**ACD\_BDD\_Session\_5\_Assignment\_2\_Main**

**Problem Statement**

1. Explain the importance of below 4 demons in job execution with minimum of 5 points

* Name node
* Data node
* Resource Manager
* Node manager

Note: Forward the solutions for review

**Solution**

**Name node:**

1. Name node is the centerpiece, also known as the master
2. The namenode manages the filesystem namespace. It contains the filesystem tree and the metadata for all the files and directories in the tree.
3. The Name node knows the Data node on which all the jobs for a given file are located; however it does not store block locations persistently, because this information is reconstructed from the datanode when the system starts.
4. Without the namenode, the file system cannot be used. In fact, if the machine running the namenode was obliterated, all the files on the file system would be lost since there would be no way of how to reconstruct the files from the blocks on the datanodes. For this reason, it is important to make the namenode resilient to failure.
5. Name node is so critical to HDFS and when the Name node is down, HDFS/Hadoop cluster is inaccessible and considered down. Thus it is said to be single point of failure.

**Data node:**

1. Datanodes are the workhorses of the filesystem.
2. They store and retrieve blocks when they are told to (by clients or the namenode), and they report back to the namenode periodically with lists of blocks that they are storing.
3. On startup, a datanode connects to the namenode; spinning until that service comes up. It then responds to requests from the namenode for filesystem operations.
4. Client applications can talk directly to a datanode, once the namenode has provided the location of the data. Similarly, MapReduce operations farmed out to TaskTracker instances near a datanode, talk directly to the datanode to access the files. TaskTracker instances can, indeed should, be deployed on the same servers that host datanode instances, so that MapReduce operations are performed close to the data.
5. Datanode instances can talk to each other, which is what they do when they are replicating data.

**Resource manager:**

1. The ResourceManager is the ultimate authority that arbitrates resources among all the applications in the system.
2. The ResourceManager has two main components: Scheduler and ApplicationsManager.
3. The Scheduler is responsible for allocating resources to the various running applications subject to familiar constraints of capacities, queues etc. The Scheduler is pure scheduler in the sense that it performs no monitoring or tracking of status for the application. Also, it offers no guarantees about restarting failed tasks either due to application failure or hardware failures.
4. The Scheduler performs its scheduling function based on the resource requirements of the applications; it does so based on the abstract notion of a resource Container which incorporates elements such as memory, cpu, disk, network etc.
5. The ApplicationsManager is responsible for accepting job-submissions, negotiating the first container for executing the application specific ApplicationMaster and provides the service for restarting the ApplicationMaster container on failure. The per-application ApplicationMaster has the responsibility of negotiating appropriate resource containers from the Scheduler, tracking their status and monitoring for progress.

**Node manager:**

1. The NodeManager is the slave for the ResourceManager.
2. It is a per-machine framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network). Container is a fraction of the NodeManager capacity and it is used by the client for running a program.
3. Each Node Manager tracks the available data processing resources on its slave node and sends regular reports to the Resource Manager.
4. Each Node Manager offers some resources to the cluster. Its resource capacity is the amount of memory and the number of vcores.