**How YARN Works**

YARN’s original purpose was **to split up the two major responsibilities** of the JobTracker/TaskTracker into separate entities:

* a global Resource Manager
* a per-application Application Master
* a per-node slave Node Manager
* a per-application Container running on a Node Manager

The **ResourceManager** and the **NodeManager** formed the new generic system for managing applications in a distributed manner. The ResourceManager is the ultimate authority that arbitrates resources among all applications in the system. The **ApplicationMaster** is a framework-specific entity that negotiates resources from the ResourceManager and works with the NodeManager(s) to execute and monitor the component tasks.

The ResourceManager has a scheduler, which is responsible for allocating resources to the various applications running in the cluster, according to constraints such as queue capacities and user limits. The scheduler schedules based on the resource requirements of each application.

Each ApplicationMaster has responsibility for negotiating appropriate resource containers from the scheduler, tracking their status, and monitoring their progress. From the system perspective, the ApplicationMaster runs as a normal container.

The NodeManager is the per-machine slave, which is responsible for launching the applications’ containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager