**Assignment 8.3**

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

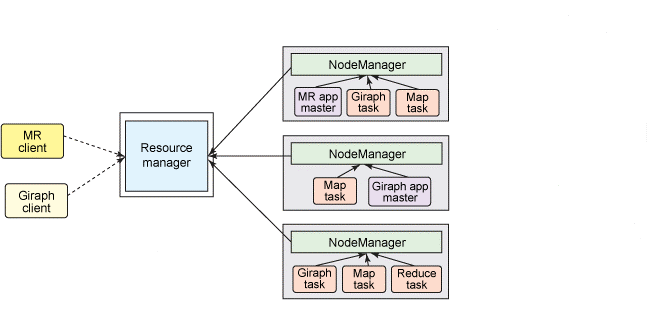
***Client:***Can submit any type of application supported by YARN.

Resource Manager (RM):

* Keeps track of live Node Managers and available resources.
* Allocates available resources to appropriate applications and tasks.
* Monitors application masters.

Node Manager (NM):

* Provides computational resources in form of containers.
* Managers processes running in containers.



Application Master (AM):

* Coordinates the execution of all tasks within its application.
* Asks for appropriate resource containers to run tasks.

Containers:

* Can run different types of tasks (also application masters).
* Has different sizes e.g. RAM, CPU.
* In the YARN architecture, a global ResourceManager runs as a master daemon, that arbitrates the available cluster resources among various competing applications. The ResourceManager tracks how many live nodes and resources are available on the cluster and coordinates what applications submitted by users should get these resources and when. The ResourceManager is the single process that has this information so it can make its allocation (or rather, scheduling) decisions in a shared, secure, and multi-tenant manner.
* ApplicationMaster is started to coordinate the execution of all tasks within the application. This includes monitoring tasks, restarting failed tasks, speculatively running slow tasks, and calculating total values of application counters. These responsibilities were previously assigned to the single JobTracker for all jobs. The ApplicationMaster and tasks that belong to its application run in resource containers controlled by the NodeManagers.
* The NodeManager is a more generic and efficient version of the TaskTracker. Instead of having a fixed number of map and reduce slots, the NodeManager has a number of dynamically created resource containers. The size of a container depends upon the amount of resources it contains, such as memory, CPU, disk, and network IO. The number of containers on a node is a product of configuration parameters and the total amount of node resources (such as total CPUs and total memory) outside the resources dedicated to the slave daemons and the OS.