openQA Project - action #69700

Predefined QEMU hardware profiles in os-autoinst

2020-08-07 10:15 - MDoucha

Status:	New	**Start date:**	
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Feature requests	**Estimated time:**	0.00 hour
Target version:	future		
Difficulty:			

Description

We've recently had a regression that made our kernels unbootable in QEMU VMs created in virt-manager. The regression was missed by OpenQA tests because os-autoinst uses VMs with minimal hardware configuration which didn't trigger the bug.

We should define multiple QEMU hardware profiles (named sets of extra device options for QEMU) which can then be selected through job settings. The hardware profiles don't need to cover every possible combination of devices, it'll be enough if each device model appears in them at least once. One of the profiles should be as close to virt-manager defaults as possible. Then it'll be sufficient to boot the existing LTP jobs on different hardware profiles. We don't need any extra tests beyond checking that the kernel is bootable.

Example profile that would trigger the regression:

```
-machine pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off
-device pcie-root-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x2
-device pcie-root-port,port=0x11,chassis=2,id=pci.2,bus=pcie.0,addr=0x2.0x1
-device pcie-pci-bridge,id=pci.9,bus=pci.2,addr=0x0
```

Related issues:

Related to openQA Tests - action #9590: [sle][sled] Laptop tests for SLED	**Resolved**	**2015-11-17**

History

#1 - 2020-08-07 10:16 - MDoucha

- *Category set to Feature requests*

- *Priority changed from Normal to High*

#2 - 2020-08-07 10:45 - okurz

- *Related to action #9590: [sle][sled] Laptop tests for SLED added*

#3 - 2020-08-07 10:47 - okurz

- *Tags set to os-autoinst, backend, qemu*

- *Assignee set to okurz*

- *Target version set to Ready*

The related ticket #9590 shows another example using a "laptop profile" by loading according DMI data which simulates a laptop. I will look into this shortly.

#4 - 2020-08-07 11:31 - okurz

- *Status changed from New to Feedback*

Experiment

I think we can easily solve this with existing test parameters without further code changes needed. E.g. I just tested from an os-autoinst checkout directory:

```
$ (cd t/data && timeout -s INT -v 2 ../../isotovideo -d casedir=/home/okurz/local/os-autoinst/os-autoinst/t/da
ta/tests qemumachine=pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off) |& grep --color qemu-kvm
[2020-08-07T13:04:16.073 CEST] [debug] starting: /usr/bin/qemu-kvm -only-migratable -chardev ringbuf,id=serial
0,logfile=serial0,logappend=on -serial chardev:serial0 -audiodev none,id=snd0 -device intel-hda -device hda-ou
```

```
tput,audiodev=snd0 -global isa-fdc.driveA= -m 1024 -machine pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest
-core=off -netdev user,id=qanet0 -device virtio-net,netdev=qanet0,mac=52:54:00:12:34:56 -boot once=d -device u
sb-ehci -device usb-tablet -smp 1 -enable-kvm -no-shutdown -vnc :90,share=force-shared -device virtio-serial -
chardev pipe,id=virtio_console,path=virtio_console,logfile=virtio_console.log,logappend=on -device virtconsole
,chardev=virtio_console,name=org.openqa.console.virtio_console -chardev socket,path=qmp_socket,server,nowait,i
d=qmp_socket,logfile=qmp_socket.log,logappend=on -qmp chardev:qmp_socket -S -device virtio-scsi-pci,id=scsi0 -
blockdev driver=file,node-name=hd0-file,filename=/home/okurz/local/os-autoinst/os-autoinst/t/data/raid/hd0,cac
he.no-flush=on -blockdev driver=qcow2,node-name=hd0,file=hd0-file,cache.no-flush=on -device virtio-blk,id=hd0-
device,drive=hd0,serial=hd0
```

so the test parameter QEMUMACHINE=pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off sets the according command line parameter -machine pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off.

And for one of the other profiles:

```
$ (cd t/data && timeout -s INT -v 2 ../../isotovideo -d casedir=/home/okurz/local/os-autoinst/os-autoinst/t/da
ta/tests qemu_append="device pcie-root-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x2"
) |& grep --color 'pcie-root-port'
[2020-08-07T13:06:30.976 CEST] [debug] Setting forced test parameter QEMU_APPEND -> device pcie-root-port,port
=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x2
[2020-08-07T13:06:31.198 CEST] [debug] starting: /usr/bin/qemu-kvm -only-migratable -chardev ringbuf,id=serial
0,logfile=serial0,logappend=on -serial chardev:serial0 -audiodev none,id=snd0 -device intel-hda -device hda-ou
tput,audiodev=snd0 -global isa-fdc.driveA= -m 1024 -netdev user,id=qanet0 -device virtio-net,netdev=qanet0,mac
=52:54:00:12:34:56 -boot once=d -device usb-ehci -device usb-tablet -smp 1 -enable-kvm -no-shutdown -vnc :90,s
hare=force-shared -device virtio-serial -chardev pipe,id=virtio_console,path=virtio_console,logfile=virtio_con
sole.log,logappend=on -device virtconsole,chardev=virtio_console,name=org.openqa.console.virtio_console -chard
ev socket,path=qmp_socket,server,nowait,id=qmp_socket,logfile=qmp_socket.log,logappend=on -qmp chardev:qmp_soc
ket -S -device pcie-root-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x2 -device virtio
-scsi-pci,id=scsi0 -blockdev driver=file,node-name=hd0-file,filename=/home/okurz/local/os-autoinst/os-autoinst
/t/data/raid/hd0,cache.no-flush=on -blockdev driver=qcow2,node-name=hd0,file=hd0-file,cache.no-flush=on -devic
e virtio-blk,id=hd0-device,drive=hd0,serial=hd0
```

