[Configuring Fair Scheduler in Hadoop Cluster](https://amalgjose.wordpress.com/2015/07/24/configuring-fair-scheduler-in-hadoop-cluster/)

Hadoop comes with various scheduling algorithms such as FIFO, Capacity, Fair, DRF etc. Here I am briefly explaining about setting up fair scheduler in hadoop. This can be performed in any distribution of hadoop. By default hadoop comes with FIFO scheduler, some distribution comes with Capacity Scheduler as the default scheduler. In multiuser environments, a scheduler other than the default FIFO is definitely required. FIFO will not help us in multiuser environments because it makes us to wait in a single queue based on the order of job submission. Creating multiple job queues and assigning a portion of the cluster capacity and adding users to these queues will help us to manage and utilize the cluster resources properly.  
For setting up a fair scheduler manually, we have to make some changes in the resource manager node. One is a change in the yarn-site.xml and another is the addition of a new configuration file fair-scheduler.xml  
The configurations for a basic set up are given below.

**Step 1:**  
Specify the scheduler class in the yarn-site.xml. If this property exists, replace it with the below value else add this property to the yarn-site.xml

<property>

<name>yarn.resourcemanager.scheduler.class</name>

<value>org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.FairScheduler</value>

</property>

**Step 2:**  
Specify the Fair Scheduler allocation file. This property has to be set in yarn-site.xml. The value should be the absolute location of *fair-scheduler.xml* file. This file should be present locally.

<property>

<name>yarn.scheduler.fair.allocation.file</name>

<value>/etc/hadoop/conf/fair-scheduler.xml</value>

</property>

**Step 3:**  
Create the allocation configuration file  
A sample allocation file is given below. We can have advanced configurations in this allocation file. This is an allocation file with a basic set of configurations  
There are five types of elements which can be set up in an allocation file

*Queue element* :– Representing queues. It has the following properties:

* minResources — Setting the minimum resources of a queue
* maxResources — Setting the maximum resources of a queue
* maxRunningApps — Setting the maximum number of apps from a queue to run at once
* weight — Sharing the cluster non-proportional with other queues. Default to 1
* schedulingPolicy — Values are “fair”/”fifo”/”drf” or any class that extends
* org.apache.hadoop.yarn.server.resourcemanager.scheduler.fair.SchedulingPolicy
* aclSubmitApps — Listing the users who can submit apps to the queue. If specified, other users will not be able to submit apps to the queue.
* minSharePreemptionTimeout — Specifying the number of seconds the queue is under its minimum share before it tries to preempt containers to take resources from other queues.

*User*elements :– Representing user behaviors. It can contain a single properties to set maximum number apps for a particular user.

*userMaxAppsDefault*element :– Setting the default running app limit for users if the limit is not otherwise specified.

*fairSharePreemptionTimeout*element :– Setting the number of seconds a queue is under its fair share before it tries to preempt containers to take resources from other queues.

*defaultQueueSchedulingPolicy* element :– Specifying the default scheduling policy for queues; overriden by the schedulingPolicy element in each queue if specified.

 <?xml version="1.0"?>

<allocations>

<queue name="queueA">

<minResources>1000 mb, 1 vcores</minResources>

<maxResources>5000 mb, 1 vcores</maxResources>

<maxRunningApps>10</maxRunningApps>

<aclSubmitApps>hdfs,amal</aclSubmitApps>

<weight>2.0</weight>

<schedulingPolicy>fair</schedulingPolicy>

</queue>

<queue name="queueB">

<minResources>1000 mb, 1 vcores</minResources>

<maxResources>2500 mb, 1 vcores</maxResources>

<maxRunningApps>10</maxRunningApps>

<aclSubmitApps>hdfs,sahad,amal</aclSubmitApps>

<weight>1.0</weight>

<schedulingPolicy>fair</schedulingPolicy>

</queue>

<queue name="queueC">

<minResources>1000 mb, 1 vcores</minResources>

<maxResources>2500 mb, 1 vcores</maxResources>

<maxRunningApps>10</maxRunningApps>

<aclSubmitApps>hdfs,sree</aclSubmitApps>

<weight>1.0</weight>

<schedulingPolicy>fair</schedulingPolicy>

</queue>

<user name="amal">

<maxRunningApps>10</maxRunningApps>

</user>

<user name="hdfs">

<maxRunningApps>5</maxRunningApps>

</user>

<user name="sree">

<maxRunningApps>8</maxRunningApps>

</user>

<user name="sahad">

<maxRunningApps>2</maxRunningApps>

</user>

<userMaxAppsDefault>5</userMaxAppsDefault>

<fairSharePreemptionTimeout>30</fairSharePreemptionTimeout>

</allocations>

Here we created three queues queueA, queueB and queueC and mapped users to these queues. While submitting the job, the user should specify the queue name. Only the user who has access to the queue can submit jobs to a particular queue. This is defined in the acls. Another thing is scheduling rules. If we specify scheduling rules, the jobs from a particular user will be directed automatically to a particular queue based on the rule. I am not mentioning the scheduling rule part here.

After making these changes, restart the *resource manager.*

Now go to the resource manager web ui. In the left side of the UI, you can see a section named *Scheduler.*Click on that section, you will be able to see the newly created queues.

Now submit a job by specifying a queue name. You can use the option as below. The below option will submit the job to queueA. All the queues that we created are the sub-pools of root queue. Because of that, we have to specify queue name in the fomat ***parentQueue.subQueue***

-Dmapred.job.queue.name=root.queueA

Eg:*hadoop jar hadoop-examples.jar wordcount -Dmapred.job.queue.name=root.queueA  <input-location>  <output-location>*

If you are running a hive query, you can set these property in the below format. This property should be set at the top.

set mapred.job.queue.name=root.queueA