Linux/QEMU/Libvirt

4 Years in the Trenches

Chet Burgess
Cisco Systems
Scale 14x Sunday January 24th

Introduction

What do I know?

I’ve spent the last 4 years designing, building, and managing OpenStack based clouds. I’ve seen millions of unique VMs running on QEMU.

What I am going to talk about?

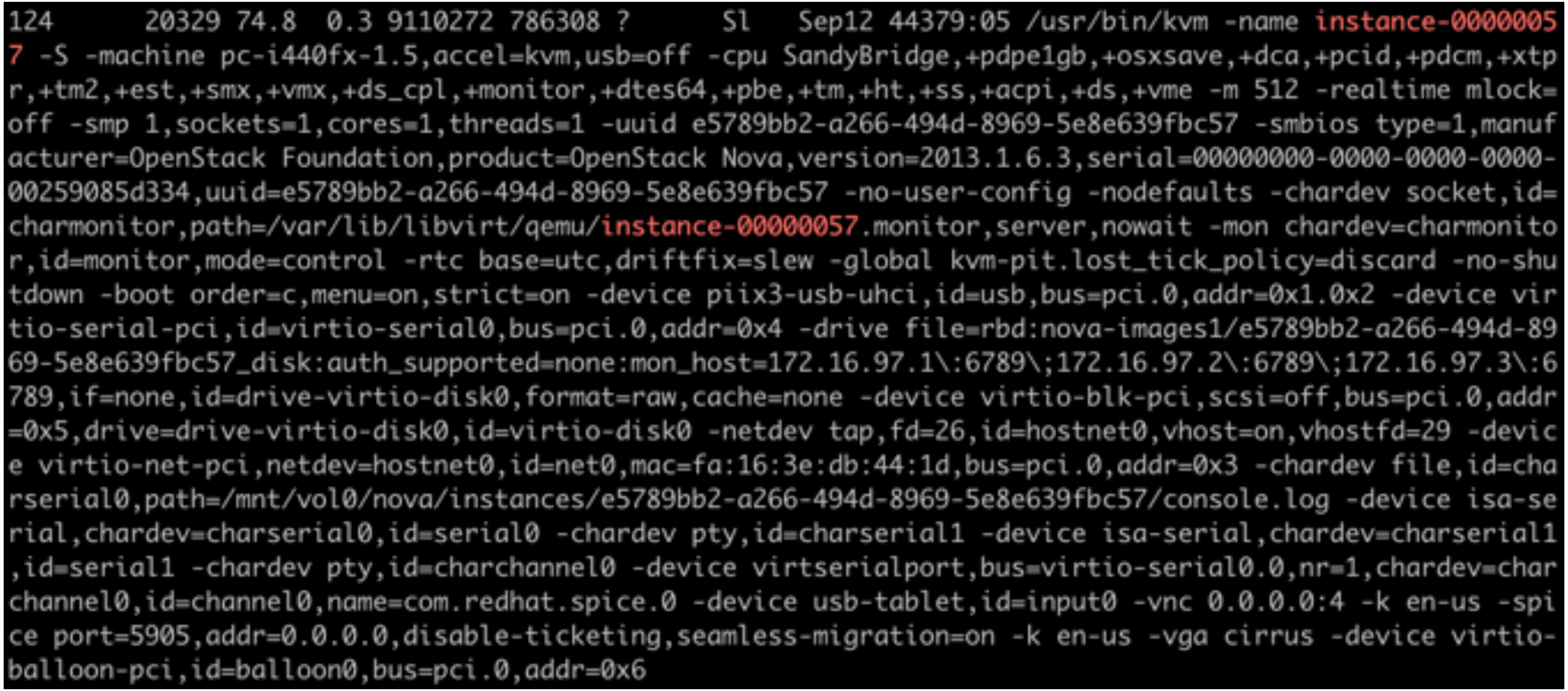
I’m going to share some interesting tips and trips we've learned over the years. I’m not covering the basics of libvirt and QEMU.

