CapacityScheduler Guide

Table of contents

Copyright © 2008 The Apache Software Foundation. All rights reserved.

1. Purpose

This document describes the CapacityScheduler, a pluggable MapReduce scheduler for Hadoop which allows for multiple-tenants to securely share a large cluster such that their applications are allocated resources in a timely manner under constraints of allocated capacities.

2. Overview

The CapacityScheduler is designed to run Hadoop Map-Reduce as a shared, multi-tenant cluster in an operator-friendly manner while maximizing the throughput and the utilization of the cluster while running Map-Reduce applications.

Traditionally each organization has it own private set of compute resources that have sufficient capacity to meet the organization's SLA under peak or near peak conditions. This generally leads to poor average utilization and the overhead of managing multiple independent clusters, one per each organization. Sharing clusters between organizations is a cost-effective manner of running large Hadoop installations since this allows them to reap benefits of economies of scale without creating private clusters. However, organizations are concerned about sharing a cluster because they are worried about others using the resources that are critical for their SLAs.

The CapacityScheduler 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 being used by others. This provides elasticity for the organizations in a cost-effective manner.

Sharing clusters across organizations necessitates strong support for multi-tenancy since each organization must be guaranteed capacity and safe-guards to ensure the shared cluster is impervious to single rouge job or user. The CapacityScheduler provides a stringent set of limits to ensure that a single job or user or queue cannot consume dispropotionate amount of resources in the cluster. Also, the JobTracker of the cluster, in particular, is a precious resource and the CapacityScheduler provides limits on initialized/pending tasks and jobs from a single user and queue to ensure fairness and stability of the cluster.

The primary abstraction provided by the CapacityScheduler is the concept of *queues*. These queues are typically setup by administrators to reflect the economics of the shared cluster.

3. Features

Copyright © 2008 The Apache Software Foundation. All rights reserved.

The CapacityScheduler supports the following features:

- Capacity Guarantees - Support for multiple queues, where a job is submitted to a queue.Queues are allocated a fraction of the capacity of the grid in the sense that a certain capacity of resources will be at their disposal. All jobs submitted to a queue will have access to the capacity allocated to the queue. Adminstrators can configure soft limits and optional hard limits on the capacity allocated to each queue.
- Security - Each queue has strict ACLs which controls which users can submit jobs to individual queues. Also, there are safe-guards to ensure that users cannot view and/or modify jobs from other users if so desired. Also, per-queue and system administrator roles are supported.
- Elasticity - Free resources can be allocated to any queue beyond it's capacity. When there is demand for these resources from queues running below capacity at a future point in time, as tasks scheduled on these resources complete, they will be assigned to jobs on queues running below the capacity. This ensures that resources are available in a predictable and elastic manner to queues, thus preventing artifical silos of resources in the cluster which helps utilization.
- Multi-tenancy - Comprehensive set of limits are provided to prevent a single job, user and queue from monpolizing resources of the queue or the cluster as a whole to ensure that the system, particularly the JobTracker, isn't overwhelmed by too many tasks or jobs.
- Operability - The queue definitions and properties can be changed, at runtime, by administrators in a secure manner to minimize disruption to users. Also, a console is provided for users and administrators to view current allocation of resources to various queues in the system.
- Resource-based Scheduling - Support for resource-intensive jobs, wherein a job can optionally specify higher resource-requirements than the default, there-by accomodating applications with differing resource requirements. Currently, memory is the the resource requirement supported.
- Job Priorities - Queues optionally support job priorities (disabled by default). Within a queue, jobs with higher priority will have access to the queue's resources before jobs with lower priority. However, once a job is running, it will not be preempted for a higher priority job, *premption* is on the roadmap is currently not supported.

4. Installation

The CapacityScheduler is available as a JAR file in the Hadoop tarball under the *contrib/capacity-scheduler* directory. The name of the JAR file would be on the lines of hadoop-capacity-scheduler-*.jar.

You can also build the Scheduler from source by executing *ant package*, in which case it would be available under *build/contrib/capacity-scheduler*.

Copyright © 2008 The Apache Software Foundation. All rights reserved.

To run the CapacityScheduler in your Hadoop installation, you need to put it on the *CLASSPATH*. The easiest way is to copy the `hadoop-capacity-scheduler-*.jar` from to `HADOOP_HOME/lib`. Alternatively, you can modify *HADOOP_CLASSPATH* to include this jar, in `conf/hadoop-env.sh`.

5. Configuration

5.1. Using the CapacityScheduler

To make the Hadoop framework use the CapacityScheduler, set up the following property in the site configuration:

Property	Value
mapred.jobtracker.taskScheduler	org.apache.hadoop.mapred.CapacityTaskScheduler

5.2. Setting up queues

You can define multiple queues to which users can submit jobs with the CapacityScheduler. To define multiple queues, you should use the *mapred.queue.names* property in `conf/hadoop-site.xml`.

The CapacityScheduler can be configured with several properties for each queue that control the behavior of the Scheduler. This configuration is in the *conf/capacity-scheduler.xml*.

You can also configure ACLs for controlling which users or groups have access to the queues in `conf/mapred-queue-acls.xml`.

For more details, refer to Cluster Setup documentation.

5.3. Queue properties

5.3.1. Resource allocation

The properties defined for resource allocations to queues and their descriptions are listed in below:

Name	Description
mapred.capacity-scheduler.queue.<queue-name>	Percentage of the number of slots in the cluster that are made to be available for jobs in this queue. The sum of capacities for all queues should be less than or equal 100.

Copyright © 2008 The Apache Software Foundation. All rights reserved.

mapred.capacity-scheduler.queue.<queue-name>	maximum-capacity defines a limit beyond which a queue cannot use the capacity of the cluster.This provides a means to limit how much excess capacity a queue can use. By default, there is no limit. The maximum-capacity of a queue can only be greater than or equal to its minimum capacity. Default value of -1 implies a queue can use complete capacity of the cluster. This property could be to curtail certain jobs which are long running in nature from occupying more than a certain percentage of the cluster, which in the absence of pre-emption, could lead to capacity guarantees of other queues being affected. One important thing to note is that maximum-capacity is a percentage , so based on the cluster's capacity it would change. So if large no of nodes or racks get added to the cluster , maximum Capacity in absolute terms would increase accordingly.
mapred.capacity-scheduler.queue.<queue-name>	Each queue enforces a limit on the percentage of resources allocated to a user at any given time, if there is competition for them. This user limit can vary between a minimum and maximum value. The former depends on the number of users who have submitted jobs, and the latter is set to this property value. For example, suppose the value of this property is 25. If two users have submitted jobs to a queue, no single user can use more than 50% of the queue resources. If a third user submits a job, no single user can use more than 33% of the queue resources. With 4 or more users, no user can use more than 25% of the queue's resources. A value of 100 implies no user limits are imposed.
mapred.capacity-scheduler.queue.<queue-name>	The multiple of the queue capacity which can be configured to allow a single user to acquire more slots. By default this is set to 1 which ensure that a single user can never take more than the queue's configured capacity irrespective of how idle th cluster is.
mapred.capacity-scheduler.queue.<queue-name>	If true, priorities of jobs will be taken into account in scheduling decisions.

Copyright © 2008 The Apache Software Foundation. All rights reserved.

5.3.2. Job initialization

Capacity scheduler lazily initializes the jobs before they are scheduled, for reducing the memory footprint on jobtracker. Following are the parameters, by which you can control the initialization of jobs per-queue.

Name	Description
mapred.capacity-scheduler.maximum-system-jobs	Maximum number of jobs in the system which can be initialized, concurrently, by the CapacityScheduler. Individual queue limits on initialized jobs are directly proportional to their queue capacities.
mapred.capacity-scheduler.queue.<queue-name>	The maximum number of tasks, across all jobs in the queue, which can be initialized concurrently. Once the queue's jobs exceed this limit they will be queued on disk.
mapred.capacity-scheduler.queue.<queue-name>	The maximum number of tasks per-user, across all the of the user's jobs in the queue, which can be initialized concurrently. Once the user's jobs exceed this limit they will be queued on disk.
mapred.capacity-scheduler.queue.<queue-name>	The multipe of (maximum-system-jobs * queue-capacity) used to determine the number of jobs which are accepted by the scheduler. The default value is 10. If number of jobs submitted to the queue exceeds this limit, job submission are rejected.

5.4. Resource based scheduling

The CapacityScheduler supports scheduling of tasks on a `TaskTracker`(TT) based on a job's memory requirements in terms of RAM and Virtual Memory (VMEM) on the TT node. A TT is conceptually composed of a fixed number of map and reduce slots with fixed slot size across the cluster. A job can ask for one or more slots for each of its component map and/or reduce slots. If a task consumes more memory than configured the TT forcibly kills the task.

Currently the memory based scheduling is only supported in Linux platform.

Additional scheduler-based config parameters are as follows:

Name	Description
mapred.cluster.map.memory.mb	The size, in terms of virtual memory, of a single

Copyright © 2008 The Apache Software Foundation. All rights reserved.

	map slot in the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single map task via `mapred.job.map.memory.mb`, upto the limit specified by `mapred.cluster.max.map.memory.mb`, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.cluster.reduce.memory.mb	The size, in terms of virtual memory, of a single reduce slot in the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single reduce task via `mapred.job.reduce.memory.mb`, upto the limit specified by `mapred.cluster.max.reduce.memory.mb`, if the scheduler supports the feature.The value of -1 indicates that this feature is turned off.
mapred.cluster.max.map.memory.mb	The maximum size, in terms of virtual memory, of a single map task launched by the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single map task via `mapred.job.map.memory.mb`, upto the limit specified by `mapred.cluster.max.map.memory.mb`, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.cluster.max.reduce.memory.mb	The maximum size, in terms of virtual memory, of a single reduce task launched by the Map-Reduce framework, used by the scheduler. A job can ask for multiple slots for a single reduce task via `mapred.job.reduce.memory.mb`, upto the limit specified by `mapred.cluster.max.reduce.memory.mb`, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off.
mapred.job.map.memory.mb	The size, in terms of virtual memory, of a single map task for the job. A job can ask for multiple slots for a single map task, rounded up to the next multiple of `mapred.cluster.map.memory.mb` and upto the limit specified by `mapred.cluster.max.map.memory.mb`, if the scheduler supports the feature. The value of

Copyright © 2008 The Apache Software Foundation. All rights reserved.