so the test parameter QEMU_APPEND=device pcie-root-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x2 (with quotes on command line) adds the corresponding device. However, the slot-function by default is already occupied by "hda":

```
[2020-08-07T13:06:32.720 CEST] [debug] QEMU: qemu-system-x86_64: -device pcie-root-port,port=0x10,chassis=1,id
=pci.1,bus=pcie.0,multifunction=on,addr=0x2: PCI: slot 2 function 0 not available for pcie-root-port, in use b
y intel-hda
```

This can be worked around by selecting a different slot-function or with QEMU_SOUNDHW=pcspk. But then other devices, e.g. USB and the network adapter already occupy any next address. Probably easier to select a higher, free address like addr=0x7 and then addr=0x7.0x1 respectively. The complete command with output looks like:

```
$ (cd t/data && timeout -s INT -v 2 ../../isotovideo -d casedir=/home/okurz/local/os-autoinst/os-autoinst/t/da
ta/tests qemumachine=pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off qemu_append="device pcie-root
-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x7 -device pcie-root-port,port=0x11,chass
is=2,id=pci.2,bus=pcie.0,addr=0x7.0x1 -device pcie-pci-bridge,id=pci.9,bus=pci.2,addr=0x0") |& grep --color 'p
cie-root-port'
[2020-08-07T13:27:08.507 CEST] [debug] Setting forced test parameter QEMU_APPEND -> device pcie-root-port,port
=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x7 -device pcie-root-port,port=0x11,chassis=2,id=pc
i.2,bus=pcie.0,addr=0x7.0x1 -device pcie-pci-bridge,id=pci.9,bus=pci.2,addr=0x0
[2020-08-07T13:27:08.726 CEST] [debug] starting: /usr/bin/qemu-kvm -only-migratable -chardev ringbuf,id=serial
0,logfile=serial0,logappend=on -serial chardev:serial0 -audiodev none,id=snd0 -device intel-hda -device hda-ou
tput,audiodev=snd0 -global isa-fdc.driveA= -m 1024 -machine pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest
-core=off -netdev user,id=qanet0 -device virtio-net,netdev=qanet0,mac=52:54:00:12:34:56 -boot once=d -device u
sb-ehci -device usb-tablet -smp 1 -enable-kvm -no-shutdown -vnc :90,share=force-shared -device virtio-serial -
chardev pipe,id=virtio_console,path=virtio_console,logfile=virtio_console.log,logappend=on -device virtconsole
,chardev=virtio_console,name=org.openqa.console.virtio_console -chardev socket,path=qmp_socket,server,nowait,i
d=qmp_socket,logfile=qmp_socket.log,logappend=on -qmp chardev:qmp_socket -S -device pcie-root-port,port=0x10,c
hassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x7 -device pcie-root-port,port=0x11,chassis=2,id=pci.2,bus
=pcie.0,addr=0x7.0x1 -device pcie-pci-bridge,id=pci.9,bus=pci.2,addr=0x0 -device virtio-scsi-pci,id=scsi0 -blo
ckdev driver=file,node-name=hd0-file,filename=/home/okurz/local/os-autoinst/os-autoinst/t/data/raid/hd0,cache.
no-flush=on -blockdev driver=qcow2,node-name=hd0,file=hd0-file,cache.no-flush=on -device virtio-blk,id=hd0-dev
ice,drive=hd0,serial=hd0
```

Recommendation

- Define openQA machine definitions for each selected profile
- Set machine test parameters, e.g.

```
QEMUMACHINE=pc-q35-4.2,accel=kvm,usb=off,vmport=off,dump-guest-core=off
QEMU_APPEND=device pcie-root-port,port=0x10,chassis=1,id=pci.1,bus=pcie.0,multifunction=on,addr=0x7 -device pc
ie-root-port,port=0x11,chassis=2,id=pci.2,bus=pcie.0,addr=0x7.0x1 -device pcie-pci-bridge,id=pci.9,bus=pci.2,a
ddr=0x0
```

- Schedule tests against each machine variant

MDoucha is this enough for you?