Building Blocks

Libvirt & QEMU

- QEMU is the emulation layer
- Libvirt is a tool for controlling QEMU
  - Provides local API http://tinyurl.com/libvirt-api-ref)
  - Provides command line interface (http://tinyurl.com/virsh-doc)
  - Supports XML configuration format (http://tinyurl.com/libvirt-xml-doc)

Libvirt saves your sanity!



```
124       20329 74.8  0.3 9110272 786308 ?      Sl   Sep12 44379:05 /usr/bin/kvm -name instance-0000005
7 -S -machine pc-i440fx-1.5,accel=kvm,usb=off -cpu SandyBridge,+pdpe1gb,+osxsave,+dca,+pcid,+pdcm,+xtp
r,+tm2,+est,+smx,+vmx,+ds_cpl,+monitor,+dtes64,+pbe,+tm,+ht,+ss,+acpi,+ds,+vme -m 512 -realtime mlock=
off -smp 1,sockets=1,cores=1,threads=1 -uuid e5789bb2-a266-494d-8969-5e8e639fbc57 -smbios type=1,manuf
acturer=OpenStack Foundation,product=OpenStack Nova,version=2013.1.6.3,serial=00000000-0000-0000-0000-
00259085d334,uuid=e5789bb2-a266-494d-8969-5e8e639fbc57 -no-user-config -nodefaults -chardev socket,id=
charmonitor,path=/var/lib/libvirt/qemu/instance-00000057.monitor,server,nowait -mon chardev=charmonito
r,id=monitor,mode=control -rtc base=utc,driftfix=slew -global kvm-pit.lost_tick_policy=discard -no-shu
tdown -boot order=c,menu=on,strict=on -device piix3-usb-uhci,id=usb,bus=pci.0,addr=0x1.0x2 -device vir
tio-serial-pci,id=virtio-serial0,bus=pci.0,addr=0x4 -drive file=rbd:nova-images1/e5789bb2-a266-494d-89
69-5e8e639fbc57_disk:auth_supported=none:mon_host=172.16.97.1\:6789\;172.16.97.2\:6789\;172.16.97.3\:6
789,if=none,id=drive-virtio-disk0,format=raw,cache=none -device virtio-blk-pci,scsi=off,bus=pci.0,addr
=0x5,drive=drive-virtio-disk0,id=virtio-disk0 -netdev tap,fd=26,id=hostnet0,vhost=on,vhostfd=29 -devic
e virtio-net-pci,netdev=hostnet0,id=net0,mac=fa:16:3e:db:44:1d,bus=pci.0,addr=0x3 -chardev file,id=cha
rserial0,path=/mnt/vol0/nova/instances/e5789bb2-a266-494d-8969-5e8e639fbc57/console.log -device isa-se
rial,chardev=charserial0,id=serial0 -chardev pty,id=charserial1 -device isa-serial,chardev=charserial1
,id=serial1 -chardev pty,id=charchannel0 -device virtserialport,bus=virtio-serial0.0,nr=1,chardev=char
channel0,id=channel0,name=com.redhat.spice.0 -device usb-tablet,id=input0 -vnc 0.0.0.0:4 -k en-us -spi
ce port=5905,addr=0.0.0.0,disable-ticketing,seamless-migration=on -k en-us -vga cirrus -device virtio-
balloon-pci,id=balloon0,bus=pci.0,addr=0x6
```

Machine Type

- Machine type defines the characteristics of the hardware that will be presented (http://tinyurl.com/qemu-machine-type)
  - USB bus, PCI bus, available types of NIC cards, video card, etc
- qemu_x86-64 -machine help
- Machine types are passed by name
  - example: -machine pc-i440fx-rhel7.1.0,accel=kvm,usb=off
- You cannot change the machine type once a VM is booted

CPU Models

- CPU Models define CPU architecture and flags
  - QEMU (http://tinyurl.com/qemu-cpu-model)
  - libvirt (http://tinyurl.com/libvirt-cpu-model)
- qemu_x86-64 -cpu help
- QEMU supports “host” model (pass all available flags of the physical CPU that are supported)
- Libvirt supports “passthrough” model (lists each flag on the command line of the physical CPU that are supported
- Some flags must be emulated

Storage

Storage Backend Considerations

- Understand your workload and how your storage backend works
- Do NOT forget about IOPS!
  - Add more spindles to increase your available IOPS
  - Consider using SSDs as cache (bcache, dm-cache, CEPH journals and monitors)
  - Be careful trading IOPS for more storage (compression, de-duplication)
- Tiered storage
  - Consider a build storage tier (spinning drives) and a high performance tier (SSDs)

Disk Errors

- What happens when QEMU can’t read/write to the device?
- Configurable via error_policy and rerror_policy in XML
- Values
  - report (default) - Send the error from the underlying storage subsystem to the guest kernel
  - stop - pause the VM instead of reporting the error
  - ignore - Error? What error?
  - enospace - Send enospace error to the guest kernel

