openQA Tests - action #36027

coordination # 23650 (Resolved): [sle][functional][ipmi][epic][u] Fix test suite gnome to work on ipmi 12-SP3 and 15 (WAS: test fails in boot_from_pxe - connection refused trying to ipmi host over ssh?)

[sle][functional][u][ipmi] test fails in boot_from_pxe - pxe boot menu doesn't show up at all

2018-05-09 08:22 - zluo

Status: Resolved
Priority: High
Assignee: xlai
Category: Bugs in existing tests
Target version: Milestone 30
Difficulty:

Description
compare with successful test run, pxe boot menu is coming up at all.

Observation
openQA test in scenario sle-15-Installer-DVD-x86_64-btrfs@64bit-ipmi fails in boot_from_pxe

Investigation

Hypotheses

- H1.1 The problem is the IPMI machine.  SUPPORTED BY E1.2-1
- H1.2 The problem only happens on that machine.  SUPPORTED BY E1.2-1
- H2.1 The problem is not the worker.  SUPPORTED BY E1.2-1

Experiments

- E1.2-1 Find the same issue on another IPMI machine.
- R1.2-1 Not found.
- E2.1-1 Find the same issue on another worker.
- R2.1-1 Not found.

Reproducible

Fails since (at least) Build 609.1 (current job)

Expected result

Last good: 600.1 (or more recent)

Further details

- Latest for osd#IPMI_SLE15_BTRFS
- Latest for osd#IPMI_SLE15-SP1_BTRFS
- Latest for osd#IPMI_SLE12-SP4-BTRFS

Related issues:
Related to openQA Project - action #44978: [ipmi unstability] jobs got blue s... Rejected 2018-12-11
Related to openQA Tests - action #41909: [sle][remote-backends]test fails in ... New 2018-10-02
Related to openQA Tests - action #42383: [sle][functional][u][sporadic][ipmi]... Rejected 2018-10-12
Related to openQA Tests - action #31375: [sle][functional][ipmi][u][hard] tes... Rejected 2018-02-05
Related to openQA Infrastructure - action #41480: [sle][functional][u][ipmi] ... Resolved 2018-09-24
Related to openQA Tests - action #38888: [functional][sle][u][sporadic][ipmi]... Rejected 2018-07-26
Related to openQA Tests - action #41693: [sle][functional][u][sporadic]... Rejected 2018-09-27
Related to openQA Tests - action #57587: [virtualization][u] test fails in re... Rejected 2019-10-01
#1 - 2018-05-17 21:32 - okurz
- Subject changed from [sle][functional][u] test fails in boot_from_pxe - pxe boot menu doesn't show up at all to [sle][functional][u][ipmi] test fails in boot_from_pxe - pxe boot menu doesn't show up at all
- Target version set to Milestone 19

#2 - 2018-06-10 04:41 - okurz
- Has duplicate action #36955: [sle][functional][y][sporadic] test fails in boot_from_pxe - without network added

#3 - 2018-06-10 04:43 - okurz
- Due date set to 2018-07-31
- Priority changed from Normal to High
- Target version changed from Milestone 19 to Milestone 17

#4 - 2018-06-10 04:43 - okurz
- Related to coordination #23650: [sle][functional][ipmi][epic][u] Fix test suite gnome to work on ipmi 12-SP3 and 15 (WAS: test fails in boot_from_pxe - connection refused trying to ipmi host over ssh?) added

#5 - 2018-06-10 04:43 - okurz
- Related to action #31375: [sle][functional][ipmi][u][hard] test fails in first_boot - VNC installation on SLE 15 failed because of various issues (ipmi worker, first_boot, boot_from_pxe, await_install) added

#6 - 2018-06-15 18:53 - okurz
- Target version changed from Milestone 17 to Milestone 17

#7 - 2018-06-24 17:01 - okurz
Latest failure: https://openqa.suse.de/tests/1777178#step/18

#8 - 2018-06-17 13:04 - okurz
- Blocked by action #38411: [functional][y][sporadic][ipmi] test fails in boot_from_pxe - installer does not show up added

#9 - 2018-07-17 13:05 - okurz
- Due date deleted (2018-07-31)
- Status changed from New to Blocked
- Assignee set to okurz
- Priority changed from High to Low
- Target version changed from Milestone 17 to Milestone 21+

seems this ticket is specific for SLE15, let's look into #38411 for SLE12SP4 first

#10 - 2018-10-12 08:30 - SLindoMansilla
- Has duplicate action #37387: [sle][functional][ipmi][u] Fix test suite gnome to work on ipmi SLE 12 and 15 added

#11 - 2018-10-12 09:16 - SLindoMansilla
- Has duplicate deleted (action #37387: [sle][functional][ipmi][u] Fix test suite gnome to work on ipmi SLE 12 and 15)

#12 - 2018-10-12 09:16 - SLindoMansilla
- Status changed from Blocked to Workable

No more blocked. Ticket resolved. Should I assigned this to me and work on it?
SLindoMansilla wrote:

No more blocked. Ticket resolved. Should I assigned this to me and work on it?

Yes

SLindoMansilla wrote:

due to changes in a related task

SLindoMansilla wrote:

Status changed from Workable to In Progress
Not able to reproduce with 10 runs: http://slindomansilla-vm.qa.suse.de/tests/overview?build=fozzie_poo41480

I assume that here there was a problem that aborted the boot from network/PXE and defaulted to boot from the next device on the list, the disk. (Not possible to verify because there is no video where I could see any error of the network/PXE boot: https://openqa.suse.de/tests/1676634#step/boot_from_pxe/4

This duplicated report also support my assumption: #36955

At the moment, the most common sporadic issue on PXE boot is when the installer takes 30 minutes to start (it looks like it gets freeze), like:

- SLE15-SP1_default
- SLE12-SP4_default
- SLE12-SP4_btrfs

This sporadic issue appears on more than one IPMI machine and more than one worker.

All of them get freeze showing the following message after the ram check (https://openqa.suse.de/tests/2114173#step/boot_from_pxe/26):

```
[...]
RAM size: xxx MB
starting setctsid `showconsole` inst_setup yast
```

On the jobs working properly, the message is like (https://openqa.suse.de/tests/2118389#step/boot_from_pxe/8):

```
[...]
RAM size: xxx MB
BAD PASSWORD: it is based on a dictionary work
starting setctsid `showconsole` inst_setup yast
```

#29 - 2018-10-15 07:53 - SLindoMansilla
- Blocks action #41480: [sle][functional][u][ipmi] Malfunction of openqaworker2:25 - Investigate, bring it back or repair it (WAS: remove openqaworker2:25 (IPMI machine) from OSD testing) added

#30 - 2018-10-15 08:06 - SLindoMansilla
10 job run didn't reproduce the issue. Let's go big:

- 100 runs on local: http://slindomansilla-vm.qa.suse.de/tests/overview?build=impi_pxeboot_poo36027
- 100 on OSD: https://openqa.suse.de/tests/overview?build=impi_pxeboot_poo36027

#31 - 2018-10-15 14:17 - SLindoMansilla
- Status changed from In Progress to Feedback

Waiting for statistical investigation to finish

#32 - 2018-10-16 11:33 - okurz
- Has duplicate action #37387: [sle][functional][ipmi][u] Fix test suite gnome to work on ipmi SLE 12 and 15 added

#33 - 2018-11-16 13:35 - okurz
100 runs on local: http://slindomansilla-vm.qa.suse.de/tests/overview?build=impi_pxeboot_poo36027

Over a month has passed, what's your state on it?

