

openQA Project - action #14582

Add virtio serial console backend and API

2016-10-31 10:40 - rpalethorpe

Status:	Resolved	Start date:	2016-10-31
Priority:	High	Due date:	
Assignee:	rpalethorpe	% Done:	50%
Category:	Feature requests	Estimated time:	0.00 hour
Target version:			
Difficulty:			

Description

I have written a new console backend which allows IO through a serial terminal, in particular the virtio console with QEMU, but I am currently thinking about how to generalise it. I started out mostly interested in how virtio could be used to speed up testing, but actually it has turned out to be mostly irrelevant. The most important thing is that communication is done as if a user is typing text into a serial terminal. For platforms which can be controlled entirely (or only) over UART (or UART over USB/Bluetooth/whatever) this opens up quite a few possibilities.

From <https://github.com/os-autoinst/os-autoinst/pull/637>:

```
This allows the test writer to log into and interact with a serial terminal directly. Initially this is just for the virtio_console under QEMU, but can be extended to any serial console.
```

```
Presently the only way of sending text to a tty running on a SUT is to send the keystrokes via VNC. Output from the SUT can be redirected to and read from, a serial port, however the test writer can not send text directly to the serial port without circumventing the test API. This patch introduces a new console backend which can be selected in the same manner that the existing console backends are, but is limited to text input and output. This means that there is no video feed, but entering commands is orders of magnitude faster because the commands are sent and interpreted as text rather than simulated keystrokes.
```

```
This adds the is_serial_terminal subroutine to the testapi which is exported on request. Otherwise the API should be unchanged and fully backwards compatible.
```

Currently the virtio serial terminal must be requested by the test case. I think this is fine for now, because it is primarily needed for the LTP native runner I am also working on. However after further testing and modifications to the UI to display a text feed rather than a video, I think it can be automatically used when 'root-console' (or similar) is requested. Further improvements and features may include:

Display serial log in OpenQA UI

Currently a bitmap feed is displayed, which is nice, and using a serial terminal breaks that. The user can still see, to some extent, what is happening in the virtual machine from the os-autoinst log feed. However it would be nice if they could see the tty log in real time. All that needs to happen is that either os-autoinst relays IO from the serial socket to the UI or the UI reads a log file containing the serial terminal output (e.g. QEMU produces such a file). There is probably a javascript library for handling terminal escape codes, but otherwise the raw text can just be displayed. Alternatively an image could be generated of the text terminal, but that will increase network and processor load.

Use serial terminal whenever possible

This should be easy to enable, but there are two issues. One is user experience, which I have discussed above. The second is the possibility of bugs caused by subtle differences in the testapi behaviour when switching to a serial terminal. If we explicitly switch a few different tests over to using serial, then we can probably catch most problems without exposing the entire test suite to them at once.

Implement it for other platforms

To my knowledge, all the backends have some ability to read input from a serial port. Ideally we will want to open a second serial

port which OpenQA is not already using, but it should also be possible to use the existing one. It is then a case of generalising the existing code for virtio_console to read from a different socket. If a backend only has one serial port then we perhaps have to come up with a mechanism for ignoring kernel log messages which are usually sent to the serial port.

Related issues:

Related to openQA Project - coordination #14626: [epic] backend and console c...	New	2019-02-17
Related to openQA Project - action #14690: Live stream for serial terminal	Resolved	2016-11-08

History

#1 - 2016-10-31 14:32 - rpalethorpe

- Checklist item changed from to [] Add virtio_console.log to result files or logs

#2 - 2016-10-31 15:39 - okurz

maybe you want to use the template from <https://progress.opensuse.org/projects/openqav3/wiki#feature-requests> to follow the common scheme for "feature requests". If I understand correctly there are two benefits:

- supporting platforms which we can't run in virtual machines with a VNC connection or physical setups which have only a serial port available
- potential speed improvement of the use of the serial console

If this is true I would be interested what use cases we have that require this.

