See What Happened With Real-Time KVM When Building Real Time Cloud?

Pei Zhang (张 培)
pezhang@redhat.com
June 20, 2017

Agenda

1. Background
2. Build real time cloud
3. Configure Open vSwitch, DPDK and vhost-user in real time environment
4. Show performance testing results

1. Background

1.1 Real-Time

Real-Time is about determinism, constant and predictive behaviour.

1.2 Network Function Virtualization(NFV)

Network functions will be implemented in software that can run on a range of industry standard server hardware.

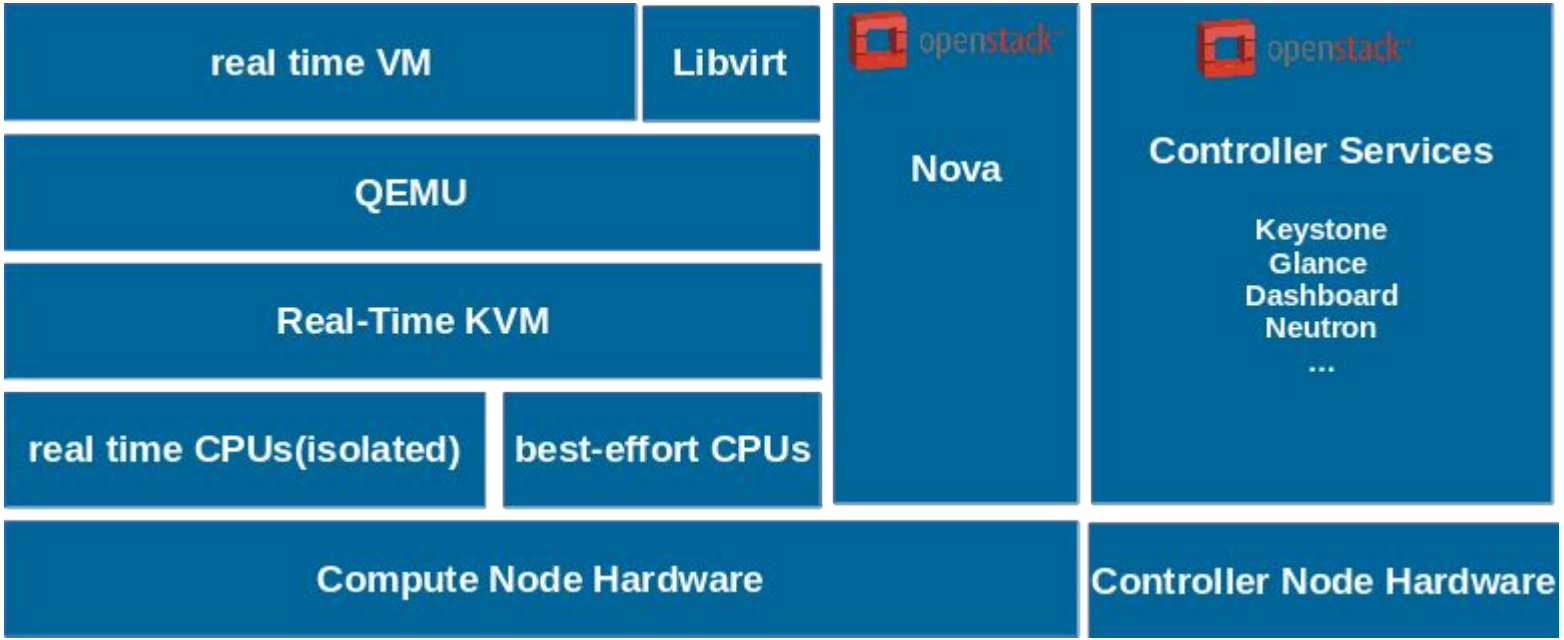
Dedicated network appliances will be replaced by virtualization and software.