#34 - 2018-11-16 13:45 - SLindoMansilla
- Status changed from Feedback to Workable

I don't remember anymore.
The verification runs still show a lot of failures on PXE boot.

#36 - 2018-11-19 06:06 - xlai

okurz wrote:

xlai https://progress.opensuse.org/issues/36027 is what I think the "main ticket" about making the booting of ipmi jobs more stable. I recommend you provide your observations, e.g. links to failing jobs, there. Also there are open PRs which I guess need some more work, e.g. https://github.com/os-autoinst/os-autoinst/pull/1041 and https://github.com/os-autoinst/os-autoinst/pull/1047. Also there is https://github.com/os-autoinst/os-autoinst/pull/1021 which was deployed just two days ago. I recommend to collect statistics if this helped or not.

okurz, Thanks for your reply.

Yes, boot_from_pxe is a typical unstable scenario for ipmi jobs, and it is keep happening now(after deployment of https://github.com/os-autoinst/os-autoinst/pull/1021). There are several variations of such failure. Take latest sle15sp1 build 96.6 for example(also often happens in other builds),

- one is no ssh_server_started needle show up, like https://openqa.suse.de/tests/2262871#step/boot_from_pxe/22
- another one of no ssh_server_started needle show up, https://openqa.suse.de/tests/2262655#step/boot_from_pxe/14
- one is blue screen which makes needle matching of openqa does not work(addign a new needle with blue screen will not help), like https://openqa.suse.de/tests/2262449#step/boot_from_pxe/26

According to our experience, when an ipmi machine has been taking jobs for a long time like a day or several days, the ipmi sol will become kind of unstable -- any kind of unstability can be happening, like connection errors, slow responding, wrong response or no response etc. When this happens, resetting the main ipmi board via 'mc reset' can help. It can recover the machine back to stable state again. That's why we created the jenkins jobs http://jenkins.qa.suse.de/job/restart-ipmi-mainboard/.

Currently that jenkins job to do 'mc reset' is disabled(over a month ago). And recent builds results show more and more boot_from_pxe failures. Yes, it has flaw that it will disturb jobs that are running when the reset job is kicked -- it is actually a known issue when it is added, and that's why do it during mid-night. So, here, via this ticket, we are aiming to find a better solution for the unstable sol issue. IMHO, the sol unstability can not be 100% eliminated, but we should find ways to decrease it to a much lower acceptable ratio. Of course the ways can be case by case according to exact failures. But below that, there misses a fundamental one -- we should find a better substitute solution for that jenkins jobs. And I think it is needed for any new product from beta phase.

Possible solutions discussed between calen and me are:

- Jenkins solution: Trigger "mc reset" on ipmi SUT before any openqa new build all jobs are triggered.
- OpenQA solution: add job to do the mc reset, and set all ipmi jobs START_AFTER that test(can it be added on webui global setting somewhere?).

Another solution is the one you mentioned -- wayne's PR#1047 which do "mc reset " in console's activate function. It may have big impact on the ipmi tests' stability. But it is hard to say how positive it can be -- need to prove after deployment when PR ready. Personally I am sure not doing 'mc reset' harm the stability, but not sure whether doing it every time can help a lot.

Do you agree with my points? Which solution do you prefer? Everyone is welcome to give opinions here :)

#37 - 2018-11-19 08:09 - SLindoMansilla

xlai wrote:

Possible solutions discussed between calen and me are:

- Jenkins solution: Trigger "mc reset" on ipmi SUT before any openqa new build all jobs are triggered.
- OpenQA solution: add job to do the mc reset, and set all ipmi jobs START_AFTER that test(can it be added on webui global setting somewhere?).

Hi xlai,

I think that both solution should be fine. I tend to think that letting Jenkins do it could be better, since this problem is an "infrastructure" problem, and not a test problem. And also, I think it is also less work and less maintenance if it is done through Jenkins.

Another solution is the one you mentioned -- wayne's PR#1047 which do "mc reset " in console's activate function. It may have big impact on the ipmi tests' stability. But it is hard to say how positive it can be -- need to prove after deployment when PR ready. Personally I am sure not doing 'mc reset' harm the stability, but not sure whether doing it every time can help a lot.

I think that would be a drastic change. The first two options you provided are better.

Kind Regards

#38 - 2018-11-19 08:22 - xlai

SLindoMansilla wrote:
xlaí wrote:

Possible solutions discussed between calen and me are:

- Jenkins solution: Trigger "mc reset" on ipmi SUT before any openqa new build all jobs are triggered.
- OpenQA solution: add job to do the mc reset, and set all ipmi jobs START_AFTER that test(can it be added on webui global setting somewhere?).

Hi xlaí,

I think that both solution should be fine. I tend to think that letting Jenkins do it could be better, since this problem is an "infrastructure" problem, and not a test problem. And also, I think it is also less work and less maintenance if it is done through Jenkins.

Another solution is the one you mentioned -- wayne's PR#1047 which do "mc reset " in console's activate function. It may have big impact on the ipmi tests' stability. But it is hard to say how positive it can be -- need to prove after deployment when PR ready. Personally I am sure not doing 'mc reset' harm the stability, but not sure whether doing it every time can help a lot.

I think that would be a drastic change. The first two options you provided are better.

Kind Regards

@SLindoMansilla, Thank you for the feedback, very useful for us!

Welcome others' ideas!

#39 - 2018-11-19 21:40 - okurz

SLindoMansilla wrote:

I think that both solution should be fine. I tend to think that letting Jenkins do it could be better, since this problem is an "infrastructure" problem, and not a test problem. And also, I think it is also less work and less maintenance if it is done through Jenkins.

If we can ensure with a jenkins job that the trigger is synchronized to test execution and not blindly, time-based, then we can try that. Goes in the direction of replacing the cron job for rsync.pl with event-triggers, potentially also by jenkins.

#40 - 2018-11-22 07:45 - xlai

okurz wrote:

SLindoMansilla wrote:

I think that both solution should be fine. I tend to think that letting Jenkins do it could be better, since this problem is an "infrastructure" problem, and not a test problem. And also, I think it is also less work and less maintenance if it is done through Jenkins.

If we can ensure with a jenkins job that the trigger is synchronized to test execution and not blindly, time-based, then we can try that. Goes in the direction of replacing the cron job for rsync.pl with event-triggers, potentially also by jenkins.

Thanks for sharing your opinion, oli!

Yes, the point of jenkins way is to not disturb test execution. Actually when we proposed this jenkins way, we thought that the openqa tests were triggered via jenkins. Obviously we were wrong ;(, they are by cron jobs.

So we currently have two choices, if we prefer this way:

- if replacing cron job with jenkins via event triggers needs long time .eg not until late 15sp1. we may need to ask the maintainer for the cron job of rsync.pl to insert similar HW reset operations before starting the jobs
- else wait until cron job replacement is done, then use jenkins jobs to trigger HW reset before trigger openqa tests with the same event like new build in repo detected

How do you choose, okurz and @SLindoMansilla? BTW do you know who maintains the cron job? We'd better also involve him.

#41 - 2018-11-22 12:53 - okurz

xlai wrote:

okurz wrote:

SLindoMansilla wrote:
I think that both solution should be fine. I tend to think that letting Jenkins do it could be better, since this problem is an "infrastructure" problem, and not a test problem. And also, I think it is also less work and less maintenance if it is done through Jenkins.

If we can ensure with a jenkins job that the trigger is synchronized to test execution and not blindly, time-based, then we can try that. Goes in the direction of replacing the cron job for rsync.pl with event-triggers, potentially also by jenkins.

Thanks for sharing your opinion, oli!

Yes, the point of Jenkins way is to not disturb test execution. Actually when we proposed this Jenkins way, we thought that the openqa tests were triggered via Jenkins. Obviously we were wrong ;), they are by cron jobs.

So we currently have two choices, if we prefer this way:

- if replacing cron job with Jenkins via event triggers needs long time , eg not until late 15sp1, we may need to ask the maintainer for the cron job of rsync.pl to insert similar HW reset operations before starting the jobs
- else wait until cron job replacement is done, then use Jenkins jobs to trigger HW reset before trigger openqa tests with the same event like new build in repo detected

Why not handle all that in the according jobs that are triggered on openQA just like https://github.com/os-autoinst/os-autoinst/pull/1021 ? Also, do we actually still have the problem you meant or did the mentioned PR fix it maybe?

How do you choose, okurz and @SLindoMansilla?

BTW do you know who maintains the cron job? We'd better also involve him.

Please keep in mind that it is common responsibility how tests are triggered, including the cron jobs. The cron jobs are maintained in https://gitlab.suse.de/openqa/salt-states-openqa/blob/master/etc/master/cron.d/SLES.CRON so everyone may open a MR with proposals.

#42 - 2018-11-23 05:42 - cachen
okurz wrote:

- if replacing cron job with Jenkins via event triggers needs long time , eg not until late 15sp1, we may need to ask the maintainer for the cron job of rsync.pl to insert similar HW reset operations before starting the jobs
- else wait until cron job replacement is done, then use Jenkins jobs to trigger HW reset before trigger openqa tests with the same event like new build in repo detected

Why not handle all that in the according jobs that are triggered on openQA just like https://github.com/os-autoinst/os-autoinst/pull/1021 ? Also, do we actually still have the problem you meant or did the mentioned PR fix it maybe?

Alice is taking leave today!
My understand is she has answered in above comment #36. And PR#1021 is just for ipmitool to disconnecting sol, which was a function missing in ipmi-backend, it won't fix unstable issue. Let me give some background, Alice is talking about to reset ipmi device, which is the known issue in supermicro ipmi firmware, that will cause its SOL output to console very unstable(such as blue screen, no respond of typing), the only workaround coolo found is to reset ipmi device to clear cache, and it actually helped as Alice has introduced about the job in Jenkins.

How do you choose, okurz and @SLindoMansilla?

BTW do you know who maintains the cron job? We'd better also involve him.

Please keep in mind that it is common responsibility how tests are triggered, including the cron jobs. The cron jobs are maintained in https://gitlab.suse.de/openqa/salt-states-openqa/blob/master/etc/master/cron.d/SLES.CRON so everyone may open a MR with proposals.

We know all can contribute code, but it need a person or should have a person responsible for this component, simply say who can decide for the code merge, I think this is what Alice's mean about 'who maintains'.

This ipmi unstable issue is widely impact and will getting worse and worse along with more ipmi baremetal testcases been added, to driving this forward, we better to ask for more idea/feedback and get agreement for the solution as first step.

#43 - 2018-11-24 07:27 - okurz
- Priority changed from Normal to High

cachen wrote:

2021-05-21
We know all can contribute code, but it need a person or should have a person responsible for this component, simply say who can decide for the code merge.

I see, that list is (or should be?) visible in each gitlab repo: https://gitlab.suse.de/openqa/salt-states-openqa/project_members

This ipmi unstable issue is widely impact and will getting worse and worse along with more ipmi baremetal testcases been added, to driving this forward, we better to ask for more idea/feedback and get agreement for the solution as first step.

I agree. This is why this ticket already "blocks" some others and I suggest to join forces or at least not duplicate the effort by working at one thing at a time. And I think SLindoMansilla is doing a very good job at this. He will not be available in the next days though.

I recommend to try the following:

- Gather statistics again as in #36027#note-30 but schedule only the test modules "boot_from_pxe" and "welcome". I don't see the need to test any further test modules. Then find the error rate from that. I would really like to see something statistically meaningful, e.g. for 100 jobs which resemble 10 sets of sample size 10 we can get mean and std
- Add a test module dynamically to call mc reset in a test module. This is what I meant with reference to gh#os-autoinst/openQA#1021. See gh#os-autoinst/openQA#1855 for documentation for dynamic test module override

#44 - 2018-11-27 03:34 - xlai
okurz wrote:

cachen wrote:

We know all can contribute code, but it need a person or should have a person responsible for this component, simply say who can decide for the code merge.

I see, that list is (or should be?) visible in each gitlab repo: https://gitlab.suse.de/openqa/salt-states-openqa/project_members

This ipmi unstable issue is widely impact and will getting worse and worse along with more ipmi baremetal testcases been added, to driving this forward, we better to ask for more idea/feedback and get agreement for the solution as first step.

I agree. This is why this ticket already "blocks" some others and I suggest to join forces or at least not duplicate the effort by working at one thing at a time. And I think SLindoMansilla is doing a very good job at this. He will not be available in the next days though.

I recommend to try the following:

- Gather statistics again as in #36027#note-30 but schedule only the test modules "boot_from_pxe" and "welcome". I don't see the need to test any further test modules. Then find the error rate from that. I would really like to see something statistically meaningful, e.g. for 100 jobs which resemble 10 sets of sample size 10 we can get mean and std
- Add a test module dynamically to call mc reset in a test module. This is what I meant with reference to gh#os-autoinst/openQA#1021. See gh#os-autoinst/openQA#1855 for documentation for dynamic test module override

IMHO, for this unstable issue, we do not need a really accurate statistic for the happening ratio. The ratio changes from build to build according to different situations. It should be more like a practical problem, so a general number to help understand the situation is enough.

According to our watching results, nearly 5% to 30% percent(yes, big variation).

And finding a practical and easy way to keep it low rather than 0 (we can never achieve it) should be what we look for.

- Add a test module dynamically to call mc reset in a test module. This is what I meant with reference to gh#os-autoinst/openQA#1021. See gh#os-autoinst/openQA#1855 for documentation for dynamic test module override

Doing 'mc reset' in test code is just what we proposed as the 'OpenQA solution' in comment #36 :). We propose it to be a standalone job rather than dynamically loaded test module in a big testsuites because the latter one will make things more complex (mc reset will fully break ipmi functions/connections for several minutes). I will not explain too much details here because it will be too long. If you like, I can explain offline :).

SLindoMansilla gave in comment#37 which you also agreed in comment#39 that this problem is an "infrastructure" problem, and not a test problem, so handling the issue in test code is not as well as handling it in jenkins/cron job.

Is this still your current opinion or you support more the 'OpenQA solution'?

Let's first clearly give the ideas and then find an agreed way and then take action :). Really appreciate this cooperation with you guys!

#45 - 2018-11-28 10:22 - okurz
- Target version changed from Milestone 21+ to Milestone 21

xlai wrote:

2021-05-21
I recommend to try the following:

- Gather statistics again as in #36027#note-30 but schedule only the test modules "boot_from_pxe" and "welcome". I don't see the need to test any further test modules. Then find the error rate from that. I would really like to see something statistically meaningful, e.g. for 100 jobs which resemble 10 sets of sample size 10 we can get mean and std.

IMHO, for this unstable issue, we do not need a really accurate statistic for the happening ratio. The ratio changes from build to build according to different situations. It should be more like a practical problem, so a general number to help understand the situation is enough.

According to our watching results, nearly 5% to 30% percent (yes, big variation).

Your assessment of 5-30% is good enough for me. That means that we should run 100 jobs at least to ensure any fix works fine.

And finding a practical and easy way to keep it low rather than 0 (we can never achieve it) should be what we look for.

