openQA Project - action #30649

[tools][openqa] Improve performance by using migrations and external snapshots

2018-01-22 12:20 - rpalethorpe

Status:	Resolved	**Start date:**	2018-04-24
Priority:	Normal	**Due date:**	
Assignee:	rpalethorpe	**% Done:**	100%
Category:	Feature requests	**Estimated time:**	0.00 hour
Target version:			
Difficulty:			

Description

Sometimes snapshots fail to save, see https://bugzilla.suse.com/show_bug.cgi?id=1035453. This is of high importance to kernel team because the LTP test runner now makes heavy use of snapshots.

According to the QEMU developers this is because 'internal' snapshots are slow and relatively untested so it is recommended that we use 'external' snapshots combined with the migration functionality[1]. This is currently how libvirt works when taking a snapshot. The downside to this is that it is more complex than simply calling savevm and loadvm.

It makes sense to fix upstream QEMU however this could potentially take a long time[2]. Therefore I think the best thing to do is to first implement a new snapshot method within OpenQA (os-autoinst) then consider making changes to QEMU based on the results. Ideally we want to align OpenQA with the common use case which is being actively maintained.

Alternatively we could convert the QEMU backend to use libvirt (or combine it with the existing virsh backend). However, this only removes some of the complication, but at the same time introduces another layer of indirection. It would be quite a large undertaking so I would put it outside of the scope of this task, at least to begin with.

From what I have seen, the new snapshot process would look something like this:

- Start QEMU with the deferred migration flag
- ...Do some work...
- Pause the virtual machine
- For each block storage device: start an incremental snapshot to an external file
- Save the CPU, RAM and other device state by migrating the VM to a file[3]
- Unpause the VM
- ...Continue until something bad happens...
- Pause the VM
- For each storage device: restore the corresponding snapshot file
- Restore the CPU, RAM and other device state by starting an incoming migration
- Unpause the VM

The details of how to do this should be in the libvirt source. The worst part is migrating to a file which will possibly require passing a file handle to QEMU using SCM rights or opening another socket which it can send the data to.

[1] https://www.mail-archive.com/qemu-devel@nongnu.org/msg504839.html
[2] Ideally we want a clean simple interface which requires little knowledge about QEMU's internal workings. However the QMP interface is necessarily low level which conflicts with ease of use.
[3] Note we are not performing a 'migration', just using the migration command to save the VM's state to a file which could then be used in a real migration. Obviously this does not include the storage device data which is taken care of separately.

Subtasks:

action # 32968: [kernel][tools] Refactor QEMU backend - Create QEMU process manager and...	**Resolved**
action # 35407: [kernel][tools] QEMU Refactor - Serialise state and reimplement SKIPTO	**Resolved**
action # 35431: [kernel][tools] QEMU Refactor - Clean up miscellaneous weird stuff	**Resolved**
action # 35434: [kernel][tools] QEMU Refactor - Ensure consistent use of List::Util, ma...	**Resolved**
action # 35437: [kernel][tools] QEMU Refactor - Publish disk	**Resolved**
action # 35440: [kernel][tools] QEMU Refactor - Code format and rebase	**Resolved**
action # 35443: [kernel][tools] QEMU Refactor - Acceptance testing	**Resolved**
action # 35815: [kernel][tools] Refactor QEMU backend - Fix VNC installation console sw...	**Resolved**
action # 36034: [kernel][tools] QEMU Refactor - Regression, first Grub boot fails after...	**Rejected**

action # 36460: [kernel][tools] QEMU Refactor - Performance settings		**Resolved**

Related issues:

Related to openQA Project - action #19390: [tools][sprint 201711.2] qemu "mig...	**Resolved**	**2018-01-12**

History

#1 - 2018-01-22 12:31 - coolo

As we have so much fun with migrate command: https://progress.opensuse.org/issues/19390

#2 - 2018-01-22 14:58 - rpalethorpe

Currently we are using it to pipe an entire memory dump into bzip then redirecting that to a file through the shell. Despite the fact this is the documented way of doing it, I don't think it is the most common way. Libvirt is passing a file descriptor using SCM rights (with the option of doing an incremental dump which is useful for snapshots), so it is possible that if we do it the same way then we will avoid some bottleneck in the exec code. Although I don't see what could be wrong with the exec code in QEMU.

Also libvirt calls migrate_set_speed with the largest value possible before taking a snapshot. I think it is possible that QEMU automatically throttles migrations to prevent them from using all the bandwidth so we should probably do the same when migrating to a file.

#3 - 2018-01-22 15:00 - rpalethorpe

- Related to action #19390: [tools][sprint 201711.2] qemu "migrate" within testapi::save_memory_dump command never finishes within 2h added

#4 - 2018-01-23 15:23 - rpalethorpe

OK, so migrating to and from a file works, however it requires restarting QEMU. That probably means refactoring the start_qemu os-autoinst code quite a lot. Alternatively it should be possible patch QEMU to allow incoming migrations without a restart.

