Explain the difference between FIFO and Capacity scheduler?

* The difference between FIFO and capacity scheduler are listed below:

|  |  |
| --- | --- |
| FIFO SCHEDULER | CAPACITY SCHEDULER |
| ->FIFO Scheduler uses the first in first out principle irrespective of the size of the applications. | ->Capacity Scheduler runs the applications based on the capacity. |
| -> FIFO Scheduler uses FIFO so the small job is blocked until the large job completes. | -> Capacity Scheduler has a separate dedicated queue allows the small job to start as soon as it is submitted |
| -> FIFO Scheduler is not suitable for shared clusters. | -> Capacity Scheduler is suitable for shared clusters, that is it allows sharing of a Hadoop cluster along organizational lines. |
| -> FIFO Scheduler is very simple and easy to understand and does not need any configuration. | -> The Capacity Scheduler is designed to allow sharing a large cluster while giving each organization a minimum capacity guarantee. |
| ->Faster execution when compared to Capacity Scheduler | ->Here large job needs to wait so normally the execution time is slow when compared to FIFO scheduler. |
| ->It does not suit well for shared clusters because large job will take the resources and run whereas small job needs to wait. | ->The capacity scheduler is designed to work on shared clusters. |

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

Thus we are discussing below the differences between Hadoop 1.x and Hadoop 2.x

* The components of hadoop 1.x are HDFS and Map reduce whereas the components of hadoop 2.x are HDFS, MapReduce, YARN that is Hadoop 2.x introduces new YARN concept for Resource management.
* Hadoop 1.x has single namenode to manage the entire name space whereas hadoop 2.x has multiple namenode to manage name space.
* As Hadoop 1.x has only single node to manage entire name space it will result in High Single-Point-of-Failure (SPOF) which is not the case in Hadoop 2.x it even has automatic recovery for SPOF.
* Hadoop 1.x supports one and only one programming model: MapReduce. Hadoop 2.x supports multiple programming models with YARN Component like MapReduce, Iterative, Streaming, Graph, Spark, Storm etc.
* In Hadoop 1.x a namenode failure affects the stack, In Hadoop 2.x has the hadoop stack – Hive, Pig which are all equipped to handle namenode.
* Hadoop 1.x has lot of limitations in Scalability. Hadoop 2.x has overcome that limitation with new architecture.
* Hadoop 1.x supports maximum 4,000 nodes per cluster where as Hadoop 2.x supports more than 10,000 nodes per cluster.
* In Hadoop 1.x, a map tasks cannot run on reduce slot. So cluster utilization is low while in Hadoop 2.x, resources are dynamic and fine grained, this leads to better cluster utilization.
* Hadoop 1.x does not have multi-tenancy support whereas hadoop 2.x has multi-tenancy support.
* Hadoop 1.x HDFS uses fixed-size Slots mechanism for storage purpose whereas Hadoop 2.x uses variable-sized Containers.