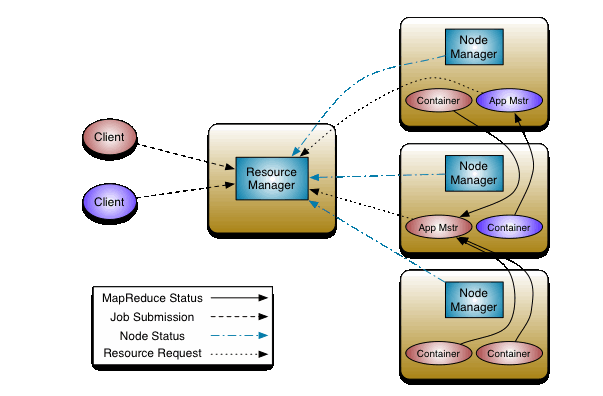
**#Assignment\_8.3**

**Q1- Explain in brief the architecture of Apache Hadoop Yarn.**

Answer:

Yarn (yet another resource negotiator) is the framework which is responsible for computational resource assignment to execute certain application. The component and architecture is explained below-



**Components of YARN:-**

**1. Node Manager:-**

Node manager runs on each of the nodes and continuously communicate with resource manager about usage on the machine. The request is received from resource manager for allocation of resource to jobs and to maintain life cycle of containers.

1. **Global Resource Manager:-**

It is the rack aware master node present in yarn. It Assigns resources among applications for optimal resource utilization. Every cluster has one instance of global resource manger

1. **Application-specific Application Master:-**

For each application running on Hadoop has its own actual instance which does processing. It periodically send heartbeat to resource manager. 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:**-

Schedulers are used in resource allocation. It is done by allocating different containers. Different schedulers allocate resources using different algorithms. It does not track or monitor the activity, so it doesn’t guarantee that the task failed will restart.

**5. Container:-**

It is a set of allocated system resources like CPU core and memory. Containers are allocated while NodeManager manage them. Containers are used by tasks.

**Flow of Execution of application in YARN-**

1. MapReduce application is submitted by client to resource manager that also contains information to launch application master.
2. A container is negotiated by resource manager for application master and then it is launched. It allow direct interface with the client.
3. Container launch specification is provided to node manager by application master which launches container for application.
4. During the execution application status and progress provided by application master to client.
5. Upon complete execution and completion, application master deregisters with resource manager and shuts down by returning its containers to return pool.