#3 - 2016-10-31 20:41 - RBrownSUSE

Oli, your Template is not an approved process by the openQA community, nor approved by either of the project leads (Cools and myself)

Please remove the wiki article, or at least stop demanding people use a process which you concocted on your own

Consider people's feature requests on THEIR own merits, not just your own criteria.

#4 - 2016-11-01 10:02 - rpalethorpe

okurz wrote:

maybe you want to use the template from <https://progress.opensuse.org/projects/openqav3/wiki#feature-requests> to follow the common scheme for "feature requests". If I understand correctly there are two benefits:

- supporting platforms which we can't run in virtual machines with a VNC connection or physical setups which have only a serial port available
- potential speed improvement of the use of the serial console

If this is true I would be interested what use cases we have that require this.

I have discussed this at great length with Stephan on the open-qa mailing list (for reference). Basically the main use case is LTP and possibly kernel self tests, but any test which makes heavy use of a text terminal will be much faster. There are other ways of getting the speed, but they add layers of separation between OpenQA and the LTP test case executable/script. I want someone to be able to look at the Perl modules in OpenQA and see exactly how the LTP tests are being executed and I want the Perl module to know exactly what was executed and when (in so far that it is possible). This means entering LTP test commands directly on the terminal which, for example, takes approx. 5.4 seconds for the math/abs01 LTP test, when entering the text through a terminal it is negligible (the whole process takes approx. 0.003 seconds including test execution). Considering that many of the tests take under a second to run it means we are spending more time entering the command rather than running it. I haven't got around to measuring the speed increase with tests other than the LTP runner yet, but the performance increase should be large. For this test <https://openqa.suse.de/tests/627318#> it should be significant even though the ncurses UI will prevent it being used some of the time, you can see this by watching the video and seeing how much time is spent typing in commands.

Ignoring LTP for a second this should also be a great advantage for the log collection modules, which spend the vast majority of their time entering text and are used extensively throughout the SUSE test suite. This is not my main concern, but it is probably of the most interest to the majority of OpenQA users. I accept that this feature results in a moderate increase in complexity, but the speed advantages are so great that I believe it to be more than worth it. Also the serial port is already heavily used in OpenQA, many testapi functions will not work without wait_serial so I am not doing anything completely alien to os-autoinst.

Stuff like supporting platforms which have no VNC capability is very interesting for bare metal testing of unusual architectures or embedded systems, but I only thought of that yesterday. I think this feature is completely justified by the LTP's requirements and that is what is driving it, but at the same time I want to maximise the benefit of the feature.

#5 - 2016-11-01 16:53 - okurz

Thank you very much for an excellent explanation. Yes, I agree with the benefits you describe. We must not make any serial console mandatory, e.g. consider "MS Windows testing" which relies only on assert_screen and type_string and send_key, I assume.

#6 - 2016-11-03 08:23 - rpalethorpe

- Status changed from New to In Progress

- Priority changed from Normal to High

- % Done changed from 0 to 50

#7 - 2016-11-03 13:29 - rpalethorpe

- Related to coordination #14626: [epic] backend and console capabilities interface to increase extensibility and code reuse added

#9 - 2016-11-07 15:37 - rpalethorpe

- Checklist item changed from [] Add virtio_console.log to result files or logs to [] Add virtio_console.log to result files or logs, [] Modify SUSE test suite helper functions to be serial console compatible or workaround them

#10 - 2016-11-08 14:32 - rpalethorpe

- Related to action #14690: Live stream for serial terminal added

#11 - 2016-11-24 08:08 - rpalethorpe

- Checklist item changed from to [x] Add virtio_console.log to result files or logs

#12 - 2016-11-30 14:36 - rpalethorpe

- Checklist item changed from [x] Add virtio_console.log to result files or logs, [] Modify SUSE test suite helper functions to be serial console compatible or workaround them to [x] Add virtio_console.log to result files or logs

- Status changed from In Progress to Resolved