openQA Tests - action #65109

[qe-core] Dynamically adjust test waits based on load

2020-03-31 16:43 - dzedro

<table>
<thead>
<tr>
<th>Status:</th>
<th>Workable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Category:</td>
<td>Enhancement to existing tests</td>
</tr>
<tr>
<td>Target version:</td>
<td>future</td>
</tr>
<tr>
<td>Difficulty:</td>
<td></td>
</tr>
<tr>
<td>Start date:</td>
<td></td>
</tr>
<tr>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>% Done:</td>
<td>0%</td>
</tr>
<tr>
<td>Estimated time:</td>
<td>0.00 hour</td>
</tr>
</tbody>
</table>

**Description**

There are some things that could be improved.

Maybe decreasing IO load on workers, snapshots on qemu backend are done basically everywhere even where it doesn't make much sense, small jobs(one, few) tests, multimachine and installations.

I add QEMU_DISABLE_SNAPSHOTS=1 into most of QAM suites, there are installations with following tests, but fail is mostly some random timeout, typing, etc. issue. The test is restarted anyway... It could make sense to disable snapshots by default on suites where snapshot is waste of IO, exceptions could overwrite it with QEMU_DISABLE_SNAPSHOTS=0.

Review testapi, remove/fix inconsistent functions and add functions which are used widely in combinations? e.g. wait_screen_change is widely used and failing where is expected that the function will wait, wait can be more than just screen change, generally is the function waiting 0 seconds. Thus I don't use it, because when you add wait time, mostly in gui/ncurses tests, you expect that it will wait some minimal time, not zero.

I guess there is some screen change like button animation, but that does not mean the action is finished and next step should continue safely.

Exactly for this I prefer wait_still_screen, as you can define min/max wait time based on screen activity.

send_key is often used with wait_still_screen or wait_screen_change, despite wait_screen_change is part of send_key as parameter to wait after key press. I guess most of people don't know that there is parameter in send_key for wait_screen_change, which is still questionable if it will be sufficient wait function.

Btw I was thinking that the parameter is not working since it was using wait_idle, which became deprecated. In one tests was created send_key_and_wait [1], use of wait_still_screen could be parameter of send_key, replace or be second parameter with wait_screen_change?

**Acceptance criteria**

- **AC1:** Unify implementation of max_interval/wait_still_screen/wait_screen_change for send_key and other testapi functions.
- **AC2:** Reduce I/O overhead proactively

**Suggestions**

- **AC1:**
  - Extend send_key with a wait_still_screen option
  - Refactor wait logic for better re-use
- **AC2:**
  - Enable and disable QEMU_DISABLE_SNAPSHOTS systematically
  - Evaluate resource usage e.g. CPU load, I/O causing timeouts

[1] [https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/yast2_gui/yast2_instserver.pm#L27](https://github.com/os-autoinst/os-autoinst-distri-opensuse/blob/master/tests/yast2_gui/yast2_instserver.pm#L27)

**History**

#1 - 2020-03-31 17:37 - cdywan

Note for the benefit of drive-by readers: This is based on previous notes, and will need some refinement. I intend to identify some actionable points from this.

#2 - 2020-04-15 16:54 - cdywan
dzedro wrote:

send_key is often used with wait_still_screen or wait_screen_change, despite wait_screen_change is part of send_key as parameter to wait after key press. I guess most of people don't know that there is parameter in send_key for wait_screen_change, which is still questionable if it will be sufficient wait function.

Btw I was thinking that the parameter is not working since it was using wait_idle, which became deprecated. In one tests was created send_key_and_wait [1], use of wait_still_screen could be parameter of send_key, replace or be second parameter with wait_screen_change?


So the current version of send_key_and_wait is basically send_key $key, $stilltime, $timeout//=5; wait_still_screen $stilltime, $timeout: On average it's called with a $stilltime of 2 seconds... so I see two attractive options:

1. Extending send_key with a new $wait_still_screen option, like type_string/password would make sense for consistency/simplicity
2. Adding send_key_still_screen which defaults to 2 seconds stilltime and 5 seconds timeout.

