

YARN-2113 Intra-Queue Preemption : Behavior

[Overview](#)

[Design Approach/Assumptions](#)

[Assumptions of preemption order](#)

[Impacts of these cases with example](#)

Overview

Currently YARN supports priority based and user-limit based intra queue preemption. These two work in sync. This document is trying to capture exact expectation and design approach for each case.

Design Approach/Assumptions

USERLIMIT_FIRST (Default is USERLIMIT_FIRST)

Case Statement: app1 can preempt a container from app2 under below criteria

Criteria 1: Application order with Priority + FIFO from low to high

app2 (pri=0) → app3 (pri=1) → app1 (pri=2)

1. *app1.priority > app2.priority && app1.used < app1.user-limit && app1.user == app2.user*
 - a. No preemption is allowed in this case.
2. *app1.priority > app2.priority && app1.used < app1.user-limit && app1.user != app2.user*
 - a. Ensure that app1.used <= app1.user-limit at any time
 - b. Ensure that app2.used > app2.user-limit. This means that preemption should not bring app2.used below app2.user-limit.
3. *app1.priority = app2.priority && app1.used < app1.user-limit && app1.user != app2.user*
 - a. Ensure that app2.used > app2.user-limit
 - b. Ensure that app1.used should not become more than app1.user-limit at any time

Now these 2 cases will be referenced in below cases as well with same indexes (for same case, behavior will be updated based on respective type)

PRIORITY_FIRST

1. *app1.priority > app2.priority && app1.used < app1.user-limit && app1.user == app2.user*
 - a. App1 can preempt as far as app1.user is below its ideal_allocation [1]

2. *app1.priority > app2.priority && app1.used < app1.user-limit && app1.user != app2.user*
 - a. Same as above (TODO: Yahoo need to confirm this one).
3. *app1.priority = app2.priority && app1.used < app1.user-limit && app1.user != app2.user*
 - a. Same as conditions of *USERLIMIT_FIRST* #3.

Assumptions of preemption order

1. Preemption module always iterate from lower to higher priority apps (if there is no priority, then based on FIFO order of submission). This is in line with scheduler.
2. To preempt, containers are selected from youngest to oldest by adhering to the fact that apps are also selected based on preemption order mentioned in Assumption 1.

Impacts of these cases with example

- A. With Case 1, it is possible that priority based preemption may not kick in when we configure *USERLIMIT_FIRST*

Scenario:

- user1 starts app1 at priority 0 and consumes the whole queue.
- user1 starts app2 at priority 1
- MULP is 100 hence it will be Queue' capacity. (given 1 user)

To honor "This means that preemption should not bring app2.used below app2.user-limit.", we can't preempt any resources here.