

openQA Project - action #81828

Jobs run into timeout_exceeded after the 'chattr' call, no output until timeout, auto_review:"(?s)Refusing to save an empty state file to avoid overwriting a useful one.*Result: timeout":retry

2021-01-06 09:54 - dzedro

Status:	New	Start date:	2021-01-06
Priority:	Low	Due date:	
Assignee:		% Done:	0%
Category:	Concrete Bugs	Estimated time:	0.00 hour
Target version:	future		
Difficulty:			

Description

Observation

<https://openqa.suse.de/tests/5251637/file/autoinst-log.txt>

```
[2021-01-06T02:02:06.500 CET] [debug] Blocking SIGTERM
[2021-01-06T02:02:06.548 CET] [debug] Unblocking SIGTERM
7670: channel_out 16, channel_in 15
[2021-01-06T02:02:06.677 CET] [debug] Blocking SIGTERM
[2021-01-06T02:02:06.718 CET] [debug] Unblocking SIGTERM
7780: cmdpipe 14, rsppipe 17
[2021-01-06T02:02:06.722 CET] [debug] started mgmt loop with pid 7780
[2021-01-06T02:02:06.784 CET] [debug] qemu version detected: 4.2.1
[2021-01-06T02:02:06.786 CET] [debug] running /usr/bin/chattr -f +C /var/lib/openqa/pool/7/raid
[2021-01-06T07:02:04.371 CET] [debug] isotovideo received signal TERM
[2021-01-06T07:02:04.371 CET] [debug] backend got TERM
[2021-01-06T07:02:04.373 CET] [info] ::: OpenQA::Qemu::Proc::save_state: Refusing to save an empty
state file to avoid overwriting a useful one
[2021-01-06T07:02:04.373 CET] [debug] Passing remaining frames to the video encoder
[2021-01-06T07:02:04.374 CET] [debug] Waiting for video encoder to finalize the video
[2021-01-06T07:02:04.374 CET] [debug] The built-in video encoder (pid 7859) terminated
[2021-01-06T07:02:04.375 CET] [debug] commands process exited: 0
[2021-01-06T07:02:04.376 CET] [debug] sending magic and exit
[2021-01-06T07:02:04.376 CET] [debug] received magic close
[2021-01-06T07:02:04.378 CET] [debug] [autotest] process exited: 1
[2021-01-06T07:02:04.383 CET] [debug] backend process exited: 0
failed to start VM at /usr/lib/os-autoinst/backend/driver.pm line 126.
[2021-01-06T07:02:04.483 CET] [debug] stopping autotest process 7701
[2021-01-06T07:02:04.483 CET] [debug] done with autotest process
7670: EXIT 1
```

Steps to reproduce

Find jobs referencing this ticket with the help of

<https://raw.githubusercontent.com/os-autoinst/scripts/master/openqa-query-for-job-label> ,
call openqa-query-for-job-label poo#81828

Problem

- **H1:** chattr itself is blocked on I/O
- **H2:** Some other process after chattr is blocked

Workaround

Retrigger. Also handled by [auto-review](#)

Related issues:

Related to openQA Project - action #96007: OpenQA jobs randomly time out duri...

Blocked

2021-07-26

Related to openQA Project - action #64917: auto_review:"(?s)qemu-img.*runcmd...	Resolved	2020-03-26
Related to openQA Infrastructure - action #98307: Many jobs in o3 fail with t...	Resolved	2021-09-08

History

#1 - 2021-01-06 09:57 - dzedro

- Project changed from openQA Infrastructure to openQA Project
- Category set to Concrete Bugs

#2 - 2021-01-11 14:23 - okurz

- Subject changed from Jobs incomplete with auto_review:"Refusing to save an empty state file to avoid overwriting a useful one":retry to Jobs incomplete with auto_review:"(?s)Refusing to save an empty state file to avoid overwriting a useful one.*Result: timeout":retry