To further put this to the test, we could merge all calls of the send_key_and_wait function and see if that works well.

dzedro wrote:

Maybe decreasing IO load on workers, snapshots on qemu backend are done basically everywhere even where it doesn't make much sense, small jobs(one, few) tests, multimachine and installations.

I add OEMU_DISABLE_SNAPSHOTS=1 into most of QAM suites, there are installations with following tests, but fail is mostly some random timeout, typing, etc. issue. The test is restarted anyway... It could make sense to disable snapshots by default on suites where snapshot is waste of IO, exceptions could overwrite it with OEMU_DISABLE_SNAPSHOTS=0.

Maybe the setting should be moved to a different layer? The job group, not the test, for instance.

okurz wrote:

dzedro should we move this to “openQA project“ or is your intention to discuss this within the scope of “test maintainers“ mainly?

I think you are raising many important points and it is good to see that you care about general performance and stability which I only see in a very limited scope among users of openQA.

In general I think we have a lot of potential for improvement as mainly we are using a system from around 2013-2015 and there had been a lot of improvements in the domain of mainly virtualized environments since then, e.g. improvement using newer qemu features, virtio, qcow tune parameters, filesystems from which machines are started, etc.

Anybody is welcome, to discuss such change.

cdywan I would vote for option 1. extend existing send_key.
I was disabling it in specific tests where snapshots are pointless, when you have big group and you don't know exactly where you want to have snapshots then per test.
dzedro wrote:

cdywan I would vote for option 1. extend existing send_key.

https://github.com/os-autoinst/os-autoinst/pull/1418

#8 - 2020-05-26 16:42 - cdywan
- Status changed from In Progress to Feedback

I closed the above PR for now since I don't necessarily want to address just the obvious symptoms, but it already helped inspire some interesting conversations.

#9 - 2020-07-23 15:23 - cdywan
- Description updated

#10 - 2020-08-28 11:56 - cdywan

- Default timeout based on (over)load
  - Query host load data (not VM guest/ worker)
  - (e.g. load average: 2.48, 1.90, 1.91)
  - 8 CPUs, 16 tasks loaded => timeout of 8 seconds
  - timeout += (load/cpu) > 1 ? (load / cpu) * seconds(=1) : 0
  - Swappiness (free, Perl API?)
  - timeout += swap ? (seconds=1) : 0
  - I/O load (iostat, Perl API?)
  - timeout += ?
  - Networking
    - Useful? may not be relevant to our setups
- Wait on ... instead of seconds
  - cpu idle cycles (context switches?)
  - disk sync
- Conceive a test with artificially busy processes
  - HUNGRY_RAM=n (malloc(1024*1024*1024;))
  - HUNGRY_PROCESSES=n (yes >/dev/null)
  - HUNGRY_DISKS=n
  - artificially hungry node app

#11 - 2020-10-02 10:41 - cdywan
- Subject changed from [tools] backend setting & API improvement to Dynamically adjust test waits based on load
- Status changed from Feedback to Workable
- Assignee deleted (cdywan)

#12 - 2020-10-04 11:37 - okurz
- Target version changed from Current Sprint to future

With your update I think some useful ideas have been collected. I will remove the ticket from the current SUSE QA Tools team backlog. So this is left for another team to pick up. In general I think the ideas in this ticket are more important than most test extensions to have even more test coverage which gets hard to maintain.

#13 - 2020-10-29 13:50 - okurz
- Assignee set to tjyrinki_suse

- tjyrinki_suse I think this is one of many tickets regarding general "improvement" of openQA tests so could be something for "QE Core" hence assigning to you. As this is just one tiny example of many many other cases I would be happy to also discuss this with you in a synchronous meeting between "QE Tools" and "QE Core" PO if you are interested.

#14 - 2021-04-29 13:05 - tjyrinki_suse
- Subject changed from Dynamically adjust test waits based on load to [qe-core] Dynamically adjust test waits based on load
- Start date deleted (2020-03-31)

#15 - 2021-04-29 13:05 - tjyrinki_suse
- Assignee deleted (tjyrinki_suse)