Scheduling Policy for Institution "A"

This institution operates a large compute cluster supporting many different types of users, and thus has complex scheduling needs. This document outlines the requirements for scheduling jobs to be run on this pool.

They run one pool with many submit points, the number of execute cores is the sum of the desired slots per job type in the table below. The various execute machines are not identical, but for scheduling reasons, today they are considered identical, with the unit of scheduling being the core. In the future, they would like to schedule multi core jobs, again with the unit of scheduling being the core. They would also like to be able to schedule jobs with varying amounts of requested memory, and bias the accounting accordingly, but aren't sure yet how they would like this to work.

The overall goal is to have some notion of fairness of job startup waiting time between the various types of jobs. To implement this, given enough demand from each of these types of jobs, they would like to see the following number of cores running:

Type	# cores	Preemption (in hours)
Large Group	8500 (hard limit even if idle machines)	
Test Group 1	200 same as above	
Test Group 2	20 same as above	
Total jobs in parent group "A" (both long and short)	At most 2000 (hard limit even if idle machines)	
Long running child subgroup of "A"	1000 (but if < 1000 short jobs, up to 2000 – sum of short + long should = 2000)	4 * 24
Short running child subgroup of "A"	1000 (but if < 1000 long jobs, up to 2000 – sum of short + long should = 2000)	4
External jobs	At least 40, but as many as idle machines .	2

Note that this is a flat hierarchy of groups, with the exception of parent group "A", which has two subgroups, one assigned to long running jobs, and another to short running jobs. Usually, there is sufficient demand of both long and short running jobs. However, occasionally, the demand for short jobs may go to zero. In this case, it is expected that all 2000 parent group "A" slots will start running "long" jobs, but there is a desire to preempt some number of these (100?) when the demand for short jobs comes back online.

Note these are absolute numbers, not ratios. If there are fewer machines, they would like the same proportion of job mix running. If there are additional machines that somehow show up in the pool, they

would like all the extra machines to run External Jobs, because all the other jobs access shared resources which they don't want to overload.

When they ask for multi core jobs, they are willing to "pay" by waiting longer for those jobs to match to machines.

If there are idle machines, they would like them to be used by External jobs when there is enough demand. But, they would like the ability to easily change these numbers from time to time. In general, demand for slots is pretty constant, but the mix between long and short jobs within parent group "A" is very dynamic.

External jobs are preempted after 2 hours for fairness. Generally, preemption is only used to limit run away jobs, and is set very high, with the assumption that the site knows the maximum legal runtime for a job, and anything over that runtime is an error, either in the job or on the machine.

Because of External jobs, they are willing to wait about 2 hours for the pool policy to be "unfair" and out of balance before External jobs should be preempted for fairness.