What you mentioned as auto-review expression looks like an unimportant symptom to me. The more relevant lines are:

```
[2021-01-06T02:02:06.786 CET] [debug] running /usr/bin/chatrr -f +C /var/lib/openqa/pool/7/raid
[2021-01-06T07:02:04.371 CET] [debug] isotovideo received signal TERM
```

5 h (!) have passed between the two lines and the job ends with "timeout_exceeded". But we can include that in the regex

#3 - 2021-01-13 17:10 - mkittler

Yes, this job has clearly just been aborted because the timeout has been exceeded. The job's result is timeout_exceeded and the MAX_JOB_TIME is in accordance with the actual runtime.

I doubt we can find out at this point what blocked chatrr. Likely something IO related.

"Refusing to save an empty state file to avoid overwriting a useful one" is likely logged because the QEMU backend has already done first initializations at the time the test was aborted but the VM hasn't been started yet.

#4 - 2021-07-22 17:06 - AdamWill

I've been seeing this same thing periodically in Fedora openQA for the last while (I should really have noted which git snapshot update made it start happening, but I didn't, sigh). <https://openqa.fedoraproject.org/tests/933232> is the most recent example. There doesn't seem to be much rhyme or reason to when or why it happens, it just...sometimes happens. Exact same behaviour, nothing seems to be logged after the chatrr command until two hours later when the timeout is exceeded and the job gets shut down.

#5 - 2021-07-26 10:32 - okurz

- Related to action #96007: OpenQA jobs randomly time out during setup phase added

#6 - 2021-07-26 10:33 - okurz

- Priority changed from Normal to High
- Target version changed from future to Ready

also seen in [#96007](#) , adding to backlog

#7 - 2021-07-26 10:46 - okurz

- Assignee set to okurz

I have an idea though. We should be able to extend the openQA config to also run openqa-review for "timeout_exceeded" jobs to catch such issues and automatically retry as workaround at least

#8 - 2021-07-26 11:01 - okurz

- Status changed from New to In Progress

Manually added a line on osd in /etc/openqa/openqa.ini:

```
job_done_hook_timeout_exceeded = env host=openqa.suse.de /opt/os-autoinst-scripts/openqa-label-known-issues-ho
ok
```

Triggering a job with way too short timeout to trigger an artificial timeout exceeded result:

```
openqa-clone-job --within-instance https://openqa.suse.de/tests/6569084 _GROUP=0 BUILD=poo81828_okurz MAX_JOB_
TIME=1
```

-> <https://openqa.suse.de/tests/6570359> which triggered the hook script.

Created https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/531 to include that change in persistent salt config.

EDIT: With echo -e "https://openqa.suse.de/tests/6552459\nhttps://openqa.suse.de/tests/6570359" | env host=openqa.suse.de dry_run=1 ./openqa-label-known-issues I verified that <https://openqa.suse.de/tests/6552459> as mentioned in [#96007](#) is detected by auto-review but the simple job I triggered <https://openqa.suse.de/tests/6570359> is not as expected.

#9 - 2021-07-27 04:09 - openqa_review

- Due date set to 2021-08-10

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

#10 - 2021-07-27 14:49 - okurz

- Status changed from In Progress to Feedback

https://gitlab.suse.de/openqa/salt-states-openqa/-/merge_requests/531 merged. Waiting some days to crosscheck for jobs in "timeout_exceeded". Then I can check if there are any jobs automatically labeled with this ticket by calling openqa-query-for-job-label poo#81828, currently none.

