Job and Host Assignment

Since there are many variables that an affect when and whether a particular job is dispatched to a worker, it might help if some of the decisions involved in this process were described in a bit of detail.

Selection Events

There are 3 different times that host selection occurs

1. When a job slot becomes available on a worker
2. When a job is submitted, retried, unblocked, or shoved (Supervisor is explicitly instructed to re-evaluate the job)
3. When a global resource becomes available

Job Slot Becomes Available

Selection criteria:

- Pending state: jobs with pending instances are selected
- Job requirements: these pending instances are filtered by requirements; can the Worker meet them?
- Job reservations: can the Worker honor the reservations for the resources specified in the reservations?

Sorting is done at the same time as filtering ("selection criteria") is done -- i.e., in the same loop, when a job passes the "selection" criteria, it's inserted into the appropriate position in the sorted list. The sorting is based on the algorithm selected with the supervisor_queue_algorithm parameter

The default sorting criteria are:

- Job cluster: jobs whose cluster matches the worker's are moved to the top of the list
- Job priority: priority is used as the tie-breaker when cluster is the same
- Job ID: since the jobID is based on submission time, the tie-breaker when cluster and priority are the same is essentially "first come, first served".

Job State Change

When a job is submitted, retried, unblocked, or shoved, the job's cluster and priority are compared to all running job instances to see if there are any running jobs that this new job can preempt. Some jobs have a flag set that indicates that they can never be preempted, these are filtered out of the list.

If there are preemptable jobs, the workers that these jobs are running on are checked to see if they can satisfy the job's requirements and reservations. If the requirements and reservations can be met by a particular Worker, then the job instance that is running on that Worker is marked for preemption, and depending on the supervisor's preemption policy (passive or aggressive), the job instance is pre-empted as follows:

- passive preemption: when the job instance finishes the frame it's currently working on
- aggressive preemption: the job instance is killed immediately, and both the instance and the frame get put back into the queue in a pending state

There are a couple of major things that happen before preemption:

1. The list of hosts are filtered down, just like the list of jobs are filtered above, using criteria such as the job's requirements and the queuing algorithm's host-job pair match/reject routines.
2. Then the Supervisor tries to find open/idle slots for the job in consideration, from the filtered down list of hosts, and dispatches instances if slots are found.
3. If there are still pending instances remaining for the job, preemption occurs.

Available Global Resource

When a global resource becomes available, an instance from a job pending for that specific resource must be considered for dispatch. In a nutshell, here’s the sequence of events:

1. Get a list of all “ready" jobs (“pending”, “running” with pending instances, or active jobs with the “expand” flag set)
2. Filter down the list to only jobs that have reserved the specific global resource
3. Loop through these jobs to find a suitable job to start
4. Start the job