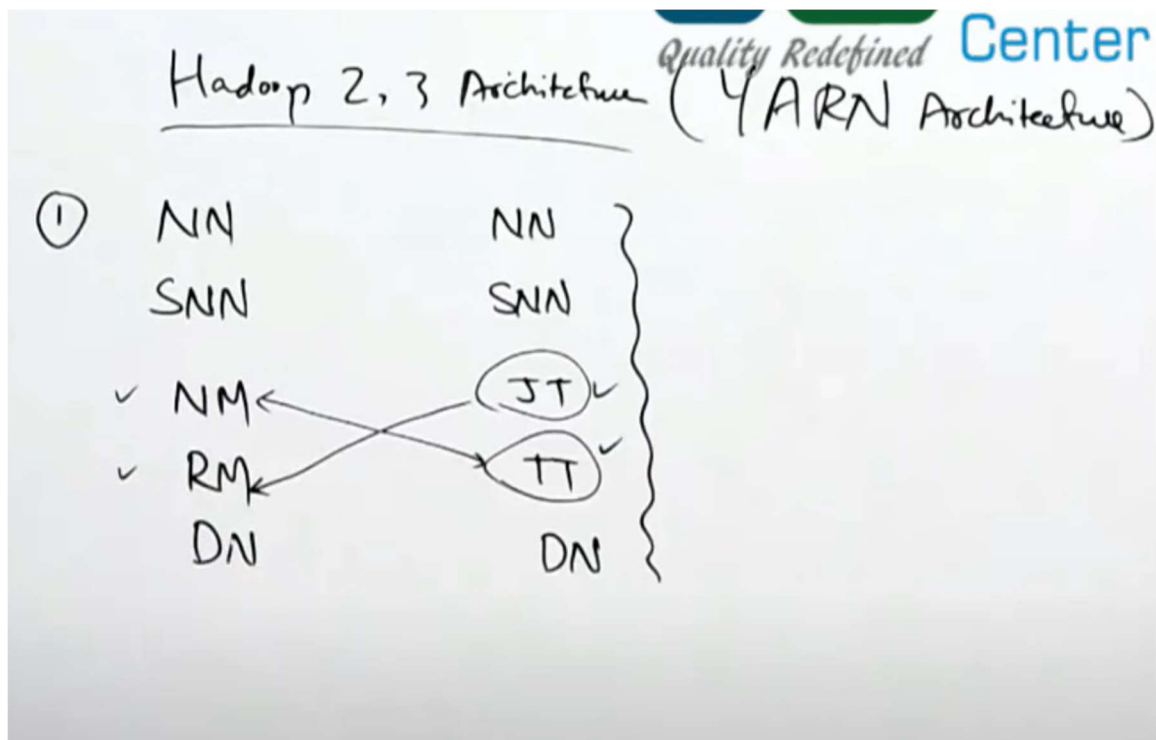
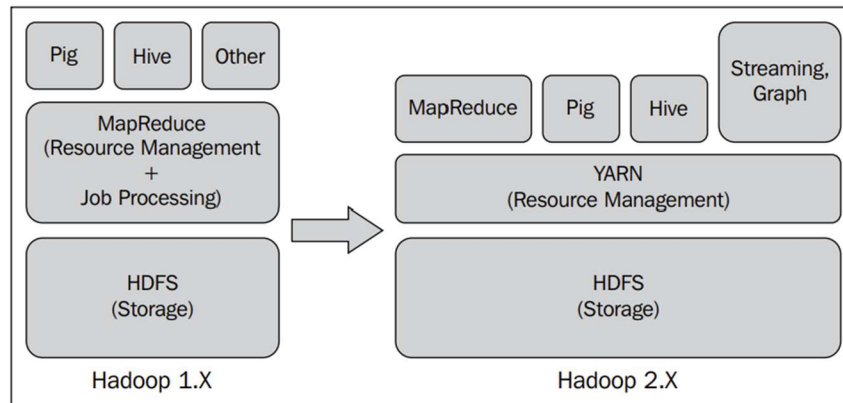
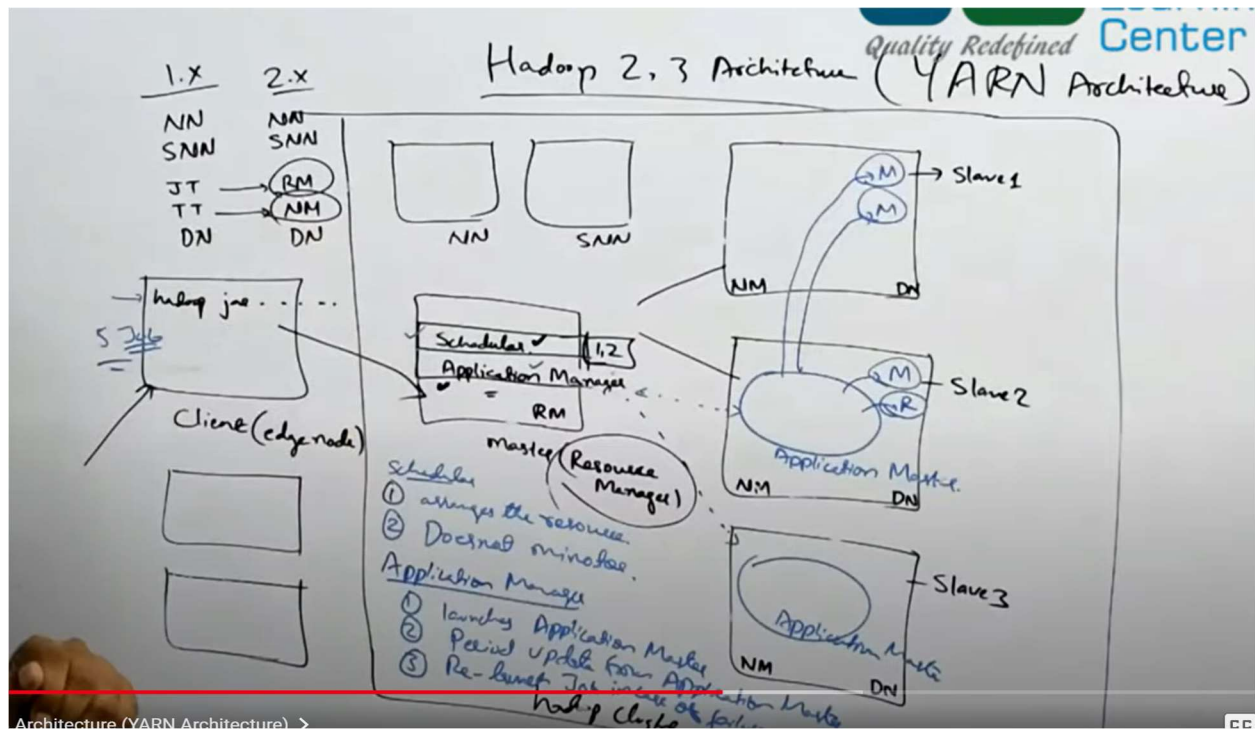


