openQA Infrastructure - action #65975

Monitoring for "scheduled but not executed" (was: perl-Mojolicious-8.37 broke WS connection for (some?) workers)

2020-04-22 10:16 - nicksinger

Status:	Resolved	**Start date:**	2020-04-22
Priority:	Low	**Due date:**	2020-10-07
Assignee:	okurz	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Ready		

Description

Motivation

There is a plan to provide more metrics within SUSE, e.g. see https://confluence.suse.com/pages/viewpage.action?pageId=156401717#RalfUnger%E2%80%99sHome-Ideasby , and as we already use grafana we should be able to gather data from there.

Acceptance criteria

- **AC1:** A notification on alerts is sent over the usual channels when there are scheduled jobs older than defined limits, e.g. 2 days
- **AC2:** A notification on alerts is sent over the usual channels when there are more than X scheduled jobs with less than Y running jobs, e.g. X > 500 scheduled (not counting blocked), Y < 20

Suggestions

Time to starting of tests on openqa.suse.de (how long does it take maximum that a test on openqa.suse.de starts; purpose: How users/teams/departments can integrate openQA tests into their own processes; data source: openQA database queries or existing data on https://stats.openqa-monitor.qa.suse.de ;presentation: time ranges as statistical values, e.g. mean+-std, median, percentiles as numbers as well as graphs for progress)

Original

Today morning, after deployment, I got pinged in RC that the workers don't execute jobs anymore. Looking at the logs I saw:

```
Apr 22 10:59:05 openqaworker2 worker[21179]: [info] [pid:21179] Registering with openQA openqa.suse.de
Apr 22 10:59:05 openqaworker2 worker[21179]: [info] [pid:21179] Establishing ws connection via ws://openqa.suse.de/api/v1/ws/1070
Apr 22 10:59:05 openqaworker2 worker[21179]: [info] [pid:21179] Registered and connected via websockets with openQA host openqa.suse.de and worker ID 1070
Apr 22 10:59:11 openqaworker2 worker[21179]: [warn] [pid:21179] Websocket connection to http://openqa.suse.de/api/v1/ws/1070 finished by remote side with code 1006, no reason - trying again in 10 seconds
```

This was repeated all the time. We "fixed" the problem by downgrading from perl-Mojolicious-8.37 to perl-Mojolicious-8.36 on OSD and restarting the openqa-websockets service.

Related issues:

Has duplicate openQA Infrastructure - action #68236: Provide metric about "ti...	**Rejected**	**2020-06-19**

History

#1 - 2020-04-22 11:05 - mkittler

This can be reproduced by running the fullstack test locally. Before upgrading to a fixed version of Mojolicious I can check locally whether it works again. If it works with the fixed version we can deploy it and also update the dependency within the CI.

#2 - 2020-04-22 11:13 - okurz

- Priority changed from Normal to High

https://github.com/os-autoinst/openQA/pull/2972 is the corresponding upgrade in the dependency file for circle CI tests. There is already a failure that

shows similar symptoms.

PR was still open and is now closed unmerged but it shows what has been *already* installed by automatic upgrades, as we don't have a "stable/tested" repo for all upgraded dependencies of openQA. all o3 and osd workers upgrade from devel:openQA and devel:openQA:Leap:15.1 automatically if the state in github master is fine, osd upgrades if o3 is "fine" and if even the openQA-in-openQA tests are fine. As this problem does not manifest itself as anything that fails in these tests we did not see the problem yet. This might also be because all the mentioned tests are actually not running with the upgraded version though. Also we can apply "Five Whys" and improve something on every level □□ . This is something I would like to do with you guys in a synchronous meeting, e.g. retrospective next week

#3 - 2020-04-22 12:34 - kraih

I've investigated and found two upstream bugs in Mojolicious. The first should be the cause for this WebSocket issue and will be resolved with Mojolicious 8.39. https://github.com/mojolicious/mojo/compare/987640df12c9...b2ad468d0725

#4 - 2020-04-23 12:06 - mkittler

- Status changed from New to In Progress

- Priority changed from High to Normal

With 8.39 I can not reproduce the problem within the fullstack test locally anymore. So it seems to fix the most severe problem.

However, kraih mentioned in the chat:

> there was another timeout bug upstream in Mojolicious, so maybe we should depend on 8.40

#5 - 2020-04-28 09:46 - okurz

- Assignee set to okurz

In the QA tools retro we conducted a "Five Whys" session and identified the following ideas to follow on:

- We can suggest to kraih that he can test openQA tests against any pending upgrade of packages he maintains, e.g. perl-Mojolicous :)
- Improve monitoring to look into "scheduled but not executed":
  - detect scheduled jobs older than defined limit, example: 2 days
  - if there are more than X scheduled jobs with less than Y running jobs, alert. Example selections: X > 500 scheduled (not counting blocked), Y < 20
- We can have a nightly CI job that upgrades dependencies additionally from CPAN and runs openQA tests

#6 - 2020-05-02 18:02 - okurz

- Subject changed from perl-Mojolicious-8.37 broke WS connection for (some?) workers to Monitoring for "scheduled but not executed" (was: perl-Mojolicious-8.37 broke WS connection for (some?) workers)

- Description updated

#7 - 2020-05-02 18:56 - okurz

- Status changed from In Progress to Workable

- Assignee deleted (okurz)

Adapted the ticket to capture the one idea, added the other idea in #65271#note-14

#8 - 2020-06-19 08:30 - okurz

- Has duplicate action #68236: Provide metric about "time to start of tests" on OSD added

#9 - 2020-06-19 08:31 - okurz

- Description updated

#10 - 2020-06-23 07:15 - okurz

I recommend we start with extending our custom SQL queries in telegraf in
https://gitlab.suse.de/openqa/salt-states-openqa/-/blob/master/openqa/telegraf-webui.conf#L76

I tried

```
openqa=> select id,t_started,t_created,state,machine,arch,test from jobs where state != 'done' and state != 'c
ancelled' order by t_created limit 30;
   id    | t_started |      t_created      |   state   |         machine          |  arch   |
          test
---------+-----------+---------------------+-----------+--------------------------+---------+------------------
-----------------------------------------
 4296324 |           | 2020-05-29 14:46:21 | scheduled | ipmi-tyrion              | x86_64  | install_ltp_barem
etal_mlx
…
 4327400 |           | 2020-06-08 15:02:57 | scheduled | ipmi-tyrion              | x86_64  | ltp_syscalls_bare
metal
 4338945 |           | 2020-06-11 02:35:13 | scheduled | ppc64le-hmc-single-disk | ppc64le | gw-online_sles15s
p1_pscc_basesys-srv_all_full_zypper_spvm
…
 4342369 |           | 2020-06-12 01:11:04 | scheduled | ipmi-tyrion              | x86_64  | ltp_syscalls_bare
metal
 4366399 |           | 2020-06-18 13:06:33 | scheduled | 64bit                    | x86_64  | hpc_DELTA_slurm_s
lave00
…
 4367363 |           | 2020-06-18 17:07:48 | scheduled | 64bit                    | x86_64  | hpc_DELTA_slurm_a
ccounting
 4367523 |           | 2020-06-18 18:08:13 | scheduled | 64bit                    | x86_64  | hpc_DELTA_slurm_a
ccounting_supportserver
```

this shows a lost of tests that have been scheduled a bit longer ago and are waiting to be executed. The first jobs are scheduled against a machine that does not seem to have any workers with matching WORKER_CLASS behind so I would handle them separately and exclude them here for now. Also jobs like https://openqa.suse.de/tests/4366399 are a bit special because they have been scheduled with WORKER_CLASS=openqaworker-arm-3 but ARCH=x86_64 which also does not make any sense. I think it can still be beneficial to alert about these though. I am thinking about ignoring outliers so e.g. check median(NOW() - t_created) against a threshold. https://www.postgresql.org/docs/current/functions-aggregate.html#FUNCTIONS-ORDEREDSET-TABLE describes calculating percentiles in postgres.

Also experimenting with calculating in grafana, e.g.

```
SELECT (last(scheduled)+last(blocked))/last("running") FROM "openqa_jobs" WHERE $timeFilter GROUP BY time($__i
nterval) fill(previous)
```

this for example shows a high value exceeding 200 on 2020-04-25 which might link to the original event but I am not convinced of this metric. Does not seem to show events like we had on 2020-06-16 when workers refused to execute tests.

EDIT: Discussed with szarate as I had a call with him anyway :) He favors in particular graphs of the scheduled vs. worker class. A derived single-value metric would then be the "start time per worker class". However we do not store the specific worker class in the jobs table of the database. For a start the "machine" might suffice. Too specific worker class settings, e.g. in test suites, are likely a test configuration smell that should be avoided.

#11 - 2020-06-24 06:25 - okurz

- Status changed from Workable to In Progress

- Assignee set to okurz

With https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/326 and followups we have a "job age" collected from telegraf. Maybe with my experiments I messed up a bit. Struggling with influx data.
postgresql,arch=x86_64,db=postgres,host=openqa,machine=64bit,server=dbname\=openqa\ host\=localhost\ user\=telegraf job_age_p100=9550i,job_age_p50=9522i,job_age_p90=9548i,job_age_p99=9550i 1592978793000000000 looks like key=value is "job_age_p100=…" and has tags for "arch" and "machine" but I can not see recent tags in manual influxdb queries and in grafana I can also only see the values but not select on the tags. maybe https://github.com/influxdata/influxdb/issues/2615 is the problem as I did the mistake to not specify the tag values previously. So with https://gitlab.suse.de/openqa/salt-states-openqa/-/commit/8e90ee9cb8ab446430da3234fb942707046ef441 I change the tag name to "test_machine" and delete arch because maybe we can live without, machine should be enough.

My primary target for the "job age" is to prevent these (embarassing) moments when QA engineers like you need to ping us in chat and ask "hey, are all arm workers down since 3 days and nobody noticed?". But not only that. For example the 100% percentile can tell us: "There are tests scheduled against a worker class but there is no such worker" -> inform tester about misconfiguration, 99% percentile "these jobs are taking ages and we always need to wait for these" -> inform tester to optimize these cases as this blocks the reporting, 90% percentile "this is the bunch that takes longer than most of the rest" -> invest here with additional worker ressources, 50% percentile "this is about the time you can expect an arbitrary job to wait before it gets picked up" -> good to know for everyone to know what to expect.

The job age is off by 2h already coming from the SQL query. I need to account for timezone diff. Fixed with https://gitlab.suse.de/openqa/salt-states-openqa/-/commit/8256d576178a3c50f89ad4b1fe44aee18cb23f2f

Added two singlestat panels to https://stats.openqa-monitor.qa.suse.de/d/7W06NBWGk/job-age?orgId=1 for "best" and "worst" case along with alerts. Now the dashboard also nicely shows that the "worst" case is in red state due to misconfigured jobs:

```
openqa=> select arch,machine,percentile_cont(.5) within group (order by age) as p50, percentile_disc(.9) withi
```

```
n group (order by age) as p90, percentile_disc(.99) within group (order by age) as p99, percentile_disc(1) wit
hin group (order by age) as p100 from (select id,state,machine,arch,test,(NOW() - t_created) as age from jobs
where state != 'done' and state != 'cancelled' order by age) as job_ages group by machine,arch;
  arch   |         machine          |           p50           |           p90           |           p99
   |          p100
---------+--------------------------+-------------------------+-------------------------+----------------------
---+-------------------------
 x86_64  | 64bit                    | 07:01:47.756841         | 08:01:18.756841         | 08:01:55.756841
   | 08:02:19.756841
 aarch64 | aarch64                  | 04:02:00.756841         | 04:02:00.756841         | 04:02:00.756841
   | 04:02:00.756841
 x86_64  | az_Standard_A2_v2        | 09:02:09.756841         | 09:02:10.756841         | 09:02:10.756841
   | 09:02:10.756841
 x86_64  | ec2_t2.large             | 09:02:09.756841         | 09:02:11.756841         | 09:02:11.756841
   | 09:02:11.756841
 x86_64  | gce_n1_standard_2        | 09:02:10.756841         | 09:02:11.756841         | 09:02:11.756841
   | 09:02:11.756841
 x86_64  | ipmi-tyrion              | 15 days 07:05:07.756841 | 20 days 08:28:50.756841 | 20 days 08:28:50.7568
41 | 20 days 08:28:50.756841
 ppc64le | ppc64le-hmc-single-disk | 12 days 19:32:50.756841 | 12 days 19:32:50.756841 | 12 days 19:32:50.7568
41 | 12 days 19:32:50.756841
 s390x   | zkvm                     | 06:02:31.756841         | 06:02:38.756841         | 06:02:38.756841
   | 06:02:38.756841
(8 rows)
```

So I cleaned up the old ones:

```
delete from jobs where state != 'done' and state != 'cancelled' and t_created < (now() - interval '10d');
```

Created https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/329 to save the complete new dashboard and deploy with salt.

