openQA Project - action #49535

Improve time to schedule a build

2019-03-21 09:30 - coolo

Status:	Resolved	**Start date:**	2019-03-21
Priority:	High	**Due date:**	
Assignee:	mkittler	**% Done:**	0%
Category:	Feature requests	**Estimated time:**	0.00 hour
Target version:	Current Sprint		
Difficulty:	hard		

Description

currently scheduling a SP1@x86_64 build takes > 180s, which was up to now our apache limit.

This really should be reduced a lot. Like all the cancelling/deprio logic could be moved to a minion job - just as the calculcation of blocked_by state I guess.

Related issues:

Related to openQA Project - action #45029: error 502 when triggering products...	**Resolved**	**2018-12-12**

History

#1 - 2019-03-21 09:46 - mkittler

- Assignee set to mkittler

- Target version changed from Ready to Current Sprint

I guess this is more important than refactoring the worker code.

#2 - 2019-03-21 10:32 - coolo

The blocked by calculation is touchy btw - we would need to make it three state. Currently it's blocked_by an id or NULL - which means the scheduler can pick it. Or we introduce yet another job state :)

#3 - 2019-03-21 14:06 - mkittler

I'm only wondering why we do the blocked_by calculation currently twice. One time directly after creating a job via create_from_settings and then again for each job after dealing with cycles, wrong parents, ...

If the last recalculation is required because it adds blocked-by IDs which couldn't be assigned in the first place, wouldn't that allow the scheduler to pick jobs accidentally (in the small period of time between the job creation and the final blocked-by calculation)? It that what you mean by *touchy*? With the right isolation level for the transaction that problem shouldn't occur, actually. Even if I remove the blocked-by calculation within create_from_settings and only do it in the end. (Everything is done in one transaction.)

Yet another job state to be sure we don't schedule jobs in an inconsistent state would make sense in general. Maybe SCHEDULING or ADDED? Or we just disable the scheduler (somehow) while job creation via posting an ISO is ongoing?

#4 - 2019-03-21 14:09 - mkittler

Likely another job state is the best. Then we don't need to care about the transaction isolation level and can do the blocked-by calculation only in a Minion job.

#5 - 2019-03-21 14:48 - coolo

Marius and me discussed this in detail - and I propose to make 'Scheduled Product' a high level DBIx class. And we create a new API that creates that and schedule a minion task to do the actual scheduling. So you can poll the status of the Scheduled Product - and it can be 'scheduling' or 'scheduled' and if it's scheduled you can query the errors it created.

This would make actually a nice addition as currently the scheduled products are extracted from audit log and the errors just appear in some log file. Plus it solves the timeout problem - clients simply poll if they care (or use the old API :)

#6 - 2019-03-22 16:34 - mkittler

- Status changed from New to In Progress

WIP-PR: https://github.com/os-autoinst/openQA/pull/2036

#7 - 2019-03-29 13:00 - mkittler

The PR is ready to merge. Further UI tweaks like filtering for failed scheduled products or showing all jobs related to one scheduled product can still be done.

#8 - 2019-04-15 13:31 - mkittler

- Status changed from In Progress to Resolved

This has been implemented and merged on the openQA-side and the PR for rsync.pl can be merged as soon as openQA is deployed.

#9 - 2019-08-13 09:11 - okurz

- Related to action #54179: Re-use YAML betweens different groups added

#10 - 2019-08-13 09:11 - okurz

- Related to deleted (action #54179: Re-use YAML betweens different groups)

#11 - 2019-08-13 09:12 - okurz

- Related to action #45029: error 502 when triggering products with rsync.pl added