#11 - 2021-07-30 10:36 - okurz

- Subject changed from Jobs incomplete with auto_review:"(?s)Refusing to save an empty state file to avoid overwriting a useful one.*Result: timeout":retry to Jobs run into timeout_exceeded after the 'chattr' call, no output until timeout, auto_review:"(?s)Refusing to save an empty state file to avoid overwriting a useful one.*Result: timeout":retry

auto-review for timeout_exceeded works, output from openqa-query-for-job-label poo#81828:

```
6602517|2021-07-29 22:15:20|done|timeout_exceeded|publiccloud_ltp_syscalls|timeout: test execution exceeded MAX_JOB_TIME|openqaworker5
6602131|2021-07-29 03:31:40|done|timeout_exceeded|create_hdd_minimal_base+sdk+python2|timeout: test execution exceeded MAX_JOB_TIME|openqaworker5
6598688|2021-07-28 21:15:50|done|timeout_exceeded|extra_tests_kdump_multipath|timeout: test execution exceeded MAX_JOB_TIME|openqaworker13
6597388|2021-07-28 20:15:25|done|timeout_exceeded|carwos-jsc-CAR-88_panic_on_softlockup|timeout: test execution exceeded MAX_JOB_TIME|openqaworker5
6593547|2021-07-28 13:32:52|done|timeout_exceeded|carwos-pcre|timeout: test execution exceeded MAX_JOB_TIME|openqaworker9
```

For example <https://openqa.suse.de/tests/6593547> is one of those cases. It was automatically retriggered as <https://openqa.suse.de/tests/6595487> as confirmed by <https://openqa.suse.de/admin/auditlog>. The cloned job turned out passed. So the workaround is in place.

Now for the actual issue. In the job <https://openqa.suse.de/tests/6593547> there is

```
[2021-07-28T13:33:17.491 CEST] [debug] running /usr/bin/chattr -f +C /var/lib/openqa/pool/3/raid
[2021-07-28T15:32:51.812 CEST] [debug] isotovideo received signal TERM
```

so same as in [#81828#note-2](#). <https://openqa.suse.de/tests/6595487/logfile?filename=autoinst-log.txt> shows what we should expect:

```
[2021-07-28T16:26:14.368 CEST] [debug] running /usr/bin/chattr -f +C /var/lib/openqa/pool/27/raid
[2021-07-28T16:26:14.394 CEST] [debug] running /usr/bin/qemu-img info --output=json /var/lib/openqa/pool/27/carwos-chuller_branch_1-509396.qcow2
[2021-07-28T16:26:14.422 CEST] [debug] running /usr/bin/qemu-img create -f qcow2 -b /var/lib/openqa/pool/27/carwos-chuller_branch_1-509396.qcow2
```

the chattr call comes from os-autoinst backend/qemu.pm

<https://github.com/os-autoinst/os-autoinst/blob/fe30f5910b9296219e2d0db02992be084a02c9cd/backend/qemu.pm#L755>. The "qemu-img create" (maybe also "qemu-img info") comes from OpenQA::Qemu::Proc::init_blockdev_images. In [#64917](#) I added retrying for the "qemu-img create" calls. But it does not look like we even reach this point. "qemu-img info" should come first. I should run in debugger or with tracing what code is run between "chattr" and "qemu-img info".

#12 - 2021-07-30 12:39 - okurz

- Related to action [#64917](#): auto_review:"(?s)qemu-img.*run.*failed with exit code 1" sometimes but no apparent error message added

#13 - 2021-07-30 12:48 - mkittler

I should run in debugger or with tracing what code is run between "chattr" and "qemu-img info".

Yes, it would be good to know what happened then. Possibly chattr also just never returned.

Btw, SIGTERM comes definitely from the worker (middle line is from worker log):

```
[2021-07-28T13:33:17.491 CEST] [debug] running /usr/bin/chattr -f +C /var/lib/openqa/pool/3/raid
[2021-07-28T15:32:51.0767 CEST] [debug] [pid:15084] Stopping job 6593547 from openqa.suse.de: 06593547-carwos-dev-qemu-x86_64-Build509396-chuller_branch_1-carwos-pcre@64bit - reason: timeout
[2021-07-28T15:32:51.812 CEST] [debug] isotovideo received signal TERM
```

#14 - 2021-07-30 13:03 - okurz

- Description updated

- Due date changed from 2021-08-10 to 2021-08-27

- Priority changed from High to Low

<https://github.com/os-autoinst/os-autoinst/pull/1738> to add debugging output to narrow down as well as ensure that chattr times out earlier in case it's actually the culprit. As the ticket already applies an automatic workaround we can decrease prio to "Low". After the mentioned PR is merged and deployed we should wait sometime and look for labeled jobs with openqa-query-for-job-label poo#81828 and see which debugging output we see to decide if we support H1 or H2

EDIT: As of 2021-08-02 the change was not yet deployed on osd due to deployment being blocked by <https://build.opensuse.org/request/show/909336> so not effective on osd yet, no related jobs recorded on o3.

