1. **Explain the difference between FIFO and Capacity scheduler**

**FIFO Scheduler**

• The FIFO Scheduler places applications in a queue and runs them in the order of submission (first in, first out).

• Requests for the first application in the queue are allocated first; once its requests have been satisfied, the next application in the queue is served, and so on.

• The FIFO Scheduler has the merit of being simple to understand and not needing any configuration, but it’s not suitable for shared clusters.

• Large applications will use all the resources in a cluster, so each application has to wait its turn.

**Capacity Scheduler**

* The Capacity Scheduler is designed to allow sharing a large cluster while giving each organization a minimum capacity guarantee.
* The central idea is that the available resources in the Hadoop Map-Reduce cluster are partitioned among multiple organizations who collectively fund the cluster based on computing needs.
* There is an added benefit that an organization can access any excess capacity
* No waiting or Less Waiting compared to FIFO.
* On a shared cluster, it is better to use the Capacity Scheduler.
* Capacity scheduler needs some configuration
* Highly suitable for Shared cluster

**2.Explain the difference between FIFO and Fair scheduler**

**FIFO Scheduler**

• The FIFO Scheduler places applications in a queue and runs them in the order of submission (first in, first out).

• Requests for the first application in the queue are allocated first; once its requests have been satisfied, the next application in the queue is served, and so on.

• The FIFO Scheduler has the merit of being simple to understand and not needing any configuration, but it’s not suitable for shared clusters.

• Large applications will use all the resources in a cluster, so each application has to wait its turn.

**Fair Scheduler**

* With the Fair Scheduler, there is no need to reserve a set amount of capacity, since it will dynamically balance resources between all running jobs.
* Just after the first (large) job starts, it is the only job running, so it gets all the resources in the cluster.
* When the second (small) job starts, it is allocated half of the cluster resources, so that each job is using its fair share of resources.
* After the small job completes and no longer requires resources, the large job goes back to using the full cluster capacity again.
* The overall effect is both high cluster utilization and timely small job completion.

**3.Explain the difference between Capacity and Fair scheduler**

**Capacity Scheduler**

* The Capacity Scheduler is designed to allow sharing a large cluster while giving each organization a minimum capacity guarantee.
* The central idea is that the available resources in the Hadoop Map-Reduce cluster are partitioned among multiple organizations who collectively fund the cluster based on computing needs.
* There is an added benefit that an organization can access any excess capacity
* On a shared cluster, it is better to use the Capacity Scheduler.
* Capacity scheduler needs some configuration
* Highly suitable for Shared cluster

**Fair Scheduler**

* With the Fair Scheduler, there is no need to reserve a set amount of capacity, since it will dynamically balance resources between all running jobs.
* Just after the first (large) job starts, it is the only job running, so it gets all the resources in the cluster.
* When the second (small) job starts, it is allocated half of the cluster resources, so that each job is using its fair share of resources.
* After the small job completes and no longer requires resources, the large job goes back to using the full cluster capacity again.
* The overall effect is both high cluster utilization and timely small job completion.

**Q4. What are the limitations of hadoop 1.x and how they were overcome in hadoop 2.x**

**Hadoop 1.x has the following Limitations:**

1.It is only suitable for Batch Processing of Huge amount of Data, which is already in Hadoop System.

2.It is not suitable for Real-time Data Processing.

3.It is not suitable for Data Streaming.

4.It supports upto 4000 Nodes per Cluster.

5.It has a single component : JobTracker to perform many activities like Resource Management, Job Scheduling, Job Monitoring, Re-scheduling Jobs etc.

6.JobTracker is the single point of failure.

7.It does not support Multi-tenancy Support.

8.It supports only one Name Node and One Namespace per Cluster.

9.It does not support Horizontal Scalability.

10.It runs only Map/Reduce jobs.

11.It follows Slots concept in HDFS to allocate Resources (Memory, RAM, CPU). It has static Map and Reduce Slots. That means once it assigns resources to Map/Reduce jobs, it cannot re-use them even though some slots are idle.

**Hadoop 2.x solves Hadoop 1.x Limitations**

1.By decoupling MapReduce component responsibilities into different components.

2.By Introducing new YARN component for Resource management.

3.By decoupling component’s responsibilities, it supports multiple namespace, Multi-tenancy, Higher Availability and Higher Scalability.

4.It Supports Horizontal Scalability

5. It supports Multi-tenancy Support.

6.It Supports Multiple Namespaces