1.3 FCAPS in Telecommunications

- **Fault management**
- **Configuration management**
- **Accounting management**
- **Performance management**
  - Throughput, **network response times**, packet loss rates, etc
- **Security management**

2. Build real time cloud

2.1 Real-Time in OpenStack



2.2 Strict pre-requisites for compute nodes

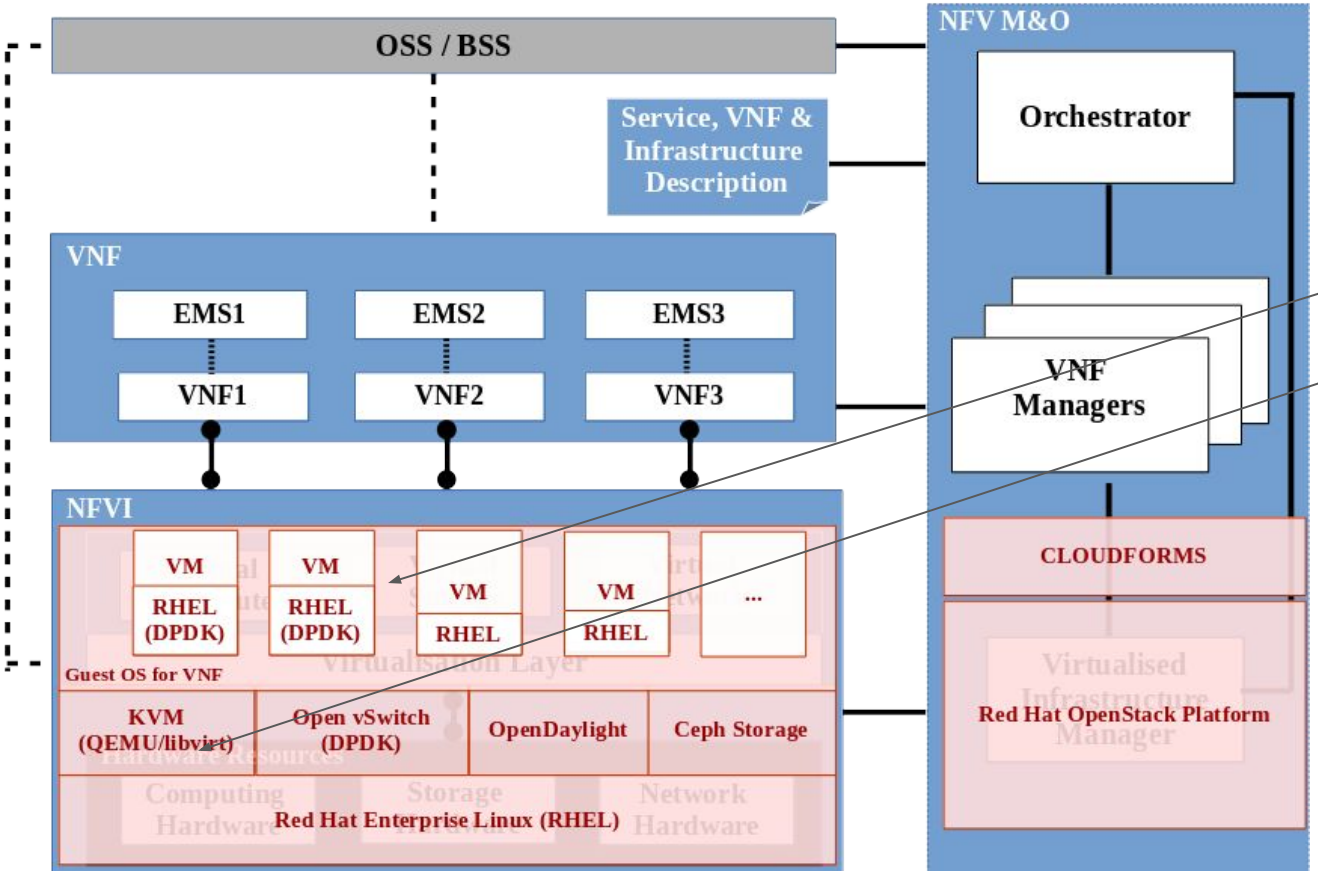
- Set up BIOS
- Install real time kernel/kvm
- Isolate cores
- Set hugepages

