Features & Futures: Red Hat Enterprise Virtualization Hypervisor (KVM)

Karen Noel – Senior Software Engineering Manager
Andrea Arcangeli – Sr. Principal Software Engineer
June 2016
Features & Futures:
• Red Hat's KVM Hypervisor
• Virtual CPU and memory hot-plug
• Real-time KVM
• Post-copy live migration
• More futures...
Red Hat Virtualization - KVM
Cloud Infrastructure for Cloud Workloads

IaaS+

Monitoring
Ceilometer

Data Processing
Sahara

Orchestration
Heat

IaaS

Compute
Nova

Storage
Cinder, Glance, Swift

Networking
Neutron

Bare Metal Provisioning
Ironic

Shared Services
Identity
Keystone

Deployment and Management
Tripleo (Director)

#redhat #rhsummit
# RHEV and Red Hat OpenStack Platform

<table>
<thead>
<tr>
<th>TRADITIONAL: SCALE UP (RHEV)</th>
<th>CLOUD: SCALE OUT (OpenStack)</th>
<th>MIXED/HYBRID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big stateful VM</td>
<td>Small stateless VMs</td>
<td>Combination of traditional scale-up and cloud scale-out workloads.</td>
</tr>
<tr>
<td>1 Application → 1 VM</td>
<td>1 Application → Many VMs</td>
<td>For example: Database may be hosted on traditional workloads, web front-end and logic layers on cloud workloads.</td>
</tr>
<tr>
<td>Lifecycle in years</td>
<td>Lifecycle hours to months</td>
<td></td>
</tr>
<tr>
<td>Increased user demand = Scale up (VM gets bigger)</td>
<td>Increased user demand = Scale out (add VMs)</td>
<td></td>
</tr>
<tr>
<td>Not designed to tolerate failure of VM, so you need features that keep VMs up</td>
<td>If a VM dies, application kills it and creates a new one, app stays up</td>
<td></td>
</tr>
<tr>
<td>Application SLA requires enterprise virtualization features (migration, HA, etc.) to keep applications available</td>
<td>Application SLA requires adding/removing VM instances to application cloud to maintain application availability</td>
<td></td>
</tr>
</tbody>
</table>
KVM Virtualization Stack - host

- libvirt
- QEMU
- KVM
- Red Hat Enterprise Linux

- virt-manager
- RHEV vds
- virsh cli
- openstack-nova

VFIO vhost
KVM Virtualization Stack - guest

QEMU process

qemu-ga

virtio drivers
Red Hat Enterprise Linux

Serial port to host

QEMU process

qemu-guest-agent-win

virtio-win drivers
Microsoft Windows

Serial port to host
Who is running RHEL? 
RHEV? 
OpenStack? 
VMware?
1. Connect to vSphere

2: Select VMs to Migrate

https://www.youtube.com/watch?v=yEjckA-FriU
3: Configure RHEV VM

<table>
<thead>
<tr>
<th>Name</th>
<th>Origin</th>
<th>Memory</th>
<th>CPUs</th>
<th>Architecture</th>
<th>Disks</th>
</tr>
</thead>
<tbody>
<tr>
<td>RHEL7_18</td>
<td>VMware</td>
<td>2048 MB</td>
<td>1</td>
<td>x86_64</td>
<td>1</td>
</tr>
</tbody>
</table>

**General**
- **Name**: RHEL7_18
- **Physical Memory**: 2048 MB
- **CPU**: 1
- **Architecture**: x86_64
- **Guest OS Memory**: Not Configured
- **Guest OS Memory PreCache**: Not Configured
- **Allocated**: Not Configured
- **Additional**:
  - **Number of CPUs**: 1
  - **Description**: Not Configured
  - **Template**: Not Configured
  - **Guest CPU Count**: Not Configured
  - **Guest CPU Count 2**: Not Configured