#12 - 2020-06-25 17:45 - okurz

- Status changed from In Progress to Feedback

Apparently there are some worker classes which are really limited triggering alerts which most likely we do not want to act on, e.g.

```
$ openqa-cli api --osd jobs state=scheduled | jq -r '.jobs | .[] | .id,.name'
4390260
sle-12-SP4-EC2-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-publiccloud@ec2_t2.large
4390263
sle-12-SP4-AZURE-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-publiccloud@az_Standard_A2_v2
4390264
sle-12-SP5-GCE-BYOS-Updates-x86_64-Build20200625-2-publiccloud_upload_img@gce_n1_standard_2
4390265
sle-12-SP5-GCE-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-docker-publiccloud@gce_n1_standard_2
4390266
sle-12-SP5-GCE-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-publiccloud@gce_n1_standard_2
4390267
sle-12-SP5-EC2-BYOS-Updates-x86_64-Build20200625-2-publiccloud_upload_img@ec2_t2.large
4390268
sle-12-SP5-EC2-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-docker-publiccloud@ec2_t2.large
4390269
sle-12-SP5-EC2-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-publiccloud@ec2_t2.large
4390270
sle-12-SP5-AZURE-BYOS-Updates-x86_64-Build20200625-2-publiccloud_upload_img@az_Standard_A2_v2
4390271
sle-12-SP5-AZURE-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-docker-publiccloud@az_Standard_A2_v2
4390272
sle-12-SP5-AZURE-BYOS-Updates-x86_64-Build20200625-2-mau-extratests-publiccloud@az_Standard_A2_v2
```

are cloud image jobs that have been scheduled since more than 4h ago. Will bump the alert limit to 8h but I fear it will loose its original purpose with this. I guess we need to find something better. And I don't think alert per machine is easily feasible. Feedback would be appreciated.

EDIT: 2020-06-28: Apologies for the many recent alerts on openqa@suse.de . I understood that this apparently overwhelms all of you and hence you missed other alerts. As a remedy I know changed the alert notification channel to go to okurz@suse.de only for the "job age (median)" panel

#13 - 2020-07-29 07:09 - okurz

- Target version set to Ready

#14 - 2020-07-29 19:08 - okurz

- Priority changed from Normal to Low

Every time I receive the alert it's actually wrongly configured jobs. This is for the "max" class, the other alert still disabled. Not exactly sure how to proceed. Maybe I will come up with a good idea some time later.

#15 - 2020-08-11 11:23 - okurz

- Description updated

#16 - 2020-08-11 11:25 - okurz

- Due date set to 2020-10-01

I have re-enabled the alert for "Job age (scheduled) (median)" as it did not trigger for more than a month and I think if it would, then it would be valid. We need to handle this again whenever it happens but I can monitor myself for the next two months or so.

#17 - 2020-09-08 12:48 - okurz

The alert failed again after more builds of SLE15SP3 were triggered but due to #69727 the capacity was limited. As also users asked about the limited testing capacity the alert was valid and telling us what we needed to know anyway. After I could bring up openqaworker-arm-3 again and seeing that https://stats.openqa-monitor.qa.suse.de/d/7W06NBWGk/job-age?orgId=1&panelId=5 looks fine I unpaused the alert now but will still keep myself as the recipient as I know my Pappenheimer :)

#18 - 2020-09-11 06:11 - okurz

the alert triggered again because currently there are a lot of aarch64 HPC jobs running and scheduled but nearly no other. I don't think anything is wrong in particular here though. Looking for better ideas what to do :(

#19 - 2020-09-18 07:40 - okurz

bumped values a bit: https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/356

#20 - 2020-09-29 14:37 - okurz

- Due date changed from 2020-10-01 to 2020-10-07

Proposing the enablement of the median job age alert for osd-admins@suse.de with
https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/362

#21 - 2020-10-08 19:04 - okurz

- Status changed from Feedback to Resolved

The MR is merged and the alert is active with sending messages to osd-admins@suse.de . It actually triggered because someone scheduled jobs that could never run because there was no matching worker class. I saw the alert but pretended I would not have seen it and I was happy to see that mkittler took the alert serious and addressed it :) With this I am now confident that my immediate work is done and we can close this ticket as "Resolved".