EDIT: As of 2021-08-03 still no deployment triggered (no one retriggered the deployment pipeline after perl-DBD-SQLite fixed)

#15 - 2021-08-27 08:29 - okurz

coolo pointed out <https://openqa.suse.de/tests/6955023#live> while it is/was running. At 09:52 the job was still stuck with the last line in live log

```
[2021-08-27T09:45:53.251 CEST] [debug] running timeout 30 /usr/bin/chattr -f +C /var/lib/openqa/pool/32/raid
```

I could login into openqaworker5 and confirm this in the autoinst-log.txt but pgrep -f -l chattr returns no match. rpm -q --changelog os-autoinst | grep 'Ensure chattr' shows that <https://github.com/os-autoinst/os-autoinst/pull/1738> is included which we can also see because we run chattr with timeout but there is no message `bmwqemu::diag(Configuring storage controllers and block devices);`;

```
# lsof .
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
lsof	6790	root	cwd	DIR	9,127	4096	17104900	.
lsof	6791	root	cwd	DIR	9,127	4096	17104900	.
worker	28342	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
perl	28658	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
/usr/bin/	28787	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
/usr/bin/	28816	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
/usr/bin/	28896	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
videoenco	28975	_openqa-worker	cwd	DIR	9,127	4096	17104900	.
bash	38791	root	cwd	DIR	9,127	4096	17104900	.

and from ps

```
_openqa+ 28342 0.2 0.0 149008 65816 ? Ss 09:45 0:03 /usr/bin/perl /usr/share/openqa/script/worker --instance 32
_openqa+ 28658 0.2 0.0 261168 139652 ? S 09:45 0:03 \_ perl /usr/bin/isotovideo -d
_openqa+ 28787 0.0 0.0 179372 87040 ? S 09:45 0:00 \_ /usr/bin/isotovideo: comman
_openqa+ 28816 0.2 0.0 4636004 138040 ? S1 09:45 0:03 \_ /usr/bin/isotovideo: autote
_openqa+ 28896 0.0 0.0 4395376 152160 ? S1 09:45 0:01 \_ /usr/bin/isotovideo: backen
_openqa+ 28975 0.2 0.0 1775880 15916 ? SN1 09:45 0:03 \_ /usr/lib/os-autoinst/videoencoder /var/lib/openqa/pool/32/vid
_openqa+ 29017 0.0 0.0 0 0 ? Z 09:45 0:00 \_ [/usr/bin/isotov] <defunct>
```

the "defunct" process is a subprocess of `_openqa+ 28896 0.0 0.0 4395376 152160 ? S1 09:45 0:01 _ /usr/bin/isotovideo: backen`

I then terminated 28896 which stopped the complete test execution.

#16 - 2021-08-30 07:55 - okurz

- Due date changed from 2021-08-27 to 2021-09-03

After the previous due-date update now the mentioned os-autoinst change <https://github.com/os-autoinst/os-autoinst/pull/1738> was deployed and we have new information with [#81828#note-15](#) . Now I am looking for ideas from other colleagues so that we can continue.

1. Could it be that <https://github.com/okurz/os-autoinst/blob/7ab3c151a9b050dc1156f56737fed7da09c20e6e/backend/qemu.pm#L757> silently dies and does not stop the backend process?
2. Do we need to abort the backend in `start_qemu`, e.g. `timeout on <current_backend>::do_start_vm`, when it blocks for too long?

#17 - 2021-09-02 13:19 - okurz

- Due date changed from 2021-09-03 to 2021-09-17

Reminded in chat to please look at the ticket

#18 - 2021-09-02 16:42 - AdamWill

I don't have any ideas off the top of my head, but can confirm a case that happened in Fedora - <https://openqa.fedoraproject.org/tests/963613> - failed in the same place:

