An Update (and Demo) on Techniques to Manage libvirt/QEMU-based Virtual Machine Snapshots and Disk Image Chains

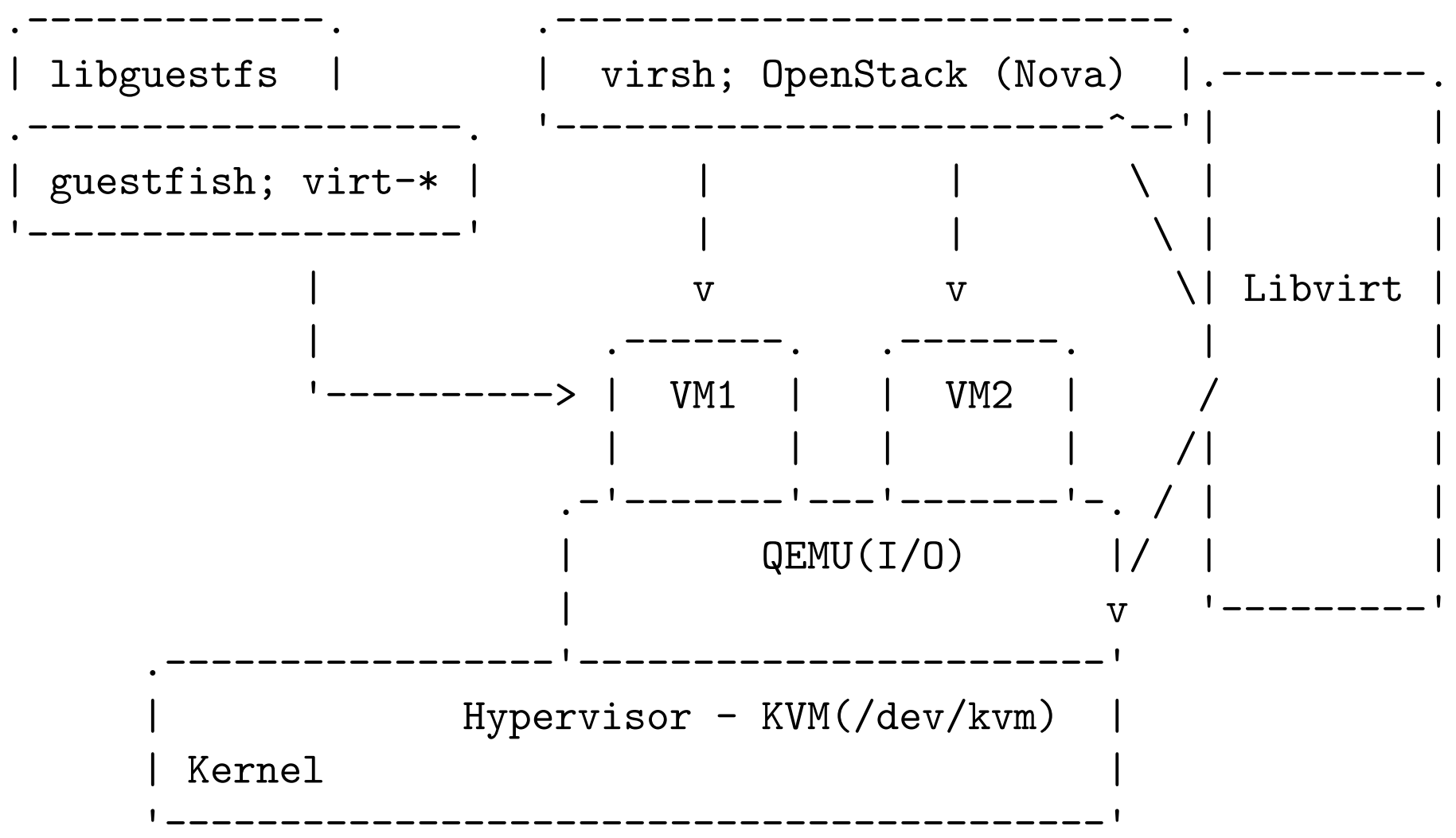
Kashyap Chamarthy <kashyap@redhat.com>, Düsseldorf

15 OCT 2014

Section 1

Background

KVM virtualization stack



Overview

- libvirt: Hypervisor agnostic virtualization library. Manage: virtual machines, disks, devices, networks
- QEMU: Emulator (CPU, devices, networks, etc). Interacts with libvirt via QEMU Machine Protocol (QMP)
  - e.g. live block operations
- Default virtualization drivers in higher-layer projects, e.g. OpenStack.

New in libvirt/QEMU for disk image mangement

- Improved external snapshot creation: with disk & memory
- Efficient live disk mirroring with **blockcopy**
- Disk image shortening – *much* faster with **blockcommit**
- Efficient Live disk migration (with **blockcopy**+**blockcommit**)
- **virsh** interface & block API enhancements, bug fixes

redhat.