2.3 Configure Nova for launching real time VM

- Real time policy
- Pin vCPUs to individual cores
- Set housekeeping cores and real time cores
- Reserve hugepages
- Avoid devices which cause high latency

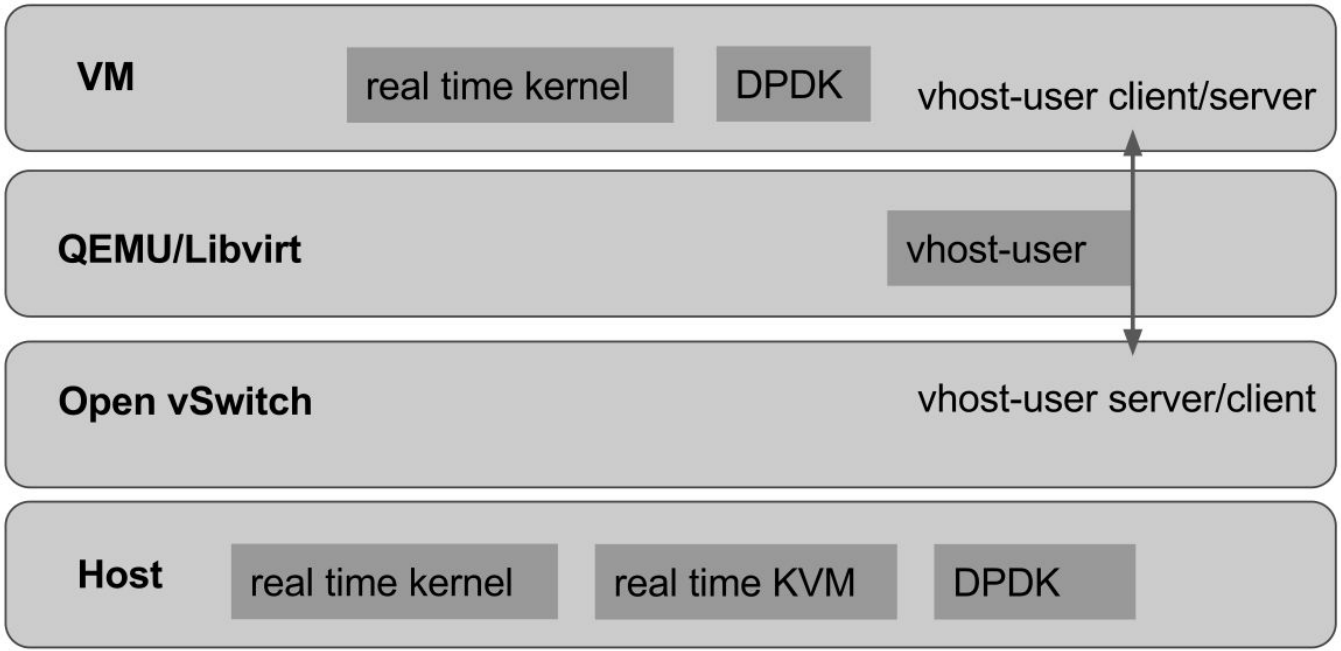
3. Configure Open vSwitch, DPDK and vhost-user in real time environment

3.1 NFV Solution



- Real-Time OS in the VNF
- Real-Time KVM in the NFVI

3.2 VM with Open vSwitch, DPDK and vhost-user



- Poll mode driver
- Hugepage
- User space
- Strict NUMA policy
- Real time scheduling
- Individual cores pinning

