

openQA Project - coordination #12876

coordination # 32851 (Resolved): [tools][EPIC] Scheduling redesign

[epic] Offer a way for jobs to dynamically schedule children

2016-07-26 12:11 - coolo

Status:	Rejected	Start date:	2018-05-05
Priority:	Low	Due date:	
Assignee:	okurz	% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	Ready		
Difficulty:			
Description			
<p>As discussed with Vladimir, we want to offer an API in mmapi for tests to schedule children. The use case are mainly slenkins control nodes, but we can also use it to ease dynamic tests.</p> <p>The use case is: slenkins-nfs-control starts and installs the slenkins package into the VM.</p> <p>The test then figures from the given control file that it needs a nfs-server and a nfs-client with a given set of settings (currently implemented in the slenkins-import script).</p> <p>It asks mmapi to schedule two tests as children following its own settings, but with a set of new variables (each).</p> <p>If mmapi refuses to schedule them, the test waits and retries in a couple of minutes.</p> <p>If the jobs are scheduled, we get to the normal mmapi function to wait for the children to boot before we call into slenkins.</p> <p>openQA in return will schedule the children jobs with a high priority (to avoid dead locks), but will refuse to create any jobs if there are still scheduled jobs with that high priority (or some other means to check other children are taken).</p> <p>And we need to make sure the slenkins tests are only run on a very limited amount of workers - as they fork children basically the formula is $\text{number}(\text{slenkins tests}) = \text{number}(\text{available nodes with slenkins worker class}) - \text{max}(\text{possible children})$. Otherwise we can create a deadlock.</p> <p>The benefit is that slenkins tests don't require manual maintenance anymore, but only once registration.</p>			

History

#1 - 2016-07-27 13:44 - nadvornik

some additional ideas since yesterday:

mmapi tries to schedule jobs, it fails, the test tries again after some time

alternative to this: scheduling jobs will never fail, the jobs will appear as "scheduled", scheduler decides when to run them.

Scheduler starts the parallel test groups one after another, the situation when multiple groups are partially started can't happen and thus it is not handled currently.

The problem with parsing node file is that the job that parses it and creates other jobs is part of a group, but the scheduler does not know it from the beginning.

Options:

1 Improve scheduler to handle the situation when the first job from each group is already running and start the remaining jobs one group after another. This leads to the limitation $\text{number}(\text{slenkins tests}) = \text{number}(\text{available nodes with slenkins worker class}) - \text{max}(\text{possible children})$.

2 Make the job that parses the node file independent of the jobs it creates. All node files can be actually parsed in one job. This is compatible with current scheduler.

3 $\text{Number}(\text{slenkins tests}) = \text{number}(\text{available nodes with slenkins worker class}) / \text{max}(\text{possible children})$

- this is most conservative limitation, no deadlock is possible
- wastes resources, unless there is enough single-machine jobs to run on the remaining workers example: 20 workers, max 8 children per test, typically 3, $\text{number of slenkins tests} = 2$, typical load $2 * 3 = 6$ workers of 20

In any case, the jobs in a group should be created atomically, current api does not allow this. I think that clone_job.pl also suffers from this problem.

Options:

- create multiple jobs in one api call
- introduce new status "prepared", create jobs with this status, add api call to atomically switch whole group from prepared to scheduled

#2 - 2017-01-17 17:13 - okurz

is this done? can you say what is the status on this one?

#3 - 2017-11-18 07:12 - coolo

- Subject changed from *Offer a way for jobs to schedule children* to *[epic] Offer a way for jobs to schedule children*
- Status changed from *In Progress* to *New*
- Assignee deleted (*nadvornik*)
- Priority changed from *Normal* to *Low*
- Target version set to *Ready*

This is still very much wanted for more comfort in multi machine tests.

#4 - 2018-03-07 10:29 - EDiGiacinto

- Parent task set to #32851

Setting it as part of the scheduling enhancement ticket

#5 - 2018-05-09 12:47 - szarate

- Start date changed from *2016-07-26* to *2018-05-05*

due to changes in a related task

#6 - 2019-06-20 16:15 - okurz

- Category changed from *122* to *Feature requests*

#7 - 2020-04-01 13:52 - okurz

- Subject changed from *[epic] Offer a way for jobs to schedule children* to *[epic] Offer a way for jobs to dynamically schedule children*

#8 - 2020-04-01 18:21 - sebchlad

"Epic" is a good category here :-)

Jokes aside: [okurz](#): is this still valid ticket? we do not run SLEnkins test in openQA, right? Or am I miss something?
"The use case are mainly slenkins control nodes"

Am I correct that in the meantime some of the ideas got implemented?

I would be quite interested in refining this ticket in technical details and see what we could do about it, as we both agree (I assume) mm tests could benefit from some enhancements or perhaps there could be completely new take on them where we would end up with another way of doing mm test in openQA?

#9 - 2020-04-03 15:33 - okurz

- Status changed from *New* to *Rejected*
- Assignee set to *okurz*

True, SLEnkins test cases do not seem to be relevant nowadays. The ticket here is about *dynamically* scheduling children, i.e. define tests from already running test code. I have noted down the idea from this ticket now in the generalized feature request tracker [#65271](#) so we can close this one here.

#10 - 2020-10-12 13:38 - szarate

- Tracker changed from *action* to *coordination*

#11 - 2020-10-12 13:49 - szarate

See for the reason of tracker change: <http://mailman.suse.de/mailman/private/qa-sle/2020-October/002722.html>