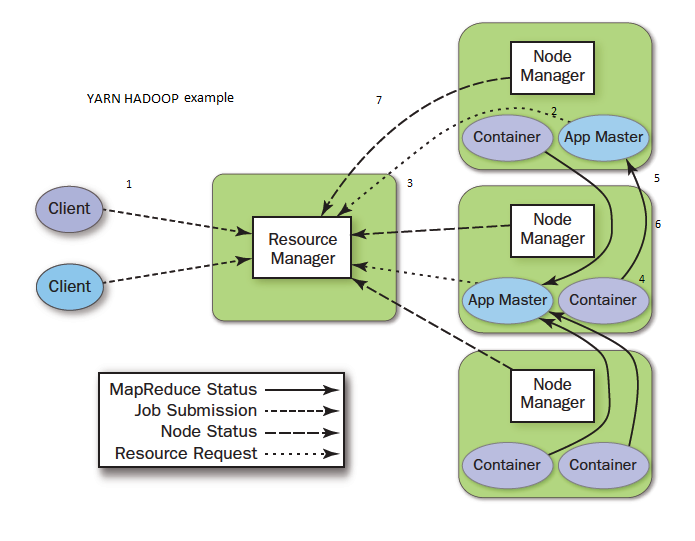
Yarn (Yet Another Resource Negotiator) is an application framework introduced in Hadoop 2.0 to overcome the problems of Hadoop 1.0 MapReduce which were Scalability, Availability, resource utilization, and support of alternative paradigms and programming environments. YARN components are Resource manager, Application Master, and Node Manager. Resource manager allocates the resources to the applications based on the applications needs. YARN uses a separate Resource manager to schedule and manage all their jobs in the cluster. The Application master component request resources from the resource manager and run/monitor the tasks. The node manager component monitors the resources usage and report back to the resource manager. All nodes are managed by a node manager working together with the resource manager. The job resources are assigned to different containers (computing resource that contains processing node and a memory). The resource manager and the node manager have no information about the jobs, they only manage the containers that are running on the clusters. Each application must start an application master to manage the actual task for the job. The application master is run by the container scheduled by the resource manager and managed by a node manager. To run an actual application, the application master request additional containers (resources) from the resource manager. These containers are where the actual work will be done for the cluster. The relationship between the resource manager and the application master can be dynamic meaning that containers can be allocated or deallocated dynamically during run time. Here is an example of flow of the MapReduce job in Hadoop 2.0. 

1. Client submits a job to the resource manager.
2. The resource manager selects a node and instructs its node manager to run the application master.
3. The application master will request additional containers from the resource manager
4. The assigned containers are started and managed in the appropriate nodes by the node manager
5. Once the application master and the containers are connected the resource manager and the node manager step away from the job
6. The job progress is reported back to the application master
7. When a task that runs in container is completed. The node manager makes the container available to the resource manager to be used by another job.

**HADOOP 2.0 ECOSYSTEM**

Hadoop 2.0 ecosystem consists of the following main components:

1. Core components

**HDFS**

**YARN**

**MAPREDUCE**

1. Hadoop Database

**Apache HCatalog:** table and storage management service for data created using Hadoop. The table abstraction removes the need for user to know where data is stored.

**Apache HBase:** is Hadoop database, designed for hosting large tables with billions of rows and columns, its non-relational database

1. MapReduce Query Tools

**Apache Pig:** high level language enables programmer to write complex MapReduce transformation using simple scripts.

**Apache Hive:** provides database query interface to Hadoop

1. Data import and export

**Apache sqoop:** tools for transferring large data between HDFS and relational database

**Apache Flume:** service for collecting, aggregating, and moving large amount of log data.

**Apache Avro:** is serialization format that makes data exchange possible between programs written in any language.

1. Workflow automation

**Apache Oozie:** service for scheduling apache Hadoop jobs.

**Apache Falcon:** enables automation of data movement and processing for replication operations and some others. Falcon triggers a job start when data changes or new data becomes available.

1. Administration

**Apache Ambari:** web based tool for managing and monitoring Apache Hadoop Clusters using GUI (Graphical User Interface).

1. YARN Application FrameWorks

Applications written specifically for the YARN environment. Like MapReduce, Apache Giraph (for graph processing), Spark (for memory processing), and others.

1. Other

**Apache ZooKeeper**

service for maintaining configuration, health, and status elements on and between the nodes. It maintains common objects needed in the cluster environment such as configurations information, naming space and so on. Provides application reliability if one application master dies zookeeper will spawn new application master to resume the tasks.

* **Apache Mahout:** machine learning library implements many approaches to machine learning.

This figure shows an overview of the Ecosystem of Hadoop 2.0.

