

openQA Project - action #72199

check and reduce runtime of t/44-scripts.t

2020-10-02 09:11 - okurz

Status:	Workable	Start date:	
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			

Description

Observation

<https://app.circleci.com/pipelines/github/os-autoinst/openQA/4393/workflows/73728ebb-2783-487d-bf09-79f3ee2270ef/jobs/42264> shows

```
[07:25:28] t/44-scripts.t ..... 16/? Bailout called. Further testing stopped: test exceeds runtime limit of '120' seconds
FAILED--Further testing stopped: test exceeds runtime limit of '120' seconds
```

I could reproduce the problem locally, e.g. from lord.arch:

```
t/44-scripts.t ..
ok 1 - Calling 'client --help' returns exit code 0
ok 2 - Calling 'create_admin --help' returns exit code 0
ok 3 - Calling 'dump_templates --help' returns exit code 0
ok 4 - Calling 'initdb --help' returns exit code 0
ok 5 - Calling 'load_templates --help' returns exit code 0
ok 6 - Calling 'modify_needle --help' returns exit code 0
ok 7 - Calling 'openqa --help' returns exit code 0
ok 8 - Calling 'openqa-cli --help' returns exit code 0
ok 9 - Calling 'openqa-clone-job --help' returns exit code 0
ok 10 - Calling 'openqa-label-all --help' returns exit code 0
ok 11 - Calling 'openqa-livehandler --help' returns exit code 0
ok 12 - Calling 'openqa-scheduler --help' returns exit code 0
ok 13 - Calling 'openqa-validate-yaml --help' returns exit code 0
ok 14 - Calling 'openqa-websockets --help' returns exit code 0
ok 15 - Calling 'openqa-workercache --help' returns exit code 0
Bailout called. Further testing stopped: test exceeds runtime limit of '20' seconds
Bail out! test exceeds runtime limit of '20' seconds
FAILED--Further testing stopped: test exceeds runtime limit of '20' seconds
## count_fail_ratio: Run: 5. Fails: 5. Fail ratio 100%
```

Prepared <https://github.com/os-autoinst/openQA/pull/3435> to cover this for now with higher timeout. In general it is unfortunate that our scripts take that long already to just show a "--help" screen. This is something that could be improved.

Acceptance criteria

- **AC1:** The test runtime in t/44-scripts.t is significantly lower than the current value of 60s

Suggestions

- Run t/44-scripts.t locally
- Check if runtime is long or not in your environment. If needed compare different environments.
- Reproduce the long runtime and identify which individual scripts take how long, e.g. run `prove -v -l t/44-scripts.t | ts`
- Either optimize individual cases that take long or identify why the overall runtime might be long.

Related issues:

Copied from openQA Project - action #72127: check and/or reduce runtime of t/...

Resolved

2020-09-30

History

#1 - 2020-10-02 09:11 - okurz

- Copied from action #72127: check and/or reduce runtime of t/44-scripts.t added

#2 - 2020-10-02 09:16 - okurz

- Due date set to 2020-10-16

- Status changed from Workable to Feedback

- Assignee set to okurz

hm. I just checked again and found the runtime locally to be consistently only 11s, so not that long. Also in lord.arch, which is based on rotating disk and slower but still shows reproducibly 25s instead of the formerly very long runtimes. Maybe there was intermittent different behaviour based on some package updates.

I will give this some time and just recheck again at a later time how results within circleCI are. Maybe we can just reduce the timeout again.

