YARN-7117 : Capacity Scheduler: Support Auto Creation of Leaf Queues during queue mapping

Suma Shivaprasad / Wangda Tan
with input from
Sunil Govind, Vinod Vavilapalli, Clay Baenziger

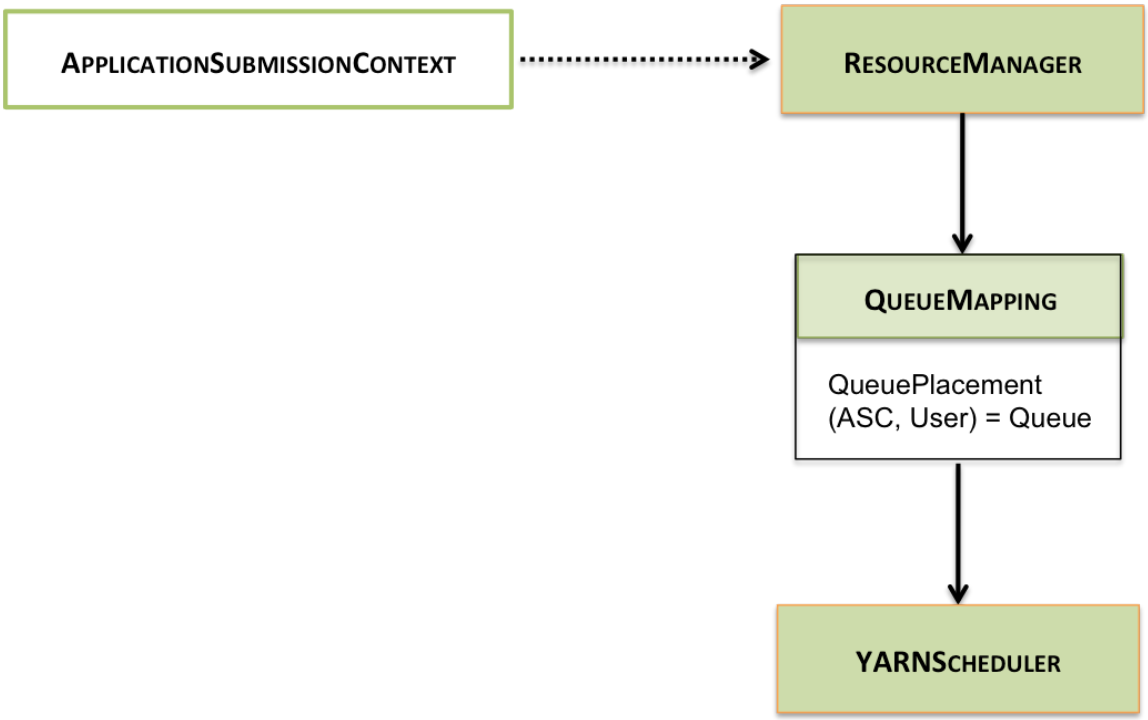
Problem statement

Capacity scheduler supports a default queue-mapping policy based on user/group - YARN-3635 For more details, refer to Apache documentation. Administrators can define a queue mapping policy to map applications submitted by users automatically to queues. YARN-6689 allows for configuration of other complex queue mapping policies. The most common use case of queue mapping is to map a queue explicitly for some user or group. However this requires additional

steps by admin to update capacity-scheduler.xml, add new queues, adjust capacities and refresh queues. The intent of this jira is to support auto creation of leaf queues when a new user/group submits a request to a parent queue which has been configured to allow auto-creation. Leaf Queues which have been manually created by admins are addressed as “pre-configured” queues in the rest of this document.

Workflow

When an application is submitted to RM, the *ApplicationSubmissionContext* and User information is used by the configured *QueueMapping* policy evaluator to place applications on to leaf queues. The workflow is depicted below



Configs

Queue Mapping

This is an example of existing queue mapping config:

```
<property>
<name>yarn.scheduler.capacity.queue-mappings</name>
<value>u:user1:queue1,g:group1:queue2,u:%user:%user</value>
</property>
```

This config maps user1 to queue1, users from group1 to queue2 and all other users to queue name same as their user name. More details of queue mapping config please refer to Apache documentation.

However this config assume queues will be created prior to launch of the job. For the use case where queues will be created automatically when applications are submitted, we need to specify the parent queue name so scheduler knows where to create the leaf queue. The proposal is to add parent queue name as part of the queue mapping policy config:

Queue mapping config includes parent queue location:

```
<property>
<name>yarn.scheduler.capacity.queue-mappings</name>
<value>u:user1:queue1(parent-queue=marketing),g:group1:queue2,u:%user:%user(parent=engineering),u:user2:%primary_group(parent-queue=finance)</value>
</property>
```

In the above examples

- `u:user1:queue1(parent-queue=marketing)`

  `queue1` is a leaf queue which will be auto created if it does not exist already when an application is submitted by user1 to parent-queue `'marketing'`.

- `u:%user:%user(parent-queue=engineering)`

  Enables auto creation of leaf queues for all users under `'engineering'` parent queue

- `u:user2:%primary_group (parent-queue=true, parent-queue=finance)`

  Enables auto-creation for applications submitted by 'user2' and the queue will be created with the user's primary group name

Queue Mappings could be updated to remove the option to auto-create for a specified user->queue/group->queue mapping. This will take effect for future auto-creations and the current auto created queues will be intact.