3.2 Single queue topology

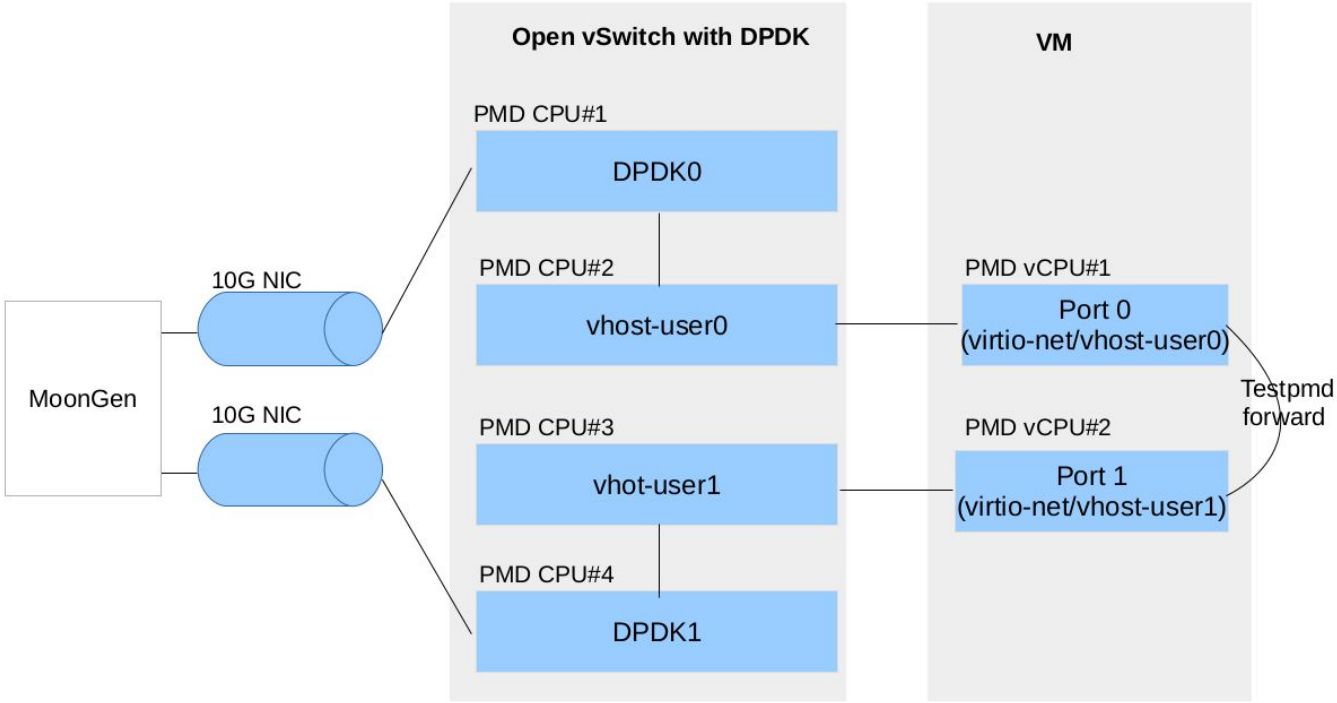


Fig. vhost-user single queue

3.3 Multiple queues topology

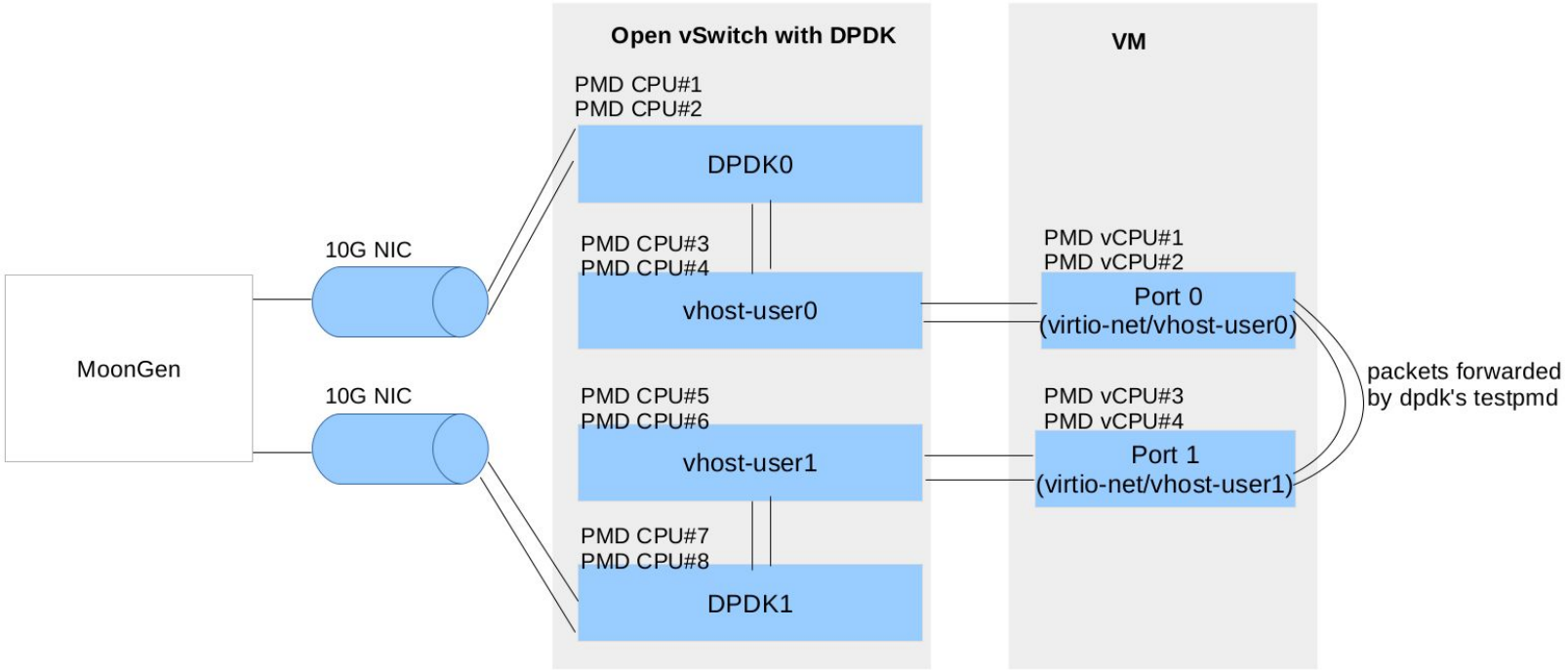


Fig. vhost-user 2 queues

4. Show performance testing results

4.1 Real-Time KVM latency testing

- **Multiple vCPUs in VM**

	vCPU 1	vCPU 2	vCPU 3	vCPU 4	vCPU 5	vCPU 6	vCPU 7	vCPU 8
Min(us)	00006	00006	00006	00006	00006	00005	00006	00006
Avg(us)	00006	00006	00006	00006	00006	00006	00006	00006
Max(us)	**00012**	**00012**	**00012**	**00011**	**00016**	**00012**	**00012**	**00017**

Testing environment:
CPU: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz

Versions: Red Hat Enterprise Linux 7

4.1 Real-Time KVM latency testing

- **Multiple VMs**

	VM 1	VM 2	VM 3	VM 4
Min(us)	00005	00005	00005	00005
Avg(us)	00006	00006	00006	00006
Max(us)	**00012**	**00011**	**00013**	**00013**

Testing environment:
CPU: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz

Versions: Red Hat Enterprise Linux 7

4.2 Network layer 2 latency testing

1 hour	Min(us)	Avg(us)	Max(us)	v_95p	v_99p	v_9999p
KVM / 2Q	15.987	17.077	777.990	17.555	18.547	23.692
RT-KVM /2Q	15.916	16.429	39.084	17.004	17.171	17.683

Testing environment:
CPU: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz
NIC: 10-Gigabit X540-AT2

Versions: Red Hat Enterprise Linux 7

Thanks.