From lord.arch:

```
## Run 400
t/44-scripts.t ..
ok 1 - Calling 'client --help' returns exit code 0
ok 2 - Calling 'create_admin --help' returns exit code 0
ok 3 - Calling 'dump_templates --help' returns exit code 0
ok 4 - Calling 'initdb --help' returns exit code 0
ok 5 - Calling 'load_templates --help' returns exit code 0
ok 6 - Calling 'modify_needle --help' returns exit code 0
ok 7 - Calling 'openqa --help' returns exit code 0
ok 8 - Calling 'openqa-cli --help' returns exit code 0
ok 9 - Calling 'openqa-clone-job --help' returns exit code 0
ok 10 - Calling 'openqa-label-all --help' returns exit code 0
ok 11 - Calling 'openqa-livehandler --help' returns exit code 0
ok 12 - Calling 'openqa-scheduler --help' returns exit code 0
ok 13 - Calling 'openqa-validate-yaml --help' returns exit code 0
ok 14 - Calling 'openqa-websockets --help' returns exit code 0
ok 15 - Calling 'openqa-workercache --help' returns exit code 0
ok 16 - Calling 'upgradedb --help' returns exit code 0
ok 17 - Calling 'worker --help' returns exit code 0
1..17
ok
All tests successful.
Files=1, Tests=17, 26 wallclock secs ( 0.03 usr  0.01 sys + 19.24 cusr  1.56 csys = 20.84 CPU)
Result: PASS
## count_fail_ratio: Run: 400. Fails: 0. Fail ratio 0%

real    169m6.675s
user    130m0.700s
sys     10m8.841s
```

#3 - 2020-10-08 18:55 - okurz

- Due date deleted (2020-10-16)

- Status changed from Feedback to Workable

- Assignee deleted (okurz)

Looking into the last 7 circleCI pipelines on master branch following <https://app.circleci.com/pipelines/github/os-autoinst/openQA?branch=master> I found times 65s, 52s, 51s, 51s, 53s, 54s, 60s so I think we can actually lower the timeout again: <https://github.com/os-autoinst/openQA/pull/3454>

To find out if there are any identifiable parts that contribute significantly to the long runtime I did:

```
$ unbuffer perl t/44-scripts.t | ts -i '%H:%M:%S' | ts -s '%H:%M:%S'
00:00:00.450432 00:00:00.446796 ok 1 - Calling 'client --help' returns exit code 0
00:00:01.595675 00:00:01.145263 ok 2 - Calling 'create_admin --help' returns exit code 0
00:00:01.947218 00:00:00.351537 ok 3 - Calling 'dump_templates --help' returns exit code 0
00:00:03.106874 00:00:01.159691 ok 4 - Calling 'initdb --help' returns exit code 0
00:00:03.376555 00:00:00.269689 ok 5 - Calling 'load_templates --help' returns exit code 0
00:00:03.487005 00:00:00.110453 ok 6 - Calling 'modify_needle --help' returns exit code 0
00:00:04.735938 00:00:01.248890 ok 7 - Calling 'openqa --help' returns exit code 0
00:00:05.005254 00:00:00.269348 ok 8 - Calling 'openqa-cli --help' returns exit code 0
00:00:05.299443 00:00:00.294165 ok 9 - Calling 'openqa-clone-job --help' returns exit code 0
00:00:05.500869 00:00:00.201414 ok 10 - Calling 'openqa-label-all --help' returns exit code 0
00:00:06.735150 00:00:01.234307 ok 11 - Calling 'openqa-livehandler --help' returns exit code 0
```

```
00:00:07.935287 00:00:01.200119 ok 12 - Calling 'openqa-scheduler --help' returns exit code 0
00:00:08.189366 00:00:00.254079 ok 13 - Calling 'openqa-validate-yaml --help' returns exit code 0
00:00:09.389636 00:00:01.200300 ok 14 - Calling 'openqa-websockets --help' returns exit code 0
00:00:09.845217 00:00:00.455582 ok 15 - Calling 'openqa-workercache --help' returns exit code 0
00:00:11.019697 00:00:01.174448 ok 16 - Calling 'upgradedb --help' returns exit code 0
00:00:11.390057 00:00:00.370397 ok 17 - Calling 'worker --help' returns exit code 0
```

For explanation: unbuffer makes sure that every line is returned as soon as it is printed so that ts can display a sensible timestamp. The first timestamp is the runtime of the complete script whereas the second timestamp is the relative runtime difference to the previous line.

To me it seems that the perl scripts that use the mojo framework and especially our services using that each take about 1.2s of loading time which is quite significant but no single script stands out. With this I will unassign from the ticket again and expect that eventually someone will have a good idea how to optimize.

#4 - 2020-11-14 22:13 - okurz

- *Target version deleted (Ready)*

#5 - 2020-11-14 22:13 - okurz

- *Target version set to future*