ParentQueue Configs

Parent queues are configured to allow auto creation of sub queues. By default, auto creation is disabled.

```
prefix.<parent-queue-path>.enable-auto-created-child-queues [default : false]
```

The following properties will be used by parent queue to configure an auto-created leaf queue.

```
<parent-queue>.leaf-queue-template.capacity

(Other existing leaf queue configs like):
    - maximum-capacity
    - user-limit
    - maximum-applications, etc.
```

LeafQueue Configs

Manual modification of auto created leaf queue configs will not be supported for now.

Implementation details

Capacity management of auto-created leaf queues

Minimum resource guaranteed could be specified for queues at the parent queue which are automatically created.

```
prefix.<parent-queue-path>.leaf-queue-template.capacity
```

After YARN-5881, absolute resources can be specified for auto created queues.
Since queues are auto-created, a very likely scenario is that queues are created but not actively used. To resolve this issue, CS could actively maintain guaranteed capacities of auto-created leaf queues.

Scheduler will maintain a list of **#active-leaf-queues** under ParentQueue. An active-leaf-queue is a leaf queue that has at least one application not in final state.

Parent queue's guaranteed resource will be checked and enforced when state of leaf queue is changed to active. YARN scheduler will enforce:

```
Σ(leafQueue.guaranteed) (leafQueue ∈ {active-leaf-queues}) <= parent.guaranteed).
```

Note that the checks that all leaf-queue capacities under parent-queue add upto 100% needs to be relaxed for parent-queues that allow auto-creation since initially there might be only a few auto-created queues which may not totally use up 100% of parent-queue's capacity.

If we don't have guaranteed room in the parent queue, queues with 0 capacity (best effort queue) will be created. Applications running in these best effort queues could be starving if no capacity is available

Deactivation of auto-created queues

A configurable policy can decide when to reset queue's capacity (quota) assignments. A default behaviour could be that the queue's guaranteed capacity is set to 0 when there is no pending/active app. An admin can disable this and always choose to keep the leaf queues active under a parent-queue.

Example of capacity management of auto-created leaf queues

To illustrate capacity management with an example,

parent.guaranteed-capacity = 15G, guaranteed-for-children=5G.

At T_0 no sub queue.
At T_1 applications submitted to q1/q2/q3, so scheduler creates q1 (5G), q2 (5G), q3 (5G).
At T_2, application submitted to q4, so scheduler creates q4 which is inactive (0G).
At T_3, application submitted to q5, so scheduler creates q5 which is inactive (0G).
At T_4, application finishes in q2, and at T_5 (T_5 - T_4 > configured-threshold), scheduler set q2.guaranteed to 0G, and q4's is now active with q4.guaranteed set to 5G.
At T_6, application finishes in q3, and at T_7 (T_7 - T_6 > configured-threshold), scheduler move q3's guaranteed capacity to q5

Alternative approaches

Another option which was suggested by Jason Lowe and considered to manage capacites across the auto created leaf queues are that these queues are created with no guarantees (capacity = 0%, max capacity = 100%) . This essentially means that there is no minimum guaranteed capacity but best-effort capacity allocated to these queues when scheduled. However there are no SLA

guarantees for the auto created leaf queues and applications which have SLA requirements cannot be submitted on these queues. Also, max-am-resource-percent will be 0 for these queues which means that these queues can allow only 1 max active application which may not be acceptable.

Queue Deletion/Cleanup

Automated Queue cleanup/deletion can be turned on based on a configuration. By default deletion of automatically created queues will be disabled

Admins will be provided an option to configure a simple ExpiryPolicy for cleaning up inactive queues - Eg: If a queue is inactive for 24h, scheduler may do GC for such queues.

Access control

ACLs of auto-created queues will inherit ACLs of parent queue

It's also important to do customizations for auto created queues, for example: queue for different users will be created on-demand under root.users queue, for example: root.users.user1, root.users.user2. It's better that we can restrict only owner of the queue can have access to the queue, we don't allow user1 submit jobs to root.users.user2 and vice-versa

This can be controlled by auto-created queue ACL policies and context from queue mapping policy can be used to be able to set up queue ACLs.

Relation to other scheduler features

Allocation / Preemption

Capacity Scheduler will treat automatically created queues no different from other leaf queues, which means scheduler will use current behaviour to do preemption / user-limit, etc. for auto-created queue.

Ordering policy within Auto created leaf queues

CS currently supports the following intra leaf queue ordering policies - FIFO/Fair which applies within a leaf queue for pending application submissions. This will suffice for auto created leaf queues as well.

Queue Priority

YARN-5864 added the notion of priorities across queues. Since we do not allow updation of priority on an auto-created leaf queue, all auto-created leaf queues will have the same priority under a parent queue. The behaviour for auto created queues will be the same as existing logic for handling queue priorities for pre-configured leaf queues. Application priority will continue to use existing logic within an active auto created leaf queues

Open items

1. Parent queue's leaf queue template configuration can be updated while there are active running leaf queues. This can take effect only after the current active applications in these queues are in final state ?
2. User might not have appropriate ACLs to submit to a queue. In this case, we need to fail application submission when auto creation of leaf queue fails?
3. Should we should the dynamic queues on UI same as pre configured queues?