## Hadoop 2.x,3.x Architecture



## Resource Managing Level Enhancements



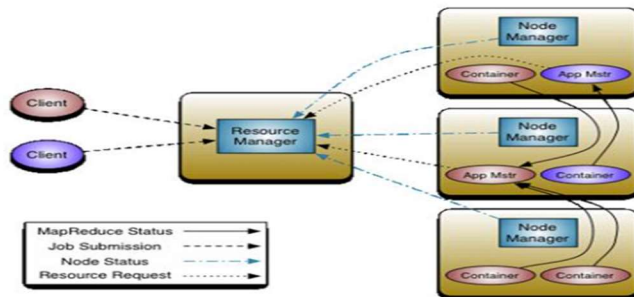
### Hadoop 2 and 3 Architecture (Yarn)

1. Namenode
2. Secondary Name Node
3. Resource Manager (instead of Job Tracker)
4. Data Node
5. Node Manager (instead of Task Tracker)

1-3 are master Daemons, 4-5 are slave daemons

In Hadoop, 1, Map-reduce used to act as a resource negotiator.

In Hadoop 2, We have yarn taking care of it.



The above image is copied from apache docs and are copyrighted to them.

### Resource Manager:

1. Installed as a master daemon, it helps in managing resources across all the job.
2. It comes with 2 different components
  - a. Scheduler  
It helps to allocate resources to the application. The scheduler is on top of the progress and does not care if the job fails due to any errors.
  - b. Applications Manager  
It launches application masters in the slave nodes and monitors the progress of application master which is internally monitoring the progress of the running application.

In case the Job fails in that particular node. Applications Manager will launch another application Master in some other node where the job will be restarted again.

The Applications Manager is responsible for accepting job-submissions, negotiating the first container for executing the application specific Application Master and provides the service for restarting the Applications Master container on failure.

#### **Node Manager**

1. Installed as Slave Daemon. It manages and monitors the containers and reports the resources used by containers to Resource Manager. Node manager also tracks the health of the node and periodically reports that to the Resource Manager.

#### **Application Master**

For each application, there would be 1 application master though out the cluster. It negotiates the resources from the Resource Manager and use that to run the job. The job is executed at the container under the guidance of Node manager. Application master talks to Resource Manager, takes the permission to launch the container. Once done, it then talks to Node Manager to start the application task and keeps updating the Node Manager.

The per-application Application Master has the responsibility of negotiating appropriate resource containers from the Scheduler, tracking their status and monitoring for progress.

#### **High Availability of Resource Manager**

Before hadoop 2.4, RM was a single point of failure. Since 2.4 now we can have multiple RM, with 1 active and other standby. All the RM has to be configured in zookeeper so that it can make the Rm highly available.

RM failover can be achieved either by manual process or automatic process.

For more info, Please refer apache hadoop documentation.

<https://hadoop.apache.org/docs/r3.1.0/index.html>

#### **High Availability of Name Node**

Hadoop 2 can have many Name nodes out of which 1 would be active name node and others will be standby name node.

The active name node will handle all client operations.

The standby name node will be ready to serve as active namenode if there are any failover.

It can use a shared location/in network drive to maintain all the metadata so that if namenode fails then standby name node can read the metadata from stand by name node).

Or There is also a concept called Journal Node, JN are usually a set of minimum 3 nodes which will be used to store the edit logs of namenode, such that at any time the active namenode fails, the standby can take over it.

When active name node performs any operation, it writes the changes in all the edit logs of all the journal node.

All data node should also be configured to send all the block information to all the name nodes so that both the name node maintains the same consistent copy.

## Storage Level Enhancements