**Network Interfaces**
- **Name**: RHEL7_18
- **Operating System**: Red Hat Enterprise Linux
- **Guest OS Memory PreCache**: Not Configured
- **Cluster**: Not Configured
- **Busy**: Not Configured
- **Failed**: Not Configured
- **UUID**: Not Configured
- **Template**: Not Configured
- **Guest CPU Count**: Not Configured
- **Guest CPU Count 2**: Not Configured

**Disks**
- **Name**: RHEL7_18
- **Operating System**: Red Hat Enterprise Linux
- **Guest OS Memory PreCache**: Not Configured
- **Cluster**: Not Configured
- **Busy**: Not Configured
- **Failed**: Not Configured
- **UUID**: Not Configured
- **Template**: Not Configured
- **Guest CPU Count**: Not Configured
- **Guest CPU Count 2**: Not Configured
4: Completion
http://libguestfs.org/virt-v2v.1.html#convert-from-vmware-vcenter-server-to-local-libvirt
From VMware to KVM...

**RHEV:**

```
# virt-v2v -ic vpx://vcenter.example.com/Datacenter/esxi vmware_guest \
  -o rhev -os rhev.nfs:/export_domain --network rhevm
```

**RHOSP - Glance:**

```
# virt-v2v -i disk disk.img -o glance
```

**RHEL:**

```
# virt-v2v -ic vpx://vcenter.example.com/Datacenter/esxi vmware_guest
```
How does V2V work? Windows Guest

- Check for Group Policy Objects → WARNING!
- Check for Anti-Virus → WARNING!
- Insert RHEV guest agent - add firstboot script to install
- Disable Windows services, intelppm.sys, processor.sys
- Disable autoreboot... just in case
- Upload virtio drivers
  - Modify HKLM\SOFTWARE registry, locate virtio-blk at boot
  - Other drivers – use PCI/driver discovery
Windows Guest – v2v support

- RHEL 7.3 adds Win8+
  - Tech preview? Supported?
  - New Windows driver installation model
  - Being very cautious, needs lots of testing ← volunteer!
How does V2V work? RHEL Guest

- Clean RPM database
- Check kernels available: virtio supported? Which drivers?
- Touch /.autorelabel – SELinux will relabel file system on next boot
- /etc/X11/xorg.conf – change to use QXL or Cirrus
- Rebuild initrd – so virtio drivers are available
Virtual CPU and memory hot-plug
Memory size increase

OK

Apply immediately
Memory hot-plug

- Configure guest with:
  - `maxMemory > currentMemory`
  - NUMA topology
    - `node0 is enough`

```
<maxMemory slots='16' unit='KiB'>4194304</maxMemory>

<cpu>
  <numa>
    <cell id='0' cpus='0-3' memory='524288' unit='KiB'/>
  </numa>
</cpu>
```
Memory hot-plug

- Prepare a device memory.xml to attach with:

```bash
# virsh attach-device memory.xml
```

```xml
<memory model='dimm'>
  <target>
    <size unit='KiB'>524288</size>
    <node>0</node>
  </target>
</memory>
```
Memory hot-plug

- Memory unplug not supported, yet*
- Use balloon to adjust guest memory down/up

* Currently targeted for a future release of RHEV
Virtual CPU hot-plug

- Configure guest with max vcpus > current vcpus

```xml
<domain type='kvm'>
  <vcpu placement='static' current='2'>16</vcpu>
</domain>
```
Virtual CPU hot-plug

- Specify total number of vcpus for guest
  - Unplug is not supported yet*
- RHEL: udev rule brings cpus online
- Fedora: use –guest or add udev rule
  - Configure QEMU guest agent

# virsh setvcpus fedora-24 4
# virsh setvcpus fedora-24 4 --guest
RHEL for Real-Time with KVM
Red Hat OpenStack Platform 8*