	-1 indicates that this feature is turned off iff `mapred.cluster.map.memory.mb` is also turned off (-1).
mapred.job.reduce.memory.mb	The size, in terms of virtual memory, of a single reduce task for the job. A job can ask for multiple slots for a single reduce task, rounded up to the next multiple of `mapred.cluster.reduce.memory.mb` and upto the limit specified by `mapred.cluster.max.reduce.memory.mb`, if the scheduler supports the feature. The value of -1 indicates that this feature is turned off iff `mapred.cluster.reduce.memory.mb` is also turned off (-1).

5.5. Reviewing the configuration of the CapacityScheduler

Once the installation and configuration is completed, you can review it after starting the MapReduce cluster from the admin UI.

- Start the MapReduce cluster as usual.
- Open the JobTracker web UI.
- The queues you have configured should be listed under the *Scheduling Information* section of the page.
- The properties for the queues should be visible in the *Scheduling Information* column against each queue.
- The /scheduler web-page should show the resource usages of individual queues.

6. Example

Here is a practical example for using CapacityScheduler:

```
<?xml version="1.0"?>
<configuration>
<!-- system limit, across all queues -->
<property>
  <name>mapred.capacity-scheduler.maximum-system-jobs</name>
  <value>3000</value>
<description>Maximum number of jobs in the system which can be
initialized,
concurrently, by the CapacityScheduler.
</description>
</property>
<!-- queue: queueA -->
<property>
```

Copyright © 2008 The Apache Software Foundation. All rights reserved.

```
  <name>mapred.capacity-scheduler.queue.queueA.capacity</name>
  <value>8</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.supports-priority</name>
  <value>false</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.user-limit-factor</name>
  <value>10</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.maximum-initialized-active-tasks</name>
  <value>200000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueA.init-accept-jobs-factor</name>
  <value>100</value>
</property>
<!-- queue: queueB -->
<property>
  <name>mapred.capacity-scheduler.queue.queueB.capacity</name>
  <value>2</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.supports-priority</name>
  <value>false</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.user-limit-factor</name>
  <value>1</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.maximum-initialized-active-tasks</name>
  <value>200000</value>
```

Copyright © 2008 The Apache Software Foundation. All rights reserved.

```
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueB.init-accept-jobs-factor</name>
  <value>10</value>
</property>
<!-- queue: queueC -->
<property>
  <name>mapred.capacity-scheduler.queue.queueC.capacity</name>
  <value>30</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.supports-priority</name>
  <value>false</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.user-limit-factor</name>
  <value>1</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.maximum-initialized-active-tasks</name>
  <value>200000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueC.init-accept-jobs-factor</name>
  <value>10</value>
</property>
<!-- queue: queueD -->
<property>
  <name>mapred.capacity-scheduler.queue.queueD.capacity</name>
  <value>1</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueD.supports-priority</name>
  <value>false</value>
</property>
```

Copyright © 2008 The Apache Software Foundation. All rights reserved.

```
<property>
  <name>mapred.capacity-scheduler.queue.queueD.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueD.user-limit-factor</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueD.maximum-initialized-active-tasks</name>
  <value>200000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueD.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueD.init-accept-jobs-factor</name>
  <value>10</value>
</property>
<!-- queue: queueE -->
<property>
  <name>mapred.capacity-scheduler.queue.queueE.capacity</name>
  <value>31</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.supports-priority</name>
  <value>false</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.user-limit-factor</name>
  <value>1</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.maximum-initialized-active-tasks</name>
  <value>200000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueE.init-accept-jobs-factor</name>
```

Copyright © 2008 The Apache Software Foundation. All rights reserved.

```
  <value>10</value>
</property>
<!-- queue: queueF -->
<property>
  <name>mapred.capacity-scheduler.queue.queueF.capacity</name>
  <value>28</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.supports-priority</name>
  <value>false</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.minimum-user-limit-percent</name>
  <value>20</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.user-limit-factor</name>
  <value>1</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.maximum-initialized-active-tasks</name>
  <value>200000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.maximum-initialized-active-tasks-per-us
  <value>100000</value>
</property>
<property>
  <name>mapred.capacity-scheduler.queue.queueF.init-accept-jobs-factor</name>
  <value>10</value>
</property>
</configuration>
```

Copyright © 2008 The Apache Software Foundation. All rights reserved.