openQA Project - coordination #93883

[epic] Speedup openQA coverage tests with running minion jobs synchronously using new upstream "perform_jobs_in_foreground" mojo function

2021-06-11 09:39 - okurz

Status:	Resolved		**Start date:**	2021-07-07
Priority:	Normal		**Due date:**	
Assignee:	mkittler		**% Done:**	100%
Category:	Feature requests		**Estimated time:**	0.00 hour
Target version:	Ready			
Difficulty:				

Description

Motivation

With #93086 we found a way to speedup t/10-jobs.t by running minion jobs synchronously from the main process instead of spawning subprocesses and collecting coverage from these because it takes ages. We should apply the same functionality to other test modules, e.g. t/42-df-based-cleanup.t, at best using the new (to come) upstream mojo feature "perform_jobs_in_foreground" *and* remove workaround for collecting coverage from forked subprocesses. In the end we might want to have only "unit tests" collecting coverage and ensuring complete statement coverage with these while having integration tests simulating more realistic multi-process behaviours.

Acceptance criteria

- **AC1:** All tests have been reviewed to apply the above
- **AC2:** We are not using the workaround to collect coverage from any subprocesses anywhere by default (could still have a switch which is disabled by default)

Suggestions

- Remove t/10-jobs.t from heavy jobs
- We could have an environment variable to switch between "perform_jobs" or "perform_jobs_in_foreground" and combine that with coverage analysis or not
- Remove the workarounds

Subtasks:		
action # 95170: Increase code coverage of critical component OpenQA::Worker::Job withou...		**Resolved**
action # 96561: Speed up `t/25-cache-service.t` by avoiding forking to run Minion jobs		**Resolved**
action # 96564: Speed up `t/ui/12-needle-edit.t` and `t/ui/21-admin-needles.t` by avoid...		**Resolved**

Related issues:			
Copied from openQA Project - action #93086: unstable test in openQA master t/...	**Resolved**	**2021-05-25**	**2021-06-23**

History

#1 - 2021-06-11 09:39 - okurz

- Copied from action #93086: unstable test in openQA master t/10-jobs.t exceeding runtime of 280s added

#2 - 2021-06-11 10:12 - tinita

- Subject changed from [epic Speedup openQA coverage tests with running minion jobs synchronously using new upstream "perform_jobs_in_foreground" mojo function to [epic] Speedup openQA coverage tests with running minion jobs synchronously using new upstream "perform_jobs_in_foreground" mojo function

#3 - 2021-07-29 11:16 - okurz

- Status changed from New to Workable

In the weekly estimation meeting we decided that for an epic it's actually "Workable" because we don't need to estimate the epic itself and the next task is simply to refine and create subtasks

#4 - 2021-07-30 13:02 - mkittler

- Status changed from Workable to In Progress

- Assignee set to mkittler

All suggestions have been implemented and there's a utility functions making use of "perform_jobs_in_foreground". So several tests have already been changed and resolving the ticket is just a matter of dealing with the remaining ones. I'll check which tests are remaining. If it can be done quickly, I'll change them right away, otherwise I'll create a ticket (and add it as sub task here).

#5 - 2021-07-30 14:03 - mkittler

- PR covering several tests: https://github.com/os-autoinst/openQA/pull/4095
- It does not cover:
  - t/25-cache-service.t: Performs Minion jobs very manually (without existing helper).
  - t/api/02-iso-download.t: Not sure in which way it actually runs Minion jobs. I created a separate PR to check that: https://github.com/os-autoinst/openQA/pull/4096
  - t/ui/12-needle-edit.t and t/ui/21-admin-needles.t: They start GRU as sub process explicitly via with_gru parameter.

#6 - 2021-07-31 04:09 - openqa_review

Setting due date based on mean cycle time of SUSE QE Tools

#7 - 2021-08-01 04:09 - openqa_review

Setting due date based on mean cycle time of SUSE QE Tools

#8 - 2021-08-01 09:06 - okurz

mkittler what the bot wants to tell you is basically: don't directly "work" on epics :)

#9 - 2021-08-01 09:08 - okurz

- Status changed from In Progress to Feedback

Both PRs merged, what next?

#10 - 2021-08-04 10:22 - mkittler

I've been creating sub-tasks for the remaining tests (because changing them maybe takes a little effort).

#11 - 2021-08-11 13:35 - mkittler

- Status changed from Feedback to Blocked

#12 - 2021-09-13 13:50 - mkittler

- Status changed from Blocked to Resolved

All sub tasks have been resolved so all tests are converted now.

#13 - 2021-09-13 21:32 - okurz

awesome, thank you!