* KVM-RT is Tech Preview in RHOSP 8
Real-time KVM

- 1/3 Code
- 1/3 Tuning & Config
- 1/3 Apps & Ops

Collaboration:
- Open source
- HW partners
- NFV partners
KVM-RT - code

- 56% Red Hat contribution as of June 2015 (48 of 86)
- 58% Red Hat contribution as of June 2016 (63 of 109)
- Kernel: mm, core, vmstat, sched, memcontrol, workqueue, timer, cpusets/isolcpus, lib/vsprintf, rcu/nohz/kvm, tracing, kvm/x86
- Ftrace: x86
- RT kernel: rt/kvm
- Libvirt: qemu
### KVM-RT – tuning/config host

```
[root@localhost ~]# hwlatdetect
hwlatdetect: test duration 120 seconds
parameters:
  Latency threshold: 10us
  Sample window: 1000000us
  Sample width: 500000us
  Non-sampling period: 500000us
  Output File: None

Starting test
  test finished
  Max Latency: 11us
  Samples recorded: 6
  Samples exceeding threshold: 7
  1466464193.0724182328  0   11
  1466464211.0725493769  11  0
  1466464234.0724915467  0   11
  1466464276.0725694788  11  0
  1466464280.0725769009  0   11
  1466464286.0724895787  11  0
```

```
[root@localhost ~]# hwlatdetect
hwlatdetect: test duration 120 seconds
parameters:
  Latency threshold: 10us
  Sample window: 1000000us
  Sample width: 500000us
  Non-sampling period: 500000us
  Output File: None

Starting test
  test finished
  Max Latency: 0us
  Samples recorded: 0
  Samples exceeding threshold: 0
```

* [https://rt.wiki.kernel.org/index.php/HOWTO:_Build_an_RT-application](https://rt.wiki.kernel.org/index.php/HOWTO:_Build_an_RT-application)
* [https://access.redhat.com/ecosystem/search/#/category/Server](https://access.redhat.com/ecosystem/search/#/category/Server)
KVM-RT - tuned profiles

Parents
- latency-performance

Children
- network-latency

Children/Grandchildren
- realtime
  - realtime-virtual-host
  - realtime-virtual-guest
KVM-RT – tuning/config host

# yum install tuned-profiles-realtime tuned-profiles-nfv

# echo "isolated_cores=3,5,7" >> /etc/tuned/realtime-virtual-host-variables.conf

# systemctl enable tuned
# systemctl start tuned
# tuned-adm profile realtime-virtual-host
KVM-RT – tuning/config host

In /etc/default/grub add:

default_hugepagesz=1G

Update the bootloader:

# grub2-mkconfig -o /boot/grub2/grub.cfg

Set hugepage reservation:

# echo 2 > /sys/devices/system/node/nodeY/hugepages/hugepages-1048576kB/nr_hugepages
```
[root@virtlab502 proc]# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-327.18.2.rt56.223.el7_2.x86_64 root=/dev/mapper/rhel_virtlab502-root ro
crashkernel=auto rd.lvm.lv=rhel_virtlab502/root rd.lvm.lv=rhel_virtlab502/swap
console=ttyS1,115200 default_hugepagesz=1G isolcpus=3,5,7 nohz=on nohz_full=3,5,7
intel_pstate=disable nosoftlockup
[root@virtlab502 proc]#

[root@virtlab502 proc]# tuned-adm active
Current active profile: realtime-virtual-host
[root@virtlab502 proc]#

[root@virtlab502 proc]# cat /usr/lib/tuned/realtime-virtual-host/lapic_timer_adv_ns
5000 = non matching 0
[root@virtlab502 proc]# cat /sys/module/kvm/parameters/lapic_timer_advance_ns
5000 = non matching 0
[root@virtlab502 proc]#

[root@virtlab502 ~]# cat /sys/devices/system/node/node1/hugepages/hugepages-1048576kB/nr_hugepages
2
[root@virtlab502 ~]#
```
[root@compute ~]# vi /etc/nova/nova.conf
....
# Defines which cpus that instance vcpus can use. For example, "4-12,"8,15"
# (string value)
vcpu_pin_set=<None>
vcpu_pin_set=3,5,7....

