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

FIFO stands for first in first out.

In FIFO scheduling, Job Tracker pulls oldest job first from job queue. It doesn't consider about priority or size of the job. Hadoop’s built-in scheduler runs jobs in FIFO order.

Scheduler scans through jobs in order of priority when a task slot becomes free. Then it picks the map task in the job with data closest to the slave.

While

In Capacity scheduling, there are some queues. Each queue has its own assigned resources and uses FIFO strategy in itself.

In order to prevent some user take too much of resources in one queue, the scheduler can limit the resources for the jobs from each user.

When scheduling, all queues are monitored, if a queue does not use its allocated capacity, the spare capacity will be assigned to other queues.

Jobs with a higher priority can access to resources sooner than lower priority jobs.

We can configure the capacity scheduler within multiple Hadoop configuration files.

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

FIFO stands for first in first out.

In FIFO scheduling, Job Tracker pulls oldest job first from job queue. It doesn't consider about priority or size of the job. Hadoop’s built-in scheduler runs jobs in FIFO order.

Scheduler scans through jobs in order of priority when a task slot becomes free. Then it picks the map task in the job with data closest to the slave.

In contrast

Fair scheduler assigns resources to jobs such that all jobs get, on average, an equal share of resources over time.

If there is a single job running, the job uses the entire cluster. When other jobs are submitted, free task slots are assigned to the new jobs, so that each job gets roughly the same amount of CPU time.

It lets short jobs complete within a reasonable time while not starving long jobs. The scheduler actually organizes jobs by resource pool, and shares resources fairly between these pools.

By default, there is a separate pool for each user. There are the limits of concurrently running Map and Reduce tasks on “Task Tracker” of node. The fair scheduler can limit the number of concurrently running jobs from each user and from each pool.

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

In Capacity scheduling, there are some queues. Each queue has its own assigned resources and uses FIFO strategy in itself.

In order to prevent some user take too much of resources in one queue, the scheduler can limit the resources for the jobs from each user.

When scheduling, all queues are monitored, if a queue does not use its allocated capacity, the spare capacity will be assigned to other queues.

Jobs with a higher priority can access to resources sooner than lower priority jobs.

We can configure the capacity scheduler within multiple Hadoop configuration files.

Whereas

Fair scheduler assigns resources to jobs such that all jobs get, on average, an equal share of resources over time.

If there is a single job running, the job uses the entire cluster. When other jobs are submitted, free task slots are assigned to the new jobs, so that each job gets roughly the same amount of CPU time.

It lets short jobs complete within a reasonable time while not starving long jobs. The scheduler actually organizes jobs by resource pool, and shares resources fairly between these pools.

By default, there is a separate pool for each user. There are the limits of concurrently running Map and Reduce tasks on “Task Tracker” of node. The fair scheduler can limit the number of concurrently running jobs from each user and from each pool.

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

Hadoop 1.x has single NameNode which is single point of failure.  In case of Namenode failure, needs manual intervention to overcome.

Hadoop 2.x has feature to overcome SPOF with a standby Namenode and in case of Namenode failure, it is configured for automatic recovery.

The scaling capacity of Hadoop 1.x is increased in Hadoop 2.x upto 10000 nodes per cluster.

The MapReduce performs the resource management as well as processing in Hadoop 1.x.In Hadoop 2.x YARN is used for resource management and it has other processing models for processing.

A Namenode failure affects the stack in Hadoop 1.x.To solve this problem in Hadoop 2.x stack – Hive, Pig, HBase etc. are all equipped to handle Namenode failure.

Hadoop 2.x has support for Microsoft Windows which is not there in Hadoop 1.x.

In this way the limitations of Hadoop 1.x are overcome in Hadoop 2.x