Queue Deletion in Fair Scheduler

Yufei Gu, 3/19/2018

Problem

YARN Fair Scheduler (FS) never cleans up queues even if they are deleted in the allocation file, or were dynamically created and are never going to be used again. Queues always remain in memory which leads to two following issues.

- Steady fairshares aren't calculated correctly due to remaining queues, which isn't a big issue since steady fairshare is not involved in real scheduling decisions, but confuses users.
- WebUI shows deleted queues, which confuses users (YARN-4022).

In FS, queues can be created in two ways: statically via the configuration and dynamically via placement rules. For static queues the FS periodically (every 10s by default) reloads the allocation file (fair-scheduler.xml) and creates any new queues from the file which don't exist in memory. Queues are dynamically created according to certain queue placement rules, e.g. a common rule is to create a dynamic queue if user submits an application to a queue that doesn't exist.

The goal

We want to support proper queue deletion without restarting Resource Manager.

- Static queues without any entries that are removed from fair-scheduler.xml should be deleted from memory.
- Dynamic queues without any entries should be deleted.
- RM Web UI should only show the queues defined in the scheduler at that point in time.

Existing functionality for queue deletion

Generally, in existing code, a queue is never deleted in FS. However, there are two tiny cases that a queue can be removed.

- To remove incompatible queues while reloading the allocation file. By default a queue is a leaf when it does not have a sub queue defined. In the configuration a queue can be labeled as a parent queue using the "parent" tag. This causes a queue without sub queues in the allocation file to be processed as a parent queue at runtime. A "parent" tagged leaf queue usually has sub queues created dynamically. A leaf queue with "parent" tag and a leaf queue without the tag aren't compatible with each other.
- To cleanup expired queues in reservation system. Reservation system isn't widely used in FS.

Non-Goals

It isn't necessary to add a CLI command or REST API to delete a queue. An administrator deletes a queue by updating the allocation file.

Application recovery that refers to non-existing or deleted queues.

Approach

Queue deletion can be triggered each time FS reloads the allocation file. The major logic will be in function `QueueManager#updateAllocationConfiguration()`.

FS doesn't reload the allocation file if there is no change. No static queue will be deleted in that case. However, FS should still delete zero-application dynamic queues. To support removing empty dynamic queues without loading the configuration each time we add a queue flag to distinguish a dynamic queue and a static queue in memory. The boolean flag, with the proposed name of *isDynamic*, must be set for both parent and leaf queues. Queues defined in the configuration and the *root* queue will have a value of *FALSE* all other queues will have a value of *TRUE*. In case of queue transition between static and dynamic, we need to make sure that the correct flag is set. The following are the rules.

1. A empty dynamic queue becomes a static queue. We delete the dynamic queue, and create a new static queue.
2. A non-empty dynamic queue becomes a static queue. We changing its *isDynamic* flag from TRUE to FALSE.
3. A empty static queue becomes a dynamic queue, which is deleted in the allocation file. We just delete it.
4. A non-empty static queue becomes a dynamic queue, which is deleted in allocation file. We change its *isDynamic* flag from FALSE to TRUE.

A dynamic queue may only be deleted if there is no application in the queue itself and all its sub queues. For a leaf queue this means that the counts for both runnable and non-runnable applications must be zero. A parent queue can only be deleted if there are no sub queues below it left and the *isDynamic* flag is set to *TRUE*.

Queue deletion should start with the removal of leaf queues and recurse up towards the root queue.

Application Recovery

An application should always be restored from state store. It won't affect queue deletion since YARN-7139 is in and YARN-7968 will be in. YARN-7968 will handle the case that an application

is restored to a queue not existing anymore. The solution is to create a special dynamic queue only for recovery if the queue associated with the application has been deleted.