- Add a test module dynamically to call mc reset in a test module. This is what I meant with reference to gh#os-autoinst/openQA#1021. See gh#os-autoinst/openQA#1855 for documentation for dynamic test module override.

Doing 'mc reset' in test code is just what we proposed as the 'OpenQA solution' in comment #36 :). We propose it to be a standalone job rather than dynamically loaded test module in a big testsuites because the latter one will make things more complex (mc reset will fully break ipmi functions/connections for several minutes). I will not explain too much details here because it will be too long. If you like, I can explain offline :).

Explain "offline"? I would like to visit you in person but I doubt we have the budget for now ;)

When we do it in a job, how do we ensure that a successor job would run on the same machine? If you say that "mc reset" will disrupt functionality for a longer time then I have a different idea: How about a specific test module which we run before all other modules in all IPMI tests and in this module check if the management console is usable, if it is not, then call 'mc reset' and wait until it is usable again.

SLindoMansilla gave in comment#37 which you also agreed in comment#39 that this problem is an "infrastructure" problem, and not a test problem, so handling the issue in test code is not as well as handling it in jenkins/cron job.

Is this still your current opinion or you support more the 'OpenQA solution'?

I don't see how we could easily sync the jenkins/cron execution to the openQA jobs to prevent any openQA jobs to be disrupted. So I guess unless we can find an answer to that I favor the "openQA test module" solution.

#46 - 2018-11-30 09:05 - xlai

okurz wrote:

xlai wrote:

I recommend to try the following:

- Gather statistics again as in #36027#note-30 but schedule only the test modules "boot_from_pxe" and "welcome". I don't see the need to test any further test modules. Then find the error rate from that. I would really like to see something statistically meaningful, e.g. for 100 jobs which resemble 10 sets of sample size 10 we can get mean and std.

IMHO, for this unstable issue, we do not need a really accurate statistic for the happening ratio. The ratio changes from build to build according to different situations. It should be more like a practical problem, so a general number to help understand the situation is enough.

According to our watching results, nearly 5% to 30% percent (yes, big variation).

Your assessment of 5-30% is good enough for me. That means that we should run 100 jobs at least to ensure any fix works fine.

And finding a practical and easy way to keep it low rather than 0 (we can never achieve it) should be what we look for.

- Add a test module dynamically to call mc reset in a test module. This is what I meant with reference to gh#os-autoinst/openQA#1021. See gh#os-autoinst/openQA#1855 for documentation for dynamic test module override.

Doing 'mc reset' in test code is just what we proposed as the 'OpenQA solution' in comment #36 :). We propose it to be a standalone job rather than dynamically loaded test module in a big testsuites because the latter one will make things more complex (mc reset will fully break ipmi functions/connections for several minutes). I will not explain too much details here because it will be too long. If you like, I can explain offline :).

Explain "offline"? I would like to visit you in person but I doubt we have the budget for now ;)

2021-05-21 9/19
Thank you for sharing your ideas and the efforts you spend on this ticket. I really appreciate!

I actually mean we may need to find a timeslot to talk via jangout to drive a quicker, deeper and more efficient discussion :).

It seems currently we both still lacks enough understanding on each other area that affects solution selection -- I need to know more about the cron job that triggers openqa jobs of a new build to let me understand why you say it is not easy to sync cron execution, while you need to know more about the consoles at different test stage of ipmi jobs and the limitation on ipmi sol itself as well as ipmi backend of openqa to support your suggested test module way(so the background for why I say nearly not feasible actually).

But I still want to invite you to an exchange program to visit beijing office ;-)  

When we do it in a job, how do we ensure that a successor job would run on the same machine?

Make all ipmi jobs to START_AFTER this single job that do 'mc reset' to all ipmi machines(simultaneously trigger reset command and wait them all up) within each worker class group.

If you say that "mc reset" will disrupt functionality for a longer time then I have a different idea: How about a specific test module which we run before all other modules in all IPMI tests and in this module check if the management console is usable, if it is not, then call 'mc reset' and wait until it is usable again.

SLindoMansilla gave in comment#37 which you also agreed in comment#39 that this problem is an "infrastructure" problem, and not a test problem, so handling the issue in test code is not as well as handling it in jenkins/cron job.

Is this still your current opinion or you support more the 'OpenQA solution'?

I don't see how we could easily sync the jenkins/cron execution to the openQA jobs to prevent any openQA jobs to be disrupted.

Sorry, I still do not understand why not easy(need more background from you) to sync cron jobs. Based on my assumption, the periodically triggered cron jobs will run basically like this: detect new build -> sync iso/repo/code -> trigger jobs(why can not pin 'mc reset' scripts(several minutes to finish) before triggering all openqa jobs?)

So I guess unless we can find an answer to that I favor the "openQA test module" solution.

The "openQA test module" may not work. Reasons are:

- the unstability of ipmi is random, so it is hard to stably detect whether ipmi sol is unstable or not. What we see often at the beginning are 3 cases: pxe menu does not show up/random typing error when typing pxe boot cmdline/blue screen after successful typing pxe boot cmdline and makes openqa needle matching does not work,
- which console we use to run the detection process? Sol console itself is possibly unstable already and ssh console is not available at the very start of our ipmi jobs( start from power on machine by ipmi backend, then takes the sol console to boot_from_pxe). Another way is to use other qemu/svirt worker to detect first, but this will not be able to combine with ipmi backend jobs any more. So not "openQA test module" way any more.

#47 - 2018-12-03 11:45 - xlai

@Oliver, Now I know why you say "I don't see how we could easily sync the jenkins/cron execution to the openQA jobs to prevent any openQA jobs to be disrupted.". In our original proposal in comment#36, actually both of the two ways "OpenQA job solution" and "Jenkins/Cron job", may disturb the already triggered running jobs in EARLIER builds. We mostly considered how to not disturb the coming NEW build's jobs, but kind of neglected the OLD builds. To avoid it, it really is not easy. Not only code will be complex, but also long time(hours in bad case), may be taken before triggering all the new build jobs which is not tolerable in the new build jobs triggering process. Big thanks to yifan -- I will not recognize it if not discussing it with yifan locally.

Regarding your proposed "OpenQA test module" solution, I kind of figure out a workable way based on it despite the reasons in comment#46. Please help to evaluate.

Overall it is a compromised solution between code complexity and possible practical effect. This solution aims to use simple code to achieve acceptable improvement. Main points are:

- We will implement the 'mc reset' in post_fail_hook of every ipmi test.
- This reset will be done on ssh based console for two reasons: 1) among the ipmi jobs failures due to unstable ipmi, most times the ssh connection is available ; 2) do 'mc reset' will only interrupt sol connection, ssh not impacted
- this 'mc reset' will only be done against this test SUT, rather than all ipmi SUT in original proposals
- no code logic to detect whether this job failure is due to unstable sol or not. reason is as stated in comment#46 that the unstability of ipmi is random, so it is hard to STABLY DETECT whether ipmi sol is unstable or not. But we will add code to detect stable failures like whether it fails due to product bug/automation issue, via either dynamic/static setting or some other way(suggestions are welcome).

