

openQA Project - action #25814

coordination # 34357 (New): [epic] Improve openQA performance

coordination # 65402 (Resolved): [epic] Revamp test details page to improve loading times and prevent timeouts

load job page, e.g. test details, only on demand

2017-10-06 08:14 - okurz

Status: Resolved	Start date: 2020-04-07
Priority: Normal	Due date:
Assignee: mkittler	% Done: 0%
Category: Feature requests	Estimated time: 0.00 hour
Target version:	
Difficulty:	
Description	
Goal	
Improve loading time on test details	
acceptance criteria	
<ul style="list-style-type: none">• AC1: Initial content of test details are showing up faster• AC2: server performance is not decreased significantly	
tasks	
<ul style="list-style-type: none">• suggestion: load content of each tab only on click on the tab header, e.g. with some javascript magic, similar to what we do for investigation and "next & previous" tabs	
Related issues:	
Related to openQA Project - action #20606: [tools][sprint 201709.2][sprint 20...	Resolved 2017-07-19
Related to openQA Project - action #13920: Showing context builds test result...	Resolved 2016-09-27
Related to openQA Project - action #19388: Show bugrefs/labels/comments and f...	Resolved 2017-05-26
Related to openQA Project - action #16556: Need to display "incomplete" jobs ...	Resolved 2017-02-07
Related to openQA Project - action #32611: job details in browser windows do ...	Resolved 2017-08-25

History

#1 - 2017-10-06 08:14 - okurz

mkittler, szarate: I think this is strongly linked to [#20606](#) , isn't it?

#2 - 2017-10-06 08:15 - okurz

- Related to action #20606: [tools][sprint 201709.2][sprint 201710.1]Disable live log view by default added

#3 - 2017-10-06 08:15 - okurz

- Related to action #13920: Showing context builds test result in the view of current "Previous Results" added

#4 - 2017-10-06 08:15 - okurz

- Related to action #19388: Show bugrefs/labels/comments and filters in /tests route like /tests/overview and #previous added

#5 - 2017-10-06 08:15 - okurz

- Related to action #16556: Need to display "incomplete" jobs in "Previous results" added

#6 - 2017-10-12 08:10 - mkittler

- Assignee set to mkittler

#7 - 2017-10-12 08:16 - mkittler

[okurz](#): Yes, my implementation of [#20606](#) covers already this ticket for the 'Live Preview' tab.

Since 'Details' is default anyways, only the tabs 'Logs & Assets', 'Settings', 'Comments' and 'Previous results' remaining. I doubt that anything else than 'Previous results' would give us a noticeable performance benefit, but I can check of course.

#8 - 2017-10-17 14:28 - mkittler

- Priority changed from Normal to Low

I did some profiling and like expected the heaviest tab is 'Details' which we want to be displayed by default anyways. Reading the test modules takes alone 35 % of the time. Rendering the job modules ('Details' tab) makes another 40 %. Rendering overall page elements another 10 %.

So I don't think there is currently much to gain from loading 'Logs & Assets', 'Settings', 'Comments' and 'Previous results' only on demand. I would let this ticket open because it maybe makes sense when we add more features to tests page (like <https://progress.opensuse.org/issues/25680>).

#9 - 2018-03-23 13:58 - mkittler

- Assignee deleted (mkittler)

#10 - 2018-04-05 14:42 - mkittler

- Parent task set to #34357

#11 - 2018-04-08 09:31 - okurz

mkittler wrote:

I did some profiling and like expected the heaviest tab is 'Details' which we want to be displayed by default anyways. Reading the test modules takes alone 35 % of the time. Rendering the job modules ('Details' tab) makes another 40 %. Rendering overall page elements another 10 %.

So I don't think there is currently much to gain from loading 'Logs & Assets', 'Settings', 'Comments' and 'Previous results' only on demand. I would let this ticket open because it maybe makes sense when we add more features to tests page (like <https://progress.opensuse.org/issues/25680>).

But this ticket mentions "test details" already. So let me refine my suggestion:

- Lazy load the details tab because it is in most cases the one with longest loading time
- Optional: Lazy load the other tabs

Still "Low"?