Disk Cache Mode

- Configures disk caching mode QEMU will use for I/O
- Values
  - none, writethrough (default), writeback, directsync, unsafe
- Enabling
  - Set cache='$VALUE' in driver definition in XML
- Detailed explanation of each at http://tinyurl.com/libvirt-cache

UNMAP/TRIM Support

- UNMAP will purge data from some disk formats and device types
  - QCOW2, RBD, some iSCSi backends
- Requirements
  - Guest Kernel Support
  - QEMU 1.5.0+
  - Libvirt 1.0.6+
  - virtio-scsi bus type
- Enabling
  - Add discard='unmap' to driver definition in XML

Libvirt XML for Disk Device

```
<disk type='file' device='disk'>
   <driver name='qemu'
           type='qcow2'
           cache='none'
           discard='unmap'
           error_policy='stop'/>
   <source file='/mnt/vm/discard/disk'/>
   <target dev='sda' bus='scsi'/>
   <alias name='scsi0-0-0-0'/>
   <address type='drive' controller='0' bus='0' target='0' unit='0'/>
</disk>
```

VM Migrations

Why Migrations Matter

- Operations
  - Key to performing non-disruptive work
  - Re-balancing workloads and resources
- Expectations versus reality
  - Special snowflakes
  - Ephemeral instances and the “cloud way”

Migrations

- “Cold” Migrations
  - Shutdown the VM, copy the data and XML, start the VM
- Live Migrations
  - Copy machine (CPU & RAM) from source to destination with **minimal** impact, implies some form of “shared” storage
- Live Block Migrations
  - Also copies the disk files of the running machine to the destination, implies “local” storage
- http://tinyurl.com/libvirt-migrate

Tips for Successful Live Migrations

- virsh migrate
  - Pause VM and migrate to new destination
  - --live flag to minimize pause time
    - Still pauses for final incremental sync of CPU and RAM
    - Impacted by high memory churn rate (JVMs)
    - Use virsh migrate-setmaxdowntime to control length of pause
    - --timeout controls how long to try before pausing and doing a full sync
  - File based disk paths cannot be changed unless you use --xml

Tips for Successful Live Block Migrations

- virsh migrate --copy-storage-all
  - Copy full content of disk to destination
  - Flattens disk on copy
- virsh migrate --copy-storage-inc
  - Doesn’t flatten disk
- Any file based disk device will be copied to destination
  - No safety check to see if the file is shared

Machine, CPU, and Live Migrations

- Machine type must be identical on source and destination during migrations
  - Since its passed by name it means the name and the actual definition must match
- CPU Model and flags also need to be identical
  - Challenges arise with heterogeneous hardware environments
  - Pick the smallest and simplest set of flag needed to ensure maximum capability

Disk Cache Mode and Live Migrations

- Libvirt will deny live migrations if cache != ‘none’
- Except……
  - RBD has special handling in libvirt. As long as the cache type is set to ‘writeback’ libvirt will allow the migration.

Upgrades

Upgrading

- In theory newer versions of QEMU are backward compatible; in theory.
    - Issues may arise attempting to boot or live migrate a VM to a newer QEMU with an older machine type
    - Consider having multiple versions installed and using a wrapper
- If you don't include a machine type in your XML you will get the latest
    - Good - Just reboot your VM to upgrade it
    - Bad - If your OS/application is machine/CPU sensitive a reboot could break your VM.
- General Rule - migrate from older to newer versions (QEMU, libvirt, Kernel)

Libvirt Potpourri

Libvirt Tunables

- Improves scalability for programatic clients
  - max_clients = 50
  - prio_workers = 25
  - min_workers = 5
  - max_workers = 50
  - max_client_requests = 25
- Libvirt UUID
  - Some distros ship with the libvirt UUID set to all 0’s in the config file
  - Be sure its unique or that ‘dmidecode -s system-uuid’ returns a unique value

Libvirt XML

- Setting smbios
  - Used by some licensing schemes to “fingerprint” the hardware

```
<sysinfo type='smbios'>
  <system>
    <entry name='manufacturer'>OpenStack Foundation</entry>
    <entry name='product'>OpenStack Nova</entry>
    <entry name=‘version'>7.0</entry>
    <entry name='serial'>12345</entry>
    <entry name='uuid'>17417240-7f62-4a30-8821-c86ef0e9bf6f</entry>
  </system>
</sysinfo>
```

Q&A

You’ve got questions?

I’ve got answers.

Maybe.

cisco