Appendix

- Configuring compute node

 # cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-661.rt56.579.el7.x86_64 ...default_hugepagesz=1G iommu=pt intel_iommu=on isolcpus=1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,30,28,26,24,22,20,18,16 nohz=on nohz_full=1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,30,28,26,24,22,20,18,16 rcu_nocbs=1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,30,28,26,24,22,20,18,16 intel_pstate=disable nosoftlockup

 # tuned-adm active
Current active profile: realtime-virtual-host

 # cat /usr/lib/tuned/realtime-virtual-host/lapic_timer_adv_ns
3500
 # cat /sys/module/kvm/parameters/lapic_timer_advance_ns
3500

 # cat /sys/devices/system/node/node0/hugepages/hugepages-1048576kB/nr_hugepages
20

- configuring nova

[root@compute1 ~]**# cat /etc/nova/nova.conf**
vcpu_pin_set=2,4,6,8,10,12,14,16,18
…

[root@compute1 ~]**# source admin-openrc**
[root@compute1 ~]**# nova flavor-key 1 set hw:cpu_policy=dedicated**
[root@compute1 ~]**# nova flavor-key 1 set hw:cpu_realtime=yes**
[root@compute1 ~]**# nova flavor-key 1 set hw:cpu_realtime_mask="^0-1"**
[root@compute1 ~]**# nova flavor-key 1 set hw:mem_page_size=1GB**

- booting VM

```
<memoryBacking>
      <hugepages>
      <page size='1048576' unit='KiB' nodeset='0'/>
      </hugepages>
      <locked/>
</memoryBacking>
<vcpu placement='static'>2</vcpu>
<cputune>
      <vcpupin vcpu='0' cpuset='19'/>
      <vcpupin vcpu='1' cpuset='18'/>
      <emulatorpin cpuset='1,3,5,7,9'/>
      <vcpusched vcpus='0-1' scheduler='fifo' priority='1'/>
</cputune>
<cpu mode='host-passthrough'>
      <feature policy='require' name='tsc-deadline'/>
</cpu>
```

Hugepage

vCPU pin

Real time

Note:

Remove devices which will cause high latency:

1. All USB support
2. All Spice support and QXL hardware
3. Virtio serial and guest agent sockets
4. Sound card

- Configuring VM

 # cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-661.rt56.579.el7.x86_64 root=/dev/mapper/rhel-root ro crashkernel=auto rd.lvm.lv=rhel/root rd.lvm.lv=rhel/swap rhgb quiet LANG=en_US.UTF-8 console=tty0 console=ttyS0,115200 isolcpus=1 nohz=on nohz_full=1 rcu_nocbs=1 intel_pstate=disable nosoftlockup

 # tuned-adm active
Current active profile: realtime-virtual-guest

- Set up of Open vSwitch

1. Bind network devices to the driver dpdk supported, like vfio-pci
2. Open DPDK support option
   # ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-init=true
3. Use hugepage
   # ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-socket-mem="1024,1024"
4. Create DPDK ports and vhost-user ports
   #ovs-vsctl add-port ovsbr1 dpdk1 -- set Interface dpdk1 type=dpdk
   # ovs-vsctl add-port ovsbr1 vhost-user1 -- set Interface vhost-user1 type=dpdkvhostuserclient options:vhost-server-path=/tmp/vhostuser1.sock
   # ovs-vsctl add-port ovsbr0 vhost-user0 -- set Interface vhost-user0 type=dpdk vhostuser
5. Pin each PMD thread to individual isolate core
   Eg. 4 port, 4 pmd thread, 4 individual cores.
6. Follow strict NUMA placement policy
   Should use cores and hugepage of same NUMA with network devices locate
7. Change the real time scheduling of all PMD processes.
   # chrt -fp 95 $processID

- Set up of VM

1. Follow strict NUMA placement policy

   Should use cores and hugepage of same NUMA with host network devices(or Open vSwitch)

2. Boot with enough vCPUs

3. Add network devices with vhost-user client/server mode

   If Open vSwitch vhost-user ports is client mode, then guest should be server mode. Vice visa.