#5 - 2020-08-07 11:53 - MDoucha

okurz wrote:

> I think we can easily solve this with existing test parameters without further code changes needed. E.g. I just tested from an os-autoinst checkout directory:

That was just the minimal bug reproducer. The virt-manager-defaults profile would have around 30 arguments and I really do **NOT** want to cram that much text into testsuite definitions. Not to mention that those profiles may need arch-specific variations.

#6 - 2020-08-07 13:49 - okurz

ok, but I guess we also do not want to cram that much text into backend/qemu.pm within os-autoinst either especially because manually maintaining them to keep in sync with what virt-manager-defaults provides isn't feasible either. https://libvirt.org/drvqemu.html#xmlexport shows how libvirt parameters can be exported into qemu command line parameters, maybe this is helpful for you as well? As mentioned already in chat, there is the backend "svirt" https://github.com/os-autoinst/os-autoinst/tree/master/backend/svirt.pm already existing that allows to just use libvirt so you can use the pre-defined profiles. This is something that the QA SLE Virtualization team already uses. Would this work? It would be helpful if you could state "acceptance criteria" for this ticket which are independant of the implementation. Then we probably know better what should be achieved.

#7 - 2020-08-07 14:59 - MDoucha

okurz wrote:

> ok, but I guess we also do not want to cram that much text into backend/qemu.pm within os-autoinst either especially because manually maintaining them to keep in sync with what virt-manager-defaults provides isn't feasible either.

We don't need to keep 100% in sync. Updating our virt-manager-defaults profile once a year would be more than enough. And you don't need to cram the profiles into backend/qemu.pm, they can be defined in a separate perl file.

Acceptance criteria

- Create a predefined set of named lists of QEMU command line arguments that can be selected using job settings. E.g. QEMU_HWPROFILE=virt-manager-defaults.
- The profiles should support all QEMU arguments, not just a subset converted from libvirt XML.
- Some conditional profile variation may be needed (e.g. allowing QEMUMACHINE to override -machine defined in the profile).
- Move part of currently hardcoded QEMU settings into a default profile that will be used when no other profile is selected. E.g. audio and network device so that we can switch between hda/ac97 and virtionet/e1000.
- The same profile names should be defined for all archs. The HW configurations with the same name should be reasonably similar but don't need to be identical.
- Optional: Allow loading of custom profile from $CASEDIR E.g. QEMU_HWPROFILE=path/to/file.txt. This should be used only for development, not regular testing. No conditional variation allowed here.

#8 - 2020-08-07 15:48 - okurz

- Status changed from Feedback to New

- Assignee deleted (okurz)

- Target version changed from Ready to future

Thanks for your effort. There are a bit more step-by-step implementation suggestions rather than acceptance criteria. I agree that some more currently hardcoded settings, e.g. for audio and network devices could be made configurable. I am not sure about how to define the profiles though and the virt-manager-defaults copy but I guess these do not change that often that it would be a problem.

To not override your priority selection I have set "future" though to denote that the team SUSE QA Tools does not plan to do that anytime soon. Should be feasible by others to do though. The alternative would have been to set "Low" prio and target version "Ready" to appear on the backlog of SUSE QA Tools.

#11 - 2020-09-29 09:57 - okurz

- Project changed from openQA Project to openQA Tests

- Subject changed from Predefined QEMU hardware profiles in os-autoinst to [qam][u] Predefined QEMU hardware profiles in os-autoinst

- Category changed from Feature requests to New test

- Priority changed from High to Normal

As was clarified in #70450 this is not considered a blocker. Also since my last comment there was no action by anyone. As stated I think the idea is feasible and useful but does not need any low-level architectural changes within os-autoinst or openQA. So probably it is better to have this within "openQA Tests" with according team tags?

#12 - 2020-11-06 09:36 - tjyrinki_suse

- Subject changed from [qam][u] Predefined QEMU hardware profiles in os-autoinst to [qe-core][u] Predefined QEMU hardware profiles in os-autoinst

#13 - 2020-11-06 10:34 - tjyrinki_suse

- Subject changed from [qe-core][u] Predefined QEMU hardware profiles in os-autoinst to [qe-core] Predefined QEMU hardware profiles in os-autoinst

#14 - 2020-11-26 11:40 - tjyrinki_suse

- Subject changed from [qe-core] Predefined QEMU hardware profiles in os-autoinst to Predefined QEMU hardware profiles in os-autoinst

- Target version changed from future to Ready

- Start date deleted (2020-08-07)

Maybe similar to ticket #69712 this would be a generic os-autoinst feature for Tools team.

#15 - 2020-11-26 16:02 - okurz

- Project changed from openQA Tests to openQA Project

- Category changed from New test to Feature requests

- Priority changed from Normal to Low

- Target version changed from Ready to future

Well, I do understand this ticket but I don't understand #69712 yet so I can't say if this is really related :) But let's see if someone clarifies it.

However for this ticket I already stated in #69700#note-11 "the idea is feasible and useful but does not need any low-level architectural changes within os-autoinst or openQA". I don't want to move this back and forth. And if no team wants to followup with this then we can leave it in "openQA Project" under "future" but it will be unlikely this would be implemented anytime soon then.