

openQA Project - action #71809

Enable multi-machine jobs trigger without "isos post"

2020-09-24 08:46 - asmorodskyi

Status:	Workable	Start date:	2020-09-24
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			
Description			
In case of single job you have two options to trigger it : 1.) isos post (requires certain setup on target openQA) 2.) jobs post (can be done on any "random" openQA server because all variables are not calculated from flavors , job groups , machines etc. but coming with a call) In case of multi-machine job you have only first option . We need to find a way to trigger multi-machine jobs on target openQA host without require certain setup on it			
Related issues:			
Related to openQA Project - coordination #58184: [saga][epic][use case] full ...		Blocked	2018-03-23

History

#1 - 2020-09-24 08:48 - asmorodskyi

- Status changed from New to In Progress
- Assignee set to asmorodskyi

First would be do initial investigation to estimate efforts needed to achieve it

#2 - 2020-09-24 13:29 - asmorodskyi

- Related to coordination #58184: [saga][epic][use case] full version control awareness within openQA, e.g. user forks and branches, fully versioned test schedules and configuration settings added

#3 - 2020-09-24 15:55 - mkittler

Note that the same limitations also apply to directly chained dependencies. The problem is also apparent when using the clone job script (which uses "jobs post" internally). It would be good to take the directly chained dependencies and the clone job script into account as well.

It is not like "jobs post" can not deal with such dependencies at all. There are the `_PARALLEL_JOBS` and `_START_DIRECTLY_AFTER_JOBS` (and `_START_AFTER_JOBS`) parameters which allow to specify parent jobs by their IDs. So you can create the parents first and then the children using these parameters with the parent job IDs. The clone script already does this. The problem which affects parallel and directly chained dependencies is that this way of job scheduling is not atomic, e.g. the scheduler wouldn't wait with assigning the parent job to a worker until all child jobs have been created.

So far one can work around this issue by stopping the scheduler or relevant workers until all jobs have been created.

Note that we also have a "repair logic" for parallel jobs. So if one is already running while the rest of the cluster is still scheduled these scheduled jobs should be picked up by the scheduler without causing trouble (see `OpenQA::Scheduler::Model::Jobs::_pick_siblings_of_running`).

One possible solution is introducing another state, e.g. "new". Such jobs wouldn't be considered by the scheduler. There would be another API call to change the jobs from "new" to "scheduled" when all jobs have been created.

Or we allow scheduling all required jobs in one go via "jobs post". It might be a little bit hard to make a sane syntax for the parameter passing (which parameter would go to which job?) but it would be possible as well. We would also need to take care that all jobs are internally created using a database transaction with the right isolation level to avoid inconsistencies.

I've just seen that you've added this ticket to the "openQA Tests" project (and not just "openQA"). However, I don't see an easy way to solve the problem on test level.

#4 - 2020-09-28 14:30 - asmorodskyi

Hi Marius ,

thanks a lot for your input. I haven't thought about clone_job functionality , good hint I will check how it was solved there. I am still in "investigation phase" so don't have clear understanding how I want it to be done, but I like your idea with "New" job status.

I've just seen that you've added this ticket to the "openQA Tests" project (and not just "openQA"). However, I don't see an easy way to solve the problem on test level.

obviously I also don't see the way to solve this on test level. Let's move it in "openQA Project"

#5 - 2020-09-28 16:29 - okurz

- Project changed from openQA Tests to openQA Project

- Category set to Feature requests

- Target version set to future

ok, moving to "openQA Project". No keyword in summary is required but I add "future" as "Target version" to show that currently the SUSE QA Tools team does not plan to do this new feature, also because you are assigned anyway.

#6 - 2020-10-05 14:17 - asmorodskiy

so after some experimenting it appears that use of `_PARALLEL_JOBS` is right way of creating cluster jobs . It is really how openQA is doing it currently even in case of isos post . So in my case only difference would be that time frame between parent job creation and actual clustering it with child job would be "slightly bigger" if you compare single isos post and two separate jobs post calls it would be milliseconds vs seconds (minutes ?) . But I don't see any danger here because REF test suite has sync point with mutex . So in most worst case REF will wait till test default timeout here https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/wicked/locks_init.pm#L34 .

[mkittler](#) , [okurz](#) anything what I have missed from your POV ?

P.S. still plan to write some tests which will prove that my flow working as expected . But for this I most probably will need your help to figure out where . But now at least I have clear scenario which I need to cover :

1. Create random job A
2. Create job with `PARALLEL_WITH=A` and `_PARALLEL_JOB=` ==> check that openQA Scheduler manage to combine them into cluster

#7 - 2020-10-07 15:58 - mkittler

It is really how openQA is doing it currently even in case of isos post .

No, in case of iso post openQA does not rely on that variable. Yes, it would technically evaluate the variable when posting an ISO but in this case it actually relies on `PARALLEL_WITH`.

So in my case only difference would be that time frame between parent job creation and actual clustering it with child job would be "slightly bigger"

No, that's not the only difference. When posting an ISO, openQA is using a database transaction to ensure that the scheduler (and any other component) only see any of the new jobs when the whole cluster is scheduled. That works regardless of how long the scheduling takes and there's no race condition.

But I don't see any danger here because REF test suite has sync point with mutex .

Like I've mentioned in the chat, there's some repair logic for half-scheduled clusters. If the tests can cope with that as well there's likely really no danger here.

anything what I have missed from your POV ?

Despite the mentioned points I suppose you came to the right conclusion: Scheduling multi-machine tests with `_PARALLEL_JOB` is possible.

Create job with `PARALLEL_WITH=A`

When using jobs post setting the `PARALLEL_WITH` variable will have no effect (although it wouldn't hurt to add it). Setting `_PARALLEL_JOB` should be sufficient.

check that openQA Scheduler manage to combine them into cluster

You could delay scheduling the 2nd job intentionally so check how the scheduler behaves when one job of a parallel cluster is already running and another job still scheduled. That would be a manual test for the mentioned repair feature (which is likely not otherwise covered by openQA's own testsuite).

#8 - 2020-10-15 13:01 - asmorodskyi

https://gitlab.suse.de/wicked-maintainers/wicked-ci/-/merge_requests/70 - PR which solving problem within our project

#9 - 2020-10-16 10:12 - asmorodskyi

- *Status changed from In Progress to Workable*

- *Assignee deleted (asmorodskyi)*

https://gitlab.suse.de/wicked-maintainers/wicked-ci/-/merge_requests/70 - was merged . this ticket remain open because in terms of it I still would like to write some tests for scheduler . But need to switch to different topic now so will get back to this later