Title: [YARN-1963] Support priorities across applications within the same queue
Authors: Sunil Govind, Vinod Kumar Vavilapalli, Rohith Sharma K S with inputs from Wangda Tan, Jian He, Jason Lowe, Eric Payne
Last modified: July 24 2015

1.1 Preamble

Many a time, it is needed to execute some YARN applications at higher priority, regardless of other applications that are already running in a YARN cluster. The existing mechanism for enabling such a use-case in YARN's CapacityScheduler is by creating multiple queues and making users submit applications to higher priority and lower priority queues separately, each potentially setup with appropriate capacities. It is desirable to enable users of YARN clusters to define a priority for an application at its submission time. YARN-1963 is focused on handling such application-priorities as described in this document.

1.2 Problem statement

Using existing features in YARN's Capacity Scheduler, to support different priorities among a queue's application, one can do the following:

a. Configure multiple queues and define resources to each queue based on the priority.
b. User submits applications to different queues based on the priority define for that queue.
c. Access control permission can be set to each queue based on demand.

This is an inflexible way to enable priorities for users and can be made easier by letting priorities be specified for Application at submission time itself. Different priority applications can run in same queue and can use the resources of that queue based on the application-priority.

1.3 Requirements

- Users should be able to set priorities for each of their applications. These priorities should be respected in every leaf-queue.
- Other scheduling constraints within a leaf-queue like user-limits, submission-time (FIFO behavior) should meaningfully interact with priorities
  - A high priority application from a user who already hit-limits should continue to wait in the queue.
  - Applications with same priority should be ordered by their submission-times
- ACLs on priorities – Access Control Lists should exist on priorities to avoid users from getting incentivized to submit apps all with the highest priority.

1.4 Proposal

YARN-1963 defines priority for an application and makes the application run in the cluster as per its assigned priority.

Solution for this proposal has mainly three parts,

- **Setting the priority to an application:** Application-priority can be set via the submission API.
- **Scheduling applications based on Priority**: Based on the application-priority, a higher priority application's requests will be scheduled first from Resource Manager. Once all these requests of higher priority applications are served, then lower priority application requests will get served from Resource Manager.
- **Access Control Lists**: User must be granted permission to run a specific priority application. This can be taken as access control configuration for each priority level.

1.5 Details

1.5.1 How can End User submit an application by specifying priority?

Priority can be set via below methods:

- **With API**: Below API can be used to set the Application priority.

  application.setApplicationPriority(Priority priority)

- **With command line parameter:** While submitting an application based on each application configuraton one can set the priority of the application. **For eg:** MapReduce can use mapreduce.job.priority.

1.5.2 How Admin can configure priority?

1. Configure application priority for the cluster:

ApplicationPriority is a continuum with their domain space as a set of integers. Administrators can define a maximum priority is cluster level. Hence a maximum cap is defined in cluster level for each application. Admins can also define a default priority per queue level.

yarn.cluster.max-application-priority = 10

End user can use this a maximum priority cap to set the corresponding priority for the application.

2. Configure default priority per queue level:

Admin can define a default priority and can be configured per queue level. This will help for those applications which are submitted to the queue without specifying any priority. All such applications can run into this category of priority label.

yarn.scheduler.root.<queue_name>.default-application-priority=3

Note: Queue Level priority configurations can be configured separately for Capacity Scheduler and Fair Scheduler for now.

1.5.3 APIs or CLI support needed for Application Priority

1. Priority in Queue and Application under different criteria:

Admins and End users will require few sets of API's (also REST support) for priority.

- API to get/set default priority in a given queue.
- API to get/set the running priority of an application.
- API to get the max priority in the cluster.

CLI should support above mentioned query result and it can be made as a yarn application/cluster command

2. Change Priority of a submitted application at runtime

As per few use cases, it will be very convenient for a user or admin to change the priority of an application during runtime.

- A User, who submitted an application with a given priority, should be able to change priority during runtime with the help of an API.
- Admin also will have privilege to change the priority of an application during time with help of API as well as CLI (admin command)

Note:

- The User or Admin, who is changing the priority of an application, must have proper ACL privileges for the successful change of priority.
- Once priority is changed for a running application, all new pending requests from RM will be granted with updated priority level.

This part will be taken up in separate JIRA and will be tracked as soon as the initial framework is ready.

1.5.4 Scheduler side support for Application-priority:

Scheduler changes for Application priority have to be done separately for Fair and Capacity schedulers. A brief design approach will be as follows.

In each queue, there can be many applications submitted with different priorities. In such cases, Scheduler will try allocating resources to an application that has higher priority than others and then for next lower priority application and then so on.

- By changing the application-comparator and Ordering Policy in Capacity and Fair Scheduler, always a higher priority application can be fetched and then used for later resource allocation. If there are multiple applications at same Application-priority level, then comparison will be based on submission-timestamp as happening normally now.
- During APP_ATTEMPT_ADDED event handling in both Capacity and Fair Scheduler, the application-priority can be fetched from Application Submission context. Application-priority can then set to the scheduler app level and can be used later.

Note: Interaction with scheduling parameters

- User-limit: When priority comes into picture for each application, user limit

check will be having a preference. If user-limit/am resource limit percentage is adhered, then only priority level comparison will done.

- User-limit-factor: Same explanation applicable for user-limit-factor also.

1.5.5 Access Control Lists for Application-priority:

User-level access permission to use different application-priorities can be configured and controlled. This is to avoid situations where all users try running at max-cluster-priority applications in the cluster and thus degrading the value of priorities.

- Access Control Lists can be set per priority within each queue.
- Separate ACL configurations are required for Fair and Capacity schedulers.

1. How ACL's can be configured per queue level?

Below is an example configuration that can be added in capacity scheduler configuration file for each Queue level.

yarn.scheduler.capacity.root.<queue_name>.<priority>.acl=user1,user2

Under each configuration, a set of users and groups can be configured and only those users/groups should be able to run that specified priority applications.

Note: Similar configuration can be kept under fair scheduler configuration file to define ACLs per queue.

2. Behavior together with queue acls:

If Queue level ACLs are configured, then this priority-based acl configuration should be a subset of that configuration – a user should have both queue-acl and priority-acl to be able to submit applications at that priority in that queue.

Note: Under both scenarios mentioned above, such application which are not able to meet the ACL criteria can be marked as Rejected. There is also an alternative to fall back to default priority label of queue and continue submitting application. This can be confirmed.

1.5.6 View Application-priority from RM Web UI:

Priority can be displayed in RM Web UI. This will help in getting the details of the submitted application-priority.

Few Points of Note:

1. **MAPREDUCE-314 Starvation problem**: Priority inversion problem can be solved by using below way.
   a. Applications Master's head room will be modified to reflect the priority change. Based on this input from Resource manager, respective application masters can act on the change.

2. **Preemption:** Without preemption, if a lower priority application is using full Queue's capacity, then a higher priority application submitted in the same Queue has to wait. This

wait period will depend on how fast the lower priority application will be completed. During this time Capacity Scheduler will ensure that resource allocation will not be happening for lower priority application further.

Priority based preemption policy can help in this regard, a sub JIRA will be opened to track this.