```
[2021-08-31T08:08:26.336 UTC] [debug] running timeout 30 /usr/bin/chatrr -f +C /var/lib/openqa/pool/21/raid
[2021-08-31T10:08:25.404 UTC] [debug] isotovideo received signal TERM
```

I didn't catch it live so can't confirm the other findings.

#19 - 2021-09-10 11:19 - okurz

- Related to action #98307: Many jobs in o3 fail with timeout_exceeded on openqaworker1 auto_review:"timeout: setup exceeded MAX_SETUP_TIME":retry size:M added

#20 - 2021-09-15 10:01 - okurz

- Due date deleted (2021-09-17)
- Status changed from Feedback to New
- Assignee deleted (okurz)
- Target version changed from Ready to future

@AdamWill thank you for pointing that out. Unfortunately still I don't have a better clue how to continue and no one else could provide a suggestion so far. So I will unassign, remove the ticket from the backlog and remove the due-date. Effectively this will stay open until anyone else has a useful idea and wants to continue.

#21 - 2021-11-18 10:37 - cfconrad

Hi,
I also see this problem from time to time. And I just had the change to live debug it <http://openqa.wicked.suse.de/tests/10035>.
What I saw was this:

The worker pid is 34787

```
> echo $(ssh agabus -- pstree -p 34787 | grep -Po '\d+') | xargs ssh agabus -- ps -lyp
S UID PID PPID C PRI NI RSS SZ WCHAN TTY TIME CMD
S 103 12827 34787 0 80 0 147932 69858 core_s ? 0:04 perl /usr/bin/isotovideo -d
S 103 18887 12827 0 80 0 93340 49130 - ? 0:01 /usr/bin/isotovideo: comman
S 103 20098 12827 0 80 0 141480 2120558 - ? 0:01 /usr/bin/isotovideo: autote
S 103 20286 12827 0 80 0 162728 2061143 hrttime ? 0:02 /usr/bin/isotovideo: backen
S 103 20451 20286 0 99 19 20692 1401185 - ? 0:00 /usr/lib/os-autoinst/videoencoder /var/lib/o
penqa/pool/26/video.ogv -n
Z 103 20525 20286 0 80 0 0 0 - ? 0:00 [/usr/bin/isotov] <defunct>
S 103 34787 1 0 80 0 242656 84018 - ? 4:18 /usr/bin/perl /usr/share/openqa/script/worke
r --instance 26
```

My guess is that we still hang somehow in that process() call and when I do a strace on 20286 which is the backend process I see this in a loop:

```
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0}, 0x7ffd076d05a0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=0}, NULL) = 0
kill(20525, 0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0}, 0x7ffd076d05a0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=0}, NULL) = 0
kill(20525, 0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0}, 0x7ffd076d05a0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=0}, NULL) = 0
kill(20525, 0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0}, 0x7ffd076d05a0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=0, tv_nsec=0}, NULL) = 0
kill(20525, 0) = 0
clock_nanosleep(CLOCK_REALTIME, 0, {tv_sec=1, tv_nsec=0})
```

Note that the 20525 is the [/usr/bin/isotov] <defunct> process.

If I run chatrr manual in that case, I face a not supported error.

```
agabus:/var/lib/openqa/pool/26 # /usr/bin/chatrr +C /var/lib/openqa/pool/26/raid
```

/usr/bin/chattr: Operation not supported while setting flags on /var/lib/opensha/pool/26/raid

#22 - 2021-11-19 11:49 - cfconrad

One try to solve this problem: <https://github.com/os-autoinst/os-autoinst/pull/1855>