Section 2

Snapshots and disk image management utilities

Disk image management: QEMU operations

`qemu-img`	Summary
snapshot	Create/manage offline disk snapshots
create	Create offline external snapshots
commit	Commit changes from 'top' to 'base'
rebase	Copy contents from 'base' to 'top'; Fix/adjust broken backing files

* Refer to the qemu-img(1) man page. But, highly recommended: libvirt or higher layer tools

Disk image management: libvirt operations (1)

`virsh`	Summary
snapshot-create-as	Internal/external snapshots
blockcopy	Copy disk image chain to destination
blockcommit	Merge files from 'top' into 'base'
blockpull	Merge files from 'base' into 'top'

```
[orig] <-- [sn1] <-- [cur] (live QEMU)
           (base)    (top)
```

Disk image management: libvirt operations (2)

* Under the hood, QEMU QMP commands are used for live block opertaions

  - To see libvirt <-> QEMU interactions, enable logging filters in /etc/libvirt/libvirtd:

    log_filters="1:qemu_monitor"
    log_outputs="1:file:/var/tmp/libvirtd.log"
    (Restart libvirt: `systemctl restart libvirtd`)

* Refer to the virsh(1) man page to explore all the powerful controls

External disk snapshots - extremely useful, but:

* PROBLEM with long disk image chains:

  [a] <- [sn1] <- [sn2] <- [cur] (live QEMU)

  - Cumbersome to mantain
  - Degrading performance

* SOLUTION:

  - Shorten disk image chains, _without_ guest down time

Section 3

Examples and demonstration

Merge (live) disk chain into base image (1)

Begin with: [base] (live QEMU)

Create external live disk snapshot:

```
$ virsh snapshot-create-as \
   --domain vm1 sn1 \
   --diskspec vda,file=/export/images/sn1.qcow2 \
   --disk-only --atomic --no-metadata
```

Repeat it, to have a chain like:

```
[base] <-- [sn1] <-- [sn2] <-- [cur]
                               (live QEMU)
```

Merge (live) disk chain into base image (2)

Perform live blockcommit:

```
$ virsh blockcommit vm1 vda \
    --active \
    --pivot  \
    --verbose
```

* Two stage operation:
  1. Commits content from top to base
  2. top & base remain in sync; live QEMU is pivoted to base image (with --pivot)

Merge (live) disk chain into base image (3)

Final result, this chain:

```
    [base] <-- [sn1] <-- [sn2] <-- [cur]
                                   (live QEMU)
```

is shortened to:

```
    [base] (live QEMU)
```

where data from 'sn1', 'sn2' and 'cur' are live committed into 'base'.

Merge (live) disk chain into base image (4)

```
* Related notes:

  - 'cur' is valid _until_ the pivot of live
    QEMU completes

  - 'sn1' and 'sn2' are no longer valid in
    isloation

  - When the pivot completes, 'cur' is also
    no longer valid

  i.e. it invalidates intermediate images
```

Efficient disk backup using active blockcommit (1)

Starting with a single disk as below:

```
[base] (live QEMU)
```

Create a temporary extenal live disk snapshot:

```
$ virsh snapshot-create-as [. . .]
```

So, it results in:

```
[base] <-- [sn1] (live QEMU)
```

Efficient disk backup using active blockcommit (2)

Take a backup of the base image:

```
$ rsync -avh --progres base.qcow2 /dst/copy.qcow2
```

Now, we have:

```
[base] <-- [snap1] (live QEMU)
  |
  |
[copy]
```

Efficient disk backup using active blockcommit (3)

Undo the external disk snapshot via active commit:

```
$ virsh blockcommit f20-orig vda \
    --active  \
    --verbose \
    --pivot
```

Finally, the chain is back to a single disk:

```
[base] (live QEMU)
```

More examples, here:

https://kashyapc.fedorapeople.org/virt/lcco-2014/examples/

* Live disk migration using blockcopy+blockcommit

* Live disk backup while reusing _existing_ destination
  - Lets you control the _type_ of backing disk and image chain depth

* And, a few more. . .

OpenStack and further. . .

- OpenStack Nova's libvirt driver: Currently uses **blockRebase**, **blockCommit**, **blockJobInfo** APIs. ('nova image-create –poll vm1 sn1')
- More work underway to take advantage of the newer libvirt & QEMU improvements
- libvirt is incrementally catching up to expose newer features QEMU has to offer (e.g. drive-backup, etc)

redhat.

Section 4

Notes, references

References

- Demo and related notes
  `https://kashyapc.fedorapeople.org/virt/lcco-2014/`
- From libvirt wiki
  `http://wiki.libvirt.org/page/I_created_an_external_snapshot,_but_libvirt_won%27t_let_me_delete_or_revert_to_it`
- Related notes from 2012 CloudOpen Eu
  `https://kashyapc.fedorapeople.org/virt/lc-2012/`
- Lots of info on upstream libvirt-users list archives
  `https://www.redhat.com/archives/libvirt-users/`
- Blog:
  `http://kashyapc.com`

The end.

Thanks for listening.