Before doing that it is probably a good idea to try making the memory dump reliable. If it suffers from the same problems as savevm (after applying all the fixes I can think of) then there are probably better things to be doing.

#6 - 2018-01-24 15:38 - rpalethorpe

It should also be possible to enable performance monitoring during savevm and memory dump which might help reveal where the problem is.

#7 - 2018-01-25 14:57 - rpalethorpe

- Priority changed from Normal to High

#8 - 2018-01-25 16:46 - rpalethorpe

On a semi-related note it looks like snapshots are adding 100GB to the image size for the image published by install_ltp

https://github.com/os-autoinst/os-autoinst-distri-opensuse/pull/4283

#9 - 2018-02-14 11:22 - rpalethorpe

From ongoing discussions on the QEMU mailing list it appears various people are working on related problems and solutions for snapshots. I'm not sure I understand all of the problems and solutions, but I think that we should be able to patch QEMU so that it allows an incoming migration without being restarted. This might turn up some bugs or problems which were previously hidden in QEMU (maybe not a bad thing), but it is probably the least intrusive change. It seems to be worth further investigation. Some people appear to be working on solutions which look better than this, but they may take a long time.

#10 - 2018-02-23 15:57 - rpalethorpe

It appears that incoming migrations do work in a non-pristine QEMU with a few minor changes, the patch is fairly simple so far: https://github.com/richiejp/qemu/commit/788156d104079ef0deb9e48048a7995f1995dc22. However I wonder what kind of edge cases there are and what run states should be considered a valid starting point.

#11 - 2018-02-23 16:12 - coolo

Well, find out what upstream tells you about the patch. In the past, this turned into a dead end often enough :(

#12 - 2018-02-27 13:03 - rpalethorpe

- Status changed from New to Blocked

I don't think they will disagree in principle as I got the idea from them, but I have sent an RFC patch and will wait to see what happens. It will probably require a lot of testing, but learning about the QEMU testing framework may be useful regardless.

#13 - 2018-02-27 13:05 - rpalethorpe

- Target version set to Milestone 15

#14 - 2018-02-28 09:09 - rpalethorpe

The mail archive link: http://lists.nongnu.org/archive/html/qemu-devel/2018-02/msg06782.html

#15 - 2018-03-05 16:47 - rpalethorpe

- Status changed from Blocked to In Progress

So far from the discussion upstream there are two things which need to be investigated:

```
1) Deleting the current "active layer" and making a new overlay based on the backing node where the snapshot w
as taken
    a) Creating a new command which drops the current active layer
    b) Recreating the block devices instead of adding a new command

2) Checking whether devices are still reading or even writing to RAM
```

Note that (2) is also a problem with the current method.

#16 - 2018-03-08 08:42 - rpalethorpe

Loading a VM snapshot/migration into a QEMU instance which has already ran a VM appears to work, but I am concerned about the following:

1) There is nothing stopping devices from reading or writing memory during the migration if they do not follow the memory or bus APIs.
2) Many devices hold state outside of guest RAM and it is down to the device to correctly load new state. It is not clear if all devices overwrite existing state correctly.
3) It is not clear that the vCPUs are all guaranteed to have stopped by the time the migration starts.

I have not found any instances of (1) or (2), but QEMU has a large number of complex devices so auditing it would be difficult. It seems that 'loadvm' usually works, but we can not be certain that all devices have a consistent state. If an error in the guest kernel is caused by a device with inconsistent state it may be almost impossible to determine what caused the bug. So while I am sure QEMU can be patched to allow incoming migrations without '-defer' and it would work most of the time (and be quite useful for some people), we really should restart QEMU before loading a VM.

#17 - 2018-03-12 09:08 - rpalethorpe

- Status changed from In Progress to Workable

#18 - 2018-04-24 09:24 - rpalethorpe

- Due date set to 2018-04-24

- Start date changed from 2018-03-09 to 2018-04-24

due to changes in a related task

#19 - 2018-04-26 09:27 - rpalethorpe

- Target version deleted (Milestone 15)

#20 - 2018-08-02 08:22 - rpalethorpe

- Status changed from Workable to In Progress

The good news is that after the deployment of the QEMU rewrite and various bug fixes for integration issues with the rest of the OpenQA framework, snapshots on ARM now appear to be reliable. The bad news is they are still slow in comparison with X86 and ppc64le. The snapshot timeout of 240 seconds is too short for machines with 4GB and frankly 240 seconds is way too long to begin with. For now, I think ARM machines with >2GB of RAM should just have snapshots disabled or we just disable them for all machines except the virtio variant used mostly with console tests.

The reason for the slowness is not clear, possibly ARM struggles with the compression, although it is very mild compression. Another possibility is that there is some bottleneck in a bus or other transport. As far as QEMU and OpenQA is concerned; we are now on the happy path, so I doubt it is anything specific to OpenQA.

#21 - 2018-08-07 11:15 - rpalethorpe

- Status changed from In Progress to Resolved