1. Explain the core changes made in Hadoop 2.x:

There is a disadvantage in Hadoop 1.x which is related to the Name Node (i.e) Single point Failure (which means that when there is a name node failure there will be some data lost instead having a secondary node for updation from the name node to the secondary node on timely basis)

There is a solution for high availability node in Hadoop 2.x .And there are two types of name node than secondary name node which are

1.Active Name Node

2.Standby Name Node

Data node will send reports regarding blocks to Active Name Node as well Standby Name Node. If in case when the active Name Node get failed or get crashed then Stand By Name node will become active .

For example, there are 3 journal nodes,

And the formula to calculate quorom =(no of journal node + 2)/2(suppose 3 journal node (3+2)/2 =2)).

And the expected QUOROM is 2.When there is a failure in active name node then the count of quorom <2 in parallel the journal node switch to Stand by name node as active name node and at the time when failed name node receives it then it will become standby and the process will continue.

HADOOP FEDERATION

There will be one name node in Hadoop 1.x version whose memory is 64 gB which got a capacity to maintain 4000 data nodes of cluster. As the data income is more we might be in need of more data nodes say 10,000 data nodes.

This means that more memory is needed as well single name node scaleup above 64GB memory but this will peak the cost so we go to Hadoop 2.x where we can use FEDERATION which is a highlighted feature in which multiple name node (for eg 3) can be used instead of single name node (which were active name node ,passive name node and secondary name node)

METADATA will be maintained for each name node (each name node will take response for a specific department) but it is capable to access all the data nodes.

**3.Dynamic Node Allocation**: the node is not fixed for any type of map or reduce process but it can act as a mapper or reducer.

The main idea is to split the functionalities into two:

1. Job Tracker

2. Resource management

3. Job scheduling/monitoring divided into daemons.

4. YARN (Yet Another Resource Negotiator)

The purpose of yarn is to reduce the load of JOB TRACKER, Yarn is a cluster resource management system which has

1. Resource Manager-Node allocation to the clients.

2. Application Manager- for accepting job-submissions, Application Master as well for restarting the Application Master container on failure.

3. Application Master- it takes care for accepting job-submissions, negotiating the first container for executing the application specific Application Master and provides the service for restarting the Application Master container on failure.

4. Container -Environment supporting to run the job.

5. Scheduler-It takes care of allocating resources for various applications. The Scheduler will not perform monitoring or tracking the status of any of the application. also will not offer any guarantees on restarting failed tasks because of application failure or hardware failure.it will perform the scheduled function based on required resource. And in the first version, only memory is supported.

QN2.● Explain the difference between MapReduce 1 and MapReduce 2 / Yarn REFER word file attached

|  |  |
| --- | --- |
| HADOOP 1.X | HADOOP 2.X |
| Rresource utilization and Job Monitoring will be followed by job tracker | Resource Management and Node Management where followed by Resource Utilization since job Monitoring is taken care by application master |
| 4000 nodes and 40000 tasks will be configured by Task Tracker. | Scale up to 10000 nodes and 10000 tasks |
| Static slots will be configured by Task Tracker which means reducer task slots cannot run a Map task and vice versa | There will not be any fixed slots because of this resource utilization will be high where resource will be dynamically assigned. |
| Supports only map reduce | Map reduce and other application like Hadoop clusters will be supported which can run interactive querying and streaming data applications in parallel Map Reduce batch jobs as well streamlines. |
| Name Node acting as Single point of failure | This failure can be overcome by high availability |
|  |  |