[root@controller ~]# . keystonerc_admin
[root@controller ~](keystone_admin)# nova flavor-create realTime.medium 99 1024 100 4
+--+-+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+-----------------+
| ID | Name      | Memory_MB | Disk | Ephemeral | Swap | VCPUs | RXTX_Factor | Is_Public |
+---+-----------+-----------+------+-----------+------+-------+-------------+---------+
| 99 | r1.small  | 1024      | 10   | 0         |      | 4     | 1.0         | True     |
+---+-----------+-----------+------+-----------+------+-------+-------------+---------+
[root@controller ~](keystone_admin)#
[root@controller ~](keystone_admin)# nova flavor-key 99 set hw:cpu_policy=dedicated
[root@controller ~](keystone_admin)# nova flavor-key 99 set hw:cpu_realtime=yes
[root@controller ~](keystone_admin)# nova flavor-key 99 set hw:cpu_realtime_mask="^6-1"
[root@controller ~](keystone_admin)# nova flavor-key 99 set hw:mem_page_size=1GB
KVM-RT – tuning/config guest

• Install kernel-rt in guest, too!
• Use same default_hugepagesz as host
• Install tuned profile: tuned-profiles-nfv

# echo “isolated_cores=2,3” >> /etc/tuned/realtime-virtual-guest-variables.conf

# tuned-adm profile realtime-virtual-guest

# grep tuned_params= /boot/grub2/grub.cfg
set tuned_params="isolcpus=2,3 nohz=on nohz_full=2,3 intel_pstate=disable nosoftlockup"
KVM-RT – apps/ops

- Target applications are NFV networking workloads
- Types of operations to avoid
  - Disk IO
  - Video or Sound
  - Page faults or swapping
  - CPU hot-plug
  - Live migration
KVM-RT – testing

Run cyclictest: confirm guest latencies within expected limits

# taskset -c 2 <application>

# taskset -c 2 cyclictest -m -n -q -p95 -D 24h -h100 -i 200 > cyclictest.out
KVM-RT – testing

- Tuning: real-time tuned profiles host and guest
- Fork app:
  - Simple application executing fork() repeatedly
  - Task that sends signal to fork app
KVM-RT – testing

cyclic -m -n -q -p95 -D 60s -h60 -i 200 -a 1

- RT kernel (3.10.0-327.18.2.rt56.223.el7_2.x86_64)
- RT kernel + fork app
- Non RT kernel (3.10.0-327.el7.x86_64) (isolated)
- Non RT kernel + fork app (isolated)
- Max for RT kernel: 16us
- Max for RT + fork-app: 27us
- Max for non-RT kernel: 36us
- Max for non-RT kernel + fork-app: 574us
KVM Live Migration

Precopy vs Auto Converge vs Postcopy
KVM Live Migration

- Software Dependability
  - Is Live Migration guaranteed to succeed?
    - Emergency Evacuation
    - Recoverable machine checks
    - Hardware maintainance
- Guest performance during live migration
  - Minimize CPU performance impact on the guest
- Live migration time
  - Take as little time as possible
    - To reduce network load as well
- Downtime latency
  - Minimize the downtime with source & destination both paused
Precopy

- Software Dependability
  - *No*
- Guest performance during live migration
  - Good
- Live migration time
  - Bad, could never end/converge
- Downtime latency
  - Low, if we sacrifice dependability & migration time
Auto Converge

- Software Dependability
  - Yes
- Guest performance during live migration
  - Bad, guest vCPUs may be throttled down heavily
- Live migration time
  - Bad, the CPU throttling process takes time
- Downtime latency
  - Low, same as precopy
    - Artificial “latency” created during the auto-converge phase
Postcopy after Precopy

- Software Dependability
  - **Yes**
    - Guest memory accesses might block waiting for network I/O *if the network hardware fails*
- Guest performance during live migration
  - **Guest vCPUs performance is never throttled down**
    - The first access to some memory page in the destination may be delivered at network I/O bandwidth instead of RAM bandwidth, similar to a disk swapin
- Live migration time
  - Lower than precopy & auto converge and *deterministic*
- Downtime latency
  - Lower than precopy & auto converge artificial latencies
1st and 2nd packet latency: 20 (ms)
Setting spike throttle to: 40 (ms)
Updating spike log initial timestamp

[2820272069] max