Main advantage of this solution:

- reset will not interrupt any openqa jobs, so synced
- reset is done when needed, rather than time based/build based, more practical
• easy code for general ipmi solution
• easy full solution, only test involved, no backend change/build trigger process change

Main disadvantage of it:
• can not fully ensure 'mc reset' is done: when job fails at stage that ssh is not available(from the history failure jobs, the followed fail job will hit the reset code)
• marking tests fail by product bug or automation issues will happen at least after review, so have time delay for new issues -- so in worst case, in one round of testing, product bug or automation issues may introduce some extra 'mc reset' than needed, and each reset may takes around 3 minutes(in my view, it is acceptable for virtualization job which takes over 2 hours generally for each job)

Classical failure recovery scenario:
ipmi job fail by unstable ipmi -> post_fail_hook detects it is not product bug/automation issue, and trigger 'mc reset' of sol on ssh console -> ipmi board recovered and this fail job exits -> new job starts with recovered machine -> stably run for hours/days -> next ipmi unstable issue happen again and fail job -> repeat ...

So how do you think of this solution? Welcome any reply :)!

#48 - 2018-12-03 13:02 - okurz
xlai wrote:
so in worst case, in one round of testing, product bug or automation issues may introduce some extra 'mc reset' than needed, and each reset may takes around 3 minutes(in my view, it is acceptable for virtualization job which takes over 2 hours generally for each job)

why not always call "mc reset" before every test? That should ensure the MC to be stable for the whole job run and would also invest just 3m vs. 2h?

the unstability of ipmi is random, so it is hard to stably detect whether ipmi sol is unstable or not. What we see often at the beginning are 3 cases: pxe menu does not show up/random typing error when typing pxe boot cmdline/blue screen after successful typing pxe boot cmdline and makes openqa needle matching does not work.

to detect an unstable MC can we try to simply type something into the connection and see that characters to be shown? How would a human detect if the "MC is stable" reliably?

Classical failure recovery scenario:
ipmi job fail by unstable ipmi -> post_fail_hook detects it is not product bug/automation issue, and trigger 'mc reset' of sol on ssh console -> ipmi board recovered and this fail job exits -> new job starts with recovered machine -> stably run for hours/days -> next ipmi unstable issue happen again and fail job -> repeat ...

I see one flaw in this: The job would fail in a seemingly random state so label carry over is unlikely to work correctly and therefore more manual review effort is required

#49 - 2018-12-04 03:41 - xlai
okurz wrote:
Thanks for the comments.

xlai wrote:
so in worst case, in one round of testing, product bug or automation issues may introduce some extra 'mc reset' than needed, and each reset may takes around 3 minutes(in my view, it is acceptable for virtualization job which takes over 2 hours generally for each job)

why not always call "mc reset" before every test? That should ensure the MC to be stable for the whole job run and would also invest just 3m vs. 2h?

As explained in earlier comment#46, do reset before every test is not feasible technically based on current openqa tool implementation. But do it at the end of every test is possible.

The most important reason for not do the reset every time is:
Doing 'mc reset' with such higher frequency is an unproved situation -- not sure whether it can really help. Comment#36 to wayne's PR#1047 is also valid explanation to this, since they are similar( do "mc reset " in console's activate function is also frequent reset):

It may have big impact on the ipmi tests' stability. But it is hard to say how positive it can be -- need to prove after deployment when PR ready. Personally I am sure not doing 'mc reset' harm the stability, but not sure whether doing it every time can help a lot.

IPMI machines’ stability changes from one to one. Local proof on one machine can not give conclusion. It needs to be watched statistically when deployed in osd on the real test SUTs under the real work load.

2021-05-21
I will implement the solution as 'do mc reset' at the end of every test. Then we monitor the statistics on osd. If it improves to what we want, then we keep it. If not, bad case is that it lowers the stability for so frequent reset, then I will go back to the proposed way in comment#47.

the unstability of ipmi is random, so it is hard to stably detect whether ipmi sol is unstable or not. What we see often at the beginning are 3 cases: pxe menu does not show up/random typing error when typing pxe boot cmdline/blue screen after successful typing pxe boot cmdline and makes openqa needle matching does not work,

to detect an unstable MC can we try to simply type something into the connection and see that characters to be shown? How would a human detect if the "MC is stable" reliably?

Well, it is not enough, and case by case. I want to correct one misunderstanding that when ipmi machine becomes unstable then every operation on it is unstable. No, it is not that way. When typing is unstable, maybe screen is stable. When screen misses key parts, maybe typing is stable. So we can not tell one behavior from others.

Classical failure recovery scenario:

ipmi job fail by unstable ipmi -> post_fail_hook detects it is not product bug/automation issue, and trigger 'mc reset' of sol on ssh console ->
ipmi board recovered and this fail job exits -> new job starts with recovered machine -> stably run for hours/days -> next ipmi unstable issue happen again and fail job -> repeat ...

I see one flaw in this: The job would fail in a seemingly random state so label carry over is unlikely to work correctly and therefore more manual review effort is required

Yes, it is possible. And as I stated earlier, the proposal is not a PERFECT one, and not intend to be that way. After analyzing the five or six proposals that you and us and yifan gives(besides those passed by abandoned solutions), we find that this issue can only expect a compromise solution among code complexity/time/involved components(backend/test/trigger process)/stability improvement percentage.

However I am glad that how we review our group now already can provide enough information to indicate automation issue/product bug. So no extra review.

Thank you again for sharing the comment frankly. We are going closer to an agreed solution.

#50 - 2018-12-04 05:04 - okurz
xlai wrote:

I will implement the solution as 'do mc reset' at the end of every test.

But then we need to ensure the MC to be ready for the next test to start. I guess that again should be done with a test module in the beginning of each test.

#51 - 2018-12-04 07:06 - xlai
okurz wrote:

xlai wrote:

I will implement the solution as 'do mc reset' at the end of every test.

But then we need to ensure the MC to be ready for the next test to start. I guess that again should be done with a test module in the beginning of each test.

Ensuring MC ready at the end of a test is most likely to actually ensure it ready for the next test to start, isn't it? :) Of course there are rare exceptions, but quite few.

This 'after' way needs to add a module at the beginning which do mc reset on ssh console, however this requires the machine to have a workable ssh connection. But the machines are assumed to start from baremetal(we do have tests like host upgrade that break the os in the end, and os can not be up any more except reinstallation). So in such situations, with your suggested way, the machines will never have chance to do real test again until manual reinstallation which should be definitely avoided.

I will think about if any possible workarounds for it with acceptable sacrifice. This way introduces more extra time in machine preparation -- 1 reboot + mc reset, ~10 minutes for every time ;(.

#52 - 2018-12-05 10:28 - SLindoMansilla
- Status changed from Workable to Blocked
- Assignee changed from SLindoMansilla to okurz
As discussed, this is blocked by the use of the INFO parameter. Please, set priority accordingly.

#53 - 2018-12-05 11:03 - okurz
- Status changed from Blocked to Workable
- Assignee changed from okurz to SLindoMansilla

Actually I think we got confused in the planning meeting. The "info-file-approach" might help with mistyping in a boot prompt but this ticket originally was – and still should be – about "pxe boot menu doesn't show up at all" so much more what we discussed with xlai.

@SLindoMansilla WDYT, still something that you would work on or unassign?

#54 - 2018-12-05 11:43 - SLindoMansilla
Since xlai is working on it, should I then assign it to xlai?

#55 - 2018-12-05 12:43 - okurz
- Assignee changed from SLindoMansilla to xlai

Let's ask her.

xlai I understood you would try an approach calling mc reset so assigning the ticket to you, ok?