#12 - 2019-08-20 21:25 - okurz

- Subject changed from load test details, e.g. previous jobs, only on demand to load job page, e.g. test details, only on demand

"next & previous" load on demand now, the details itself would be cool next candidate

#13 - 2019-12-11 14:59 - okurz

- Description updated

- Status changed from New to Workable

#14 - 2020-04-07 09:07 - okurz

- Related to action #32611: job details in browser windows do not automatically jump from "assigned" to "running" when they start - take 2 added

#15 - 2020-04-07 09:39 - okurz

I have experimented with this myself recently when I was adding the "investigation" feature. But for now I ended up keeping the behaviour to always show the "Details" tab even in case there are no module results at all because we just show the log content, already asynchronously, in case of incompletes.

QA tools team discussed this 2020-04-07 and we have the following ideas:

- By default we should only load content synchronously that comes from the database, everything loading from filesystem should be done asynchronously or "on demand", either later by requests from the initial page loaded or by user actions, e.g. when clicking on tabs or buttons
- Of course loading from "fast" (expensive) storage like SSD or NVMe can help but shouldn't be relied upon
- Also see related ticket [#32611](#) about correct page self-refreshes
- We have a "reason" in every job that we already use successfully for incompletes. Our initial goal was to use the reason when the worker did not manage to upload any log file at all. Then we extended to always use the reason for every incomplete job. We can extend that concept to every not-passed job as long as we limit the size that we store in the database, e.g. cut reason string at a reasonable, human-readable length, e.g. 120 characters. This way we can also load the reason synchronously.

#16 - 2020-04-07 15:40 - mkittler

- Start date changed from 2017-10-06 to 2020-04-07

due to changes in a related task

#17 - 2020-04-07 15:40 - mkittler

- Parent task changed from #34357 to #65402

#18 - 2020-04-07 15:46 - mkittler

- Priority changed from Low to Normal

I'm raising the priority because the timeouts are annoying. (Like explained in the parent task: *Only* loading the details on demand is not sufficient for preventing timeouts. The query needed to be splitted as well.)

#19 - 2020-04-07 19:29 - okurz

maybe, but please consider any "ltp" jobs out of scope for this ticket. There are LTP specific tickets.

#20 - 2020-04-08 13:09 - mkittler

Those are not just ltp jobs, e.g. on o3 these are currently the biggest jobs:

```
openqa=# select id, test, result_size, (result_size / 1024 / 1024 / 1024) as result_size_gb from jobs where r
result_size is not null order by result_size desc limit 20;
```

id	test	result_size	result_size_gb
1224142	gnome	1438745990	1
1219900	gnome	1064366118	0
1219897	gnome	1022172477	0
1226040	textmode	1016078684	0
1224146	textmode	906820427	0
1224144	install_with_updates_kde	769961168	0
1216733	gnome	671770211	0
1224147	cryptlvm	666113740	0
1226041	cryptlvm	661436222	0
1216736	gnome	653178283	0
1217807	install_with_updates_kde	646933169	0
1217806	install_with_updates_gnome	641942140	0
1223483	kde-wayland	619408445	0
1218735	kde-wayland	613924168	0
1226065	RAID5	609392278	0
1227140	RAID5	597121762	0
1217809	textmode	591906838	0
1223563	extra_tests_in_textmode	583677747	0
1218229	gnome	582889441	0
1221359	gnome	573328610	0

(20 Zeilen)

The top ones can not be accessed because they time out.

#21 - 2020-04-15 09:41 - mkittler

- Status changed from Workable to In Progress

- Assignee set to mkittler

- Target version set to Current Sprint

#22 - 2020-04-17 16:25 - mkittler

PR: <https://github.com/os-autoinst/openQA/pull/2943>

The PR does not include fragmented loading of the "Details" or "External results" tab so it is not solving the timeout problem yet. However, I suppose it already fulfills the acceptance criteria.

#23 - 2020-04-21 17:38 - okurz

- Status changed from In Progress to Resolved

async details loading is live on o3. And it's awesome ☺☺

Yes, I see all ACs fulfilled. Thanks a lot.

#24 - 2020-04-23 11:47 - mkittler

- *Target version deleted (Current Sprint)*

Regarding the timeout problem: Maybe it is the easiest to simply increase the timeout instead of making the JavaScript handling this. There's already a [PR](#) so let's see whether it helps.