considered to be the first container of any application

Step 5: ApplicationMaster negotiates the other containers from ResourceManager by providing the details like location of data on slave nodes, required cpu, memory, cores etc.

Step 6: Resource Manager allocates the best suitable resources on slave nodes and responds to ApplicationMaster with node details and other details

Step 7: Then, ApplicationMaster send requests to Node Managers on suggested slave nodes to start the containers

Step 8: ApplicationMaster then manages the resources of requested containers while job execution and notifies the ResourceManager when execution is completed

Step 9: NodeManagers periodically notify the ResourceManager with the current status of available resources on the node as to what information can be used by scheduler to schedule new application on the clusters

Step 10: In case of any failure of slave node, ResourceManager will try to allocate new container on other best suitable node so that ApplicationMaster can complete the process using new container.