I read the whole discussion again and have the following questions:

- Is it that all failing jobs fail in assert_screen((check_var('VIDEOMODE', 'text') ? 'sshd' : 'vnc') . '-server-started', $ssh_vnc_wait_time);? If yes, why not use exactly that as detection for the issue?
- Why do you need any SSH connection to call commands interacting with the MC? The function "ipmitool" in https://github.com/os-autoinst/os-autoinst/blob/master/backend/ipmi.pm#L56 IIUC is called on the worker which is always “reachable”, otherwise the openQA job would not even exist

With my limited knowledge I would propose the following:

- in the boot_from_pxe test module rename "sub run" to "sub try_boot_from_pxe"
- call this method with mc reset based repair until it works or we hit a limit of retries:

```
sub run {
  my ($self) = @_; 
  my $retries = 7;
  for (1 .. $retries) {
    eval { try_boot_from_pxe() };
    return 1 unless ($@);
    backend->ipmitool('mc reset');
  }
  die "Could not boot from PXE over IPMI";
}
```

WDYT?

#56 - 2018-12-06 03:03 - xlai

okurz wrote:

Let's ask her.

xlai I understood you would try an approach calling mc reset so assigning the ticket to you, ok?

Yes, I will take it.

FYI, my solution will not limit to precisely fix boot_from_pxe menu not show up issue. It will be a general solution to ensure ipmi board not become too unstable because of long time not resetting board. So hopefully, with it, it will decrease the happening ratio of this ticket, not can not promise to 0% since ipmi itself is not 100% stable even with reset.

I read the whole discussion again and have the following questions:

- Is it that all failing jobs fail in assert_screen((check_var('VIDEOMODE', 'text') ? 'sshd' : 'vnc') . '-server-started', $ssh_vnc_wait_time);? If yes, why not use exactly that as detection for the issue?

No, various.
Why do you need any SSH connection to call commands interacting with the MC? The function "ipmitool" in https://github.com/os-autoinst/os-autoinst/blob/master/backend/ipmi.pm#L56 IIUC is called on the worker which is always "reachable", otherwise the openQA job would not even exist.

With my limited knowledge I would propose the following:

- in the boot_from_pxe test module rename "sub run" to "sub try_boot_from_pxe"
- call this method with mc reset based repair until it works or we hit a limit of retries:

```perl
sub run {
    my ($self) = @_; 
    my $retries = 7;
    for (1 .. $retries) {
        eval { try_boot_from_pxe() }; 
        return 1 unless ($@);
        backend->ipmitool('mc reset');
    }
    die "Could not boot from PXE over IPMI";
}
```

WDYT?

Yes, call it from backend is a good choice. I also noticed it when look through backend and openqa code yesterday :). Thanks very much for pointing it out.

I would suggest, as a first try, to wrap this 'mc reset' to a test pm file which is loaded before boot_from_pxe. It is supposed to be enough to fix ipmi unstability issue or even more than enough.

If it is not enough, then we will try to precisely reset when various kinds of ipmi unstability is hit. I am afraid it will dirty the test code too much since we have so many points to do it.

#57 - 2018-12-11 03:25 - xlai
- Related to action #44978: [ipmi unstability] jobs got blue screen making openqa needle match does not work added

#58 - 2018-12-19 15:33 - okurz
- Blocks action #45362: [functional][u][ipmi][sporadic] Key press doesn't reach the system added

#59 - 2018-12-21 05:06 - xlai
PR proposed, please help to review!


#60 - 2019-01-08 08:33 - okurz
- Status changed from Workable to Feedback
- Target version changed from Milestone 21 to Milestone 22

PR for os-autoinst merged but not yet deployed due to winter break. Test PR not merged due to os-autoinst PR not yet deployed. Setting target version to M22.

#61 - 2019-01-25 12:25 - mghiessmeier
deployment should be done by now
btw I don't see any boot_from_pxe fails in the last 12 runs

#62 - 2019-01-28 08:58 - xlai

The mc reset flag is only open on workers openqaworker2:23/openqaworker2:24. I checked history jobs on these two workers, 2 failures and both are typing problem on boot_from_pxe -- not related to this mc reset fix, but exposed that typing speed needs to be slower for stability, will look into it later.

Failure job link:
https://openqa.suse.de/tests/2401624#step/boot_from_pxe/24
https://openqa.suse.de/tests/2407340#step/boot_from_pxe/7

#63 - 2019-01-29 05:36 - xlai
Blue screen issue which makes openqa needle matching not work also happens after deployment, https://openqa.nue.suse.com/tests/2410535.
So mc reset can not stop such blue screen unstability.
But shouldn't we need https://github.com/os-autoinst/os-autoinst-distri-opensuse/pull/6424 for this first?

NO. With the backend PR and worker enable mc-reset PR, tests already do mc_reset before boot_from_pxe. The test PR is to provide api to call mc_reset anywhere testers want and we add that before every reboot. So before confirming that mc reset really works to increase stability in boot_from_pxe, adding the test PR is kind of useless.

Blue screen issue which makes openqa needle matching not work also happens after deployment, https://openqa.nue.suse.com/tests/2410535. So mc reset can not stop such blue screen unstability.

During the past 10 days, the two workers that enabled mc_reset flag, openqaworker2:21 and openqaworker2:22 get 1 more blue screen failure, https://openqa.nue.suse.com/tests/2447772. No other ipmi unstable issues.

Meanwhile, the other two workers that are not enabled mc_reset flag (bought nearly same time with similar HW), openqaworker2:23 and openqaworker2:24, got following 5 boot_from_pxe failures:

- https://openqa.nue.suse.com/tests/2447773#step/boot_from_pxe/7 //duplicate typing
- https://openqa.nue.suse.com/tests/2441580 // blue screen issue
- https://openqa.nue.suse.com/tests/2441476 // blue screen issue
- https://openqa.nue.suse.com/tests/2440421#step/boot_from_pxe/11 // no sshd_server_started screen shown

So seems mc_reset helps increase ipmi stability to some extent. Will keep monitoring.
I just collected the statistic for the past month, monitoring boot_from_pxe failures due to unstable ipmi.

Result is UNEXPECTED! OPPOSITE with last time. The two workers with mc_reset(openqaworker2:21 and openqaworker2:22), got nearly twice the failure (total 29 jobs) than the two workers without mc_reset (openqaworker2:23 and openqaworker2:24, total 15 jobs).

#Statistics:

boot_from_pxe failure by unstable ipmi statistic for the past 1 month

##openqaworker2:21(sp.kermit.qa.suse.de):

- total: 248 jobs, [https://openqa.suse.de/admin/workers/977](https://openqa.suse.de/admin/workers/977)
- unstable ipmi resulted failure: total 16 jobs
  - blue screen issue: 13 jobs [https://openqa.suse.de/tests/2764049](https://openqa.suse.de/tests/2764049), [https://openqa.suse.de/tests/2752962](https://openqa.suse.de/tests/2752962), [https://openqa.suse.de/tests/2741647](https://openqa.suse.de/tests/2741647), [https://openqa.suse.de/tests/2739753](https://openqa.suse.de/tests/2739753), [https://openqa.suse.de/tests/2745587](https://openqa.suse.de/tests/2745587), [https://openqa.suse.de/tests/2707450](https://openqa.suse.de/tests/2707450), [https://openqa.suse.de/tests/2767738](https://openqa.suse.de/tests/2767738), [https://openqa.suse.de/tests/2767656](https://openqa.suse.de/tests/2767656), [https://openqa.suse.de/tests/2807608](https://openqa.suse.de/tests/2807608), [https://openqa.suse.de/tests/2790192](https://openqa.suse.de/tests/2790192), [https://openqa.suse.de/tests/2790179](https://openqa.suse.de/tests/2790179), [https://openqa.suse.de/tests/2773265](https://openqa.suse.de/tests/2773265), [https://openqa.suse.de/tests/2767714](https://openqa.suse.de/tests/2767714)
  - not respond to send_key: 2 job [https://openqa.suse.de/tests/2780694](https://openqa.suse.de/tests/2780694), [https://openqa.suse.de/tests/2767714](https://openqa.suse.de/tests/2767714)
  - incomplete screen: 1 job [https://openqa.suse.de/tests/2751067](https://openqa.suse.de/tests/2751067)

##openqaworker2:22(sp.gonzo.qa.suse.de):

- total: 271 jobs, [https://openqa.suse.de/admin/workers/991](https://openqa.suse.de/admin/workers/991)
- unstable ipmi resulted failure: total 16 jobs
  - blue screen issue: 13 jobs [https://openqa.suse.de/tests/2795764](https://openqa.suse.de/tests/2795764), [https://openqa.suse.de/tests/2785240](https://openqa.suse.de/tests/2785240), [https://openqa.suse.de/tests/2785226](https://openqa.suse.de/tests/2785226), [https://openqa.suse.de/tests/2769994](https://openqa.suse.de/tests/2769994), [https://openqa.suse.de/tests/2767723](https://openqa.suse.de/tests/2767723), [https://openqa.suse.de/tests/2746957](https://openqa.suse.de/tests/2746957), [https://openqa.suse.de/tests/2756071](https://openqa.suse.de/tests/2756071), [https://openqa.suse.de/tests/2739619](https://openqa.suse.de/tests/2739619), [https://openqa.suse.de/tests/2739759](https://openqa.suse.de/tests/2739759), [https://openqa.suse.de/tests/2725198](https://openqa.suse.de/tests/2725198), [https://openqa.suse.de/tests/2726650](https://openqa.suse.de/tests/2726650), [https://openqa.suse.de/tests/2735071](https://openqa.suse.de/tests/2735071), [https://openqa.suse.de/tests/2739835](https://openqa.suse.de/tests/2739835)
  - not respond to send_key: 1 job [https://openqa.suse.de/tests/2785239](https://openqa.suse.de/tests/2785239)
  - wrong typing: 2 jobs [https://openqa.suse.de/tests/2737191](https://openqa.suse.de/tests/2737191), [https://openqa.suse.de/tests/2688340](https://openqa.suse.de/tests/2688340), [https://openqa.suse.de/tests/2790192](https://openqa.suse.de/tests/2790192), [https://openqa.suse.de/tests/2770180](https://openqa.suse.de/tests/2770180), [https://openqa.suse.de/tests/2769968](https://openqa.suse.de/tests/2769968), [https://openqa.suse.de/tests/2745561](https://openqa.suse.de/tests/2745561), [https://openqa.suse.de/tests/2747463](https://openqa.suse.de/tests/2747463)

##openqaworker2:23(sp.fozzie.qa.suse.de):

- total: 252 jobs, [https://openqa.suse.de/admin/workers/1089](https://openqa.suse.de/admin/workers/1089)
- unstable ipmi resulted failure: total 9 jobs
  - blue screen issue: 6 jobs [https://openqa.suse.de/tests/2804785](https://openqa.suse.de/tests/2804785), [https://openqa.suse.de/tests/2777904](https://openqa.suse.de/tests/2777904), [https://openqa.suse.de/tests/2770180](https://openqa.suse.de/tests/2770180), [https://openqa.suse.de/tests/2769968](https://openqa.suse.de/tests/2769968), [https://openqa.suse.de/tests/2745561](https://openqa.suse.de/tests/2745561), [https://openqa.suse.de/tests/2747463](https://openqa.suse.de/tests/2747463)
  - wrong typing: 1 job [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218), [https://openqa.suse.de/tests/2726218](https://openqa.suse.de/tests/2726218)
  - slow reaction: 2 jobs [https://openqa.suse.de/tests/2712312](https://openqa.suse.de/tests/2712312), [https://openqa.suse.de/tests/2735348](https://openqa.suse.de/tests/2735348)

##openqaworker2:24(sp.scooter.qa.suse.de):

- total: 256 jobs, [https://openqa.suse.de/admin/workers/1086](https://openqa.suse.de/admin/workers/1086)
- unstable ipmi resulted failure: total 6 jobs
  - blue screen issue: 4 jobs [https://openqa.suse.de/tests/2790375](https://openqa.suse.de/tests/2790375), [https://openqa.suse.de/tests/2778087](https://openqa.suse.de/tests/2778087), [https://openqa.suse.de/tests/2767708](https://openqa.suse.de/tests/2767708), [https://openqa.suse.de/tests/2745581](https://openqa.suse.de/tests/2745581)
  - wrong typing: 2 jobs [https://openqa.suse.de/tests/2791022](https://openqa.suse.de/tests/2791022), [https://openqa.suse.de/tests/2756047](https://openqa.suse.de/tests/2756047)

#Conclusion:

From the two times result, it seems that the ipmi unstable issue is quite random. Doing MC_RESET at the beginning of every ipmi test can not always increase the stability, and sometimes even worse. Seems to prove the comment #49:

```
The most important reason for not do the reset every time is:
Doing 'mc reset' with such higher frequency is an unproved situation -- not sure whether it can really help. Comment#36 to wayne's PR#1047 is also valid explanation to this, since they are similar( do "mc reset " in console's activate function is also frequent reset):
```

#71 - 2019-04-16 07:40 - xlai

I looked through all comments again to recall all discussed proposals and their reasons/advantage/disadvantages. Now it is so hard to find a good
The only experience that we can base on to recover machine from unstable ipmi is via 'mc reset' when unstability happens. However actually even if not do 'mc reset', it is also possible that ipmi recovers after some time. Also even if after 'mc reset', it is not 100% recovered from unstable ipmi, just with high chance.

So if doing 'mc reset' when unstability happens is what we can best do, then we only have following options:

1) to sync with openqa jobs so as to not disturb innocent jobs, then 'mc reset' can only be spun in openqa codes, either test code or backend code, options are:
   - a) do it blindly in every test at beginning(or end) : proves to be not usable in comment #70
   - b) do it at specific points when unstability is detected: however as comment#47 gave, it is hard to STABLELY DETECT whether ipmi sol is unstable or not.
     - b1) blue screen issue: this happens most from data in comment #70, the only way to detect it should be via graphic needle, however when it happens, needle match of openqa does not work
     - b2) wrong typing: it is so random(makes needle not good way), and can not be easily distinguished from repo image issue or network issue when loading pxe
     - b3) slow response or no response to send_key: even harder to detect from point to point

2) not sync with job status, just use jenkins or cron jobs to reset all machines via 'mc reset': this is the initial way, however may disturb all jobs on all ipmi machines when mc reset, so still not a good way, just practical way.

So on such unstable ipmi machine, which is different from one machine to others , and from time to time, it is so hard to find a STABLE GOOD way. Look forward to others' brilliant ideas.

#72 - 2019-04-16 09:49 - cachen
The conclusion in #70 shows Blue Screen is the most serious issue within those SOL unstable samples. Next step can we just focus on this Blue Screen issue? as we all aware it's impossible 100% fix the unstable.

@Alice, do you think needles in boot_from_pxe step for the Blue screen can make it works?

#73 - 2019-04-17 01:58 - xlai

cachen wrote:

The conclusion in #70 shows Blue Screen is the most serious issue within those SOL unstable samples. Next step can we just focus on this Blue Screen issue? as we all aware it's impossible 100% fix the unstable.

@Alice, do you think needles in boot_from_pxe step for the Blue screen can make it works?

@Calen, I agree to fix this blue screen issue first. However from my experience, I have not find any manual workable way to avoid or fix this blue screen issue(mc reset can only fix it at middle chance). Just as I give in comment#71 b1), needle/GUI way should be the way to detect or workaround it, however the reality is when we add that specific blue screen image as needle, this will make openqa needle match does not work at all, from the beginning of test, that is the first needle match of pxe boot which is not this blue screen showing point. So adding such blue screen needles makes openqa needle match not work. This is opened as ticket https://progress.opensuse.org/issues/44976.

#74 - 2019-04-24 06:49 - xlai
- Status changed from Feedback to Workable

Open for solution discussion again.

#75 - 2019-04-28 06:46 - cachen
Add Dawei in the loop.
Dawei is much expert on HW level, he suspects there is something can be tried on XTERM configure to help the stability of ipmi connection. Alice and Dawei will do more investigation and get some tries.

#76 - 2019-05-17 09:54 - mgriessmeier
- Target version changed from Milestone 24 to Milestone 25

#77 - 2019-07-03 09:27 - mgriessmeier
- Target version changed from Milestone 25 to Milestone 26

#78 - 2019-08-20 12:46 - mgriessmeier
- Target version changed from Milestone 26 to Milestone 27

2021-05-21
xlai - there was no progress in here for 4 months... what do you think we should do to move forward here?

#79 - 2019-08-21 05:41 - xlai
mgriessmeier wrote:

    xlai - there was no progress in here for 4 months... what do you think we should do to move forward here?

Yes, i am kind of stuck here for no good enough solution. I am going to visit nuremburg in september and i plan to have a talk with coolo about ipmi unstability solutions. Will let you know the results then.

BTW, as workaround, once ipmi unstability issue happens especially in boot_from_pxe(wrong typing, no screen update, blue screen issue etc), retrigger the tests generally help.

#80 - 2019-08-23 10:00 - SLindoMansilla
- Blocks deleted (action #42383: [sle][functional][u][sporadic][ipmi] test fails in grub_test - does not boot from local disk)

#81 - 2019-08-23 10:00 - SLindoMansilla
- Related to action #42383: [sle][functional][u][sporadic][ipmi] test fails in grub_test - does not boot from local disk added

#82 - 2019-08-23 10:34 - SLindoMansilla
- Blocks deleted (action #41480: [sle][functional][u][ipmi] Malfunction of openqaworker2:25 - Investigate, bring it back or repair it (WAS: remove openqaworker2:25 (IPMI machine) from OSD testing))

#83 - 2019-08-23 10:36 - SLindoMansilla
- Blocks deleted (action #31375: [sle][functional][ipmi][u][hard] test fails in first_boot - VNC installation on SLE 15 failed because of various issues (ipmi worker, first_boot, boot_from_pxe, await_install))

#84 - 2019-08-23 10:37 - SLindoMansilla
- Related to action #31375: [sle][functional][ipmi][u][hard] test fails in first_boot - VNC installation on SLE 15 failed because of various issues (ipmi worker, first_boot, boot_from_pxe, await_install) added

#85 - 2019-08-23 10:38 - SLindoMansilla
- Related to action #41480: [sle][functional][ipmi] Malfunction of openqaworker2:25 - Investigate, bring it back or repair it (WAS: remove openqaworker2:25 (IPMI machine) from OSD testing) added

#86 - 2019-08-23 10:44 - SLindoMansilla
- Blocks deleted (action #38888: [functional][sle][u][sporadic][ipmi] test fails in boot_from_pxe - SOL misbehave booting drivers on linuxrc (text shown repeatedly and in colors))

#87 - 2019-08-23 10:44 - SLindoMansilla
- Related to action #38888: [functional][sle][u][sporadic][ipmi] test fails in boot_from_pxe - SOL misbehave booting drivers on linuxrc (text shown repeatedly and in colors) added

#88 - 2019-08-23 10:48 - SLindoMansilla
- Blocks deleted (action #41693: [sle][functional][u][ipmi][sporadic] test fails in boot_from_pxe - needs to increase ssh_vnc_wait_time)

#89 - 2019-08-23 10:48 - SLindoMansilla
- Related to action #41693: [sle][functional][u][ipmi][sporadic] test fails in boot_from_pxe - needs to increase ssh_vnc_wait_time added

#90 - 2019-09-06 11:58 - pdostal
Hello xlai, another workaround for this scenario (when you reboot the machine and then the virt_autotest/login_console testmodule should continue but not always does so) is to switch to the SOL console in the previous testmodule even before the reboot is executed. See: script_run '{ sleep 15 & & reboot & & }'; save_screenshot; switch_from_ssh_to_sol_console(reset_console_flag => 'on')

#91 - 2019-09-09 02:39 - xlai
pdostal wrote:

Hello xlai, another workaround for this scenario (when you reboot the machine and then the virt_autotest/login_console testmodule should continue but not always does so) is to switch to the SOL console in the previous testmodule even before the reboot is executed. See: [...]

2021-05-21
Glad to see that you are active in this issue. Here is the state:

For ipmi jobs, at the very beginning of test, it is on sol console (and this is the only console that is able to do boot_from_pxe as ssh is not ready yet). For jobs that need to reboot during test, eg reboot after host installation or reboot after updating pkgs to let them take effect, for virtualization tests at least, we actually do what you suggest already -- switch to sol console, then reboot, then login and switch to ssh console again, as tests/virt_autoest/reboot_and_wait_up.pm coded. However ipmi unstability issues happen from time to time both at boot_from_pxe start and during/after reboot, eg sol console not responding, type_string error, blue screen issue which fails needle matching and so on.

#92 - 2019-09-16 08:37 - mgriessmeier
- Target version changed from Milestone 27 to Milestone 28

#93 - 2019-09-23 15:10 - xlai
These are mainly the discussion results between me, coolo and Michael Moese separately:

To improve the test stability which must rely on ipmi somehow, the direction looks like less interaction via ipmi, like no typing, no send_key. For example, pxe installation can be done via ipxe with no typing, booting via grub can be done via grub2-set-default, and final checking after booting can be done via ssh, etc. These test workarounds are quite case by case, depending on every test module and should be handled by test maintainers or users. And they can not be eliminated completely since ipmi itself is not 100% stable.

If with this, still failing on ipmi on some points which make no such workarounds, then adding CI monitors against jobs and doing job retrigger to those which fail at steps mainly relying on ipmi, eg boot_from_pxe and reboot, can be an effective way to avoid manual job retrigger in daily review and save time.

I will add this job monitor and retrigger tool after finishing 15sp2 virtualization preparation, likely November. But for the first part, workaround to use sol less, it should be done by test maintainer or users separately.

#94 - 2019-09-24 17:17 - pvorel
Yes, we moved some tests to use Michie’s iPXE implementation and it works well, see https://openqa.suse.de/tests/3395767
(it requires AUTOYAST=path-to-autoyast.xml, AUTOYAST_PREPARE_PROFILE‘1 IPXE=1)

#95 - 2019-10-18 09:07 - xlai
- Related to action #57587: [virtualization][u] test fails in reboot_and_wait_up_upgrade - Test looks for grub screen but system already booted added

#96 - 2020-01-03 09:16 - mgriessmeier
- Target version changed from Milestone 28 to Milestone 30

needs to be discussed offline

#97 - 2020-03-20 08:56 - xlai
- Status changed from Workable to Resolved

Instead of direct solution, as comment #93 gave, after discussion with coolo, we finally selected the indirect solution -- job retrigger tool, which can monitor openqa jobs and retrigger those failing at ipmi sol related steps. We have used it in a CI way for virtualization and performance tests on osd and openqa.qa2.suse.asia. It works well and saves our daily review and retrigger effort. Meanwhile more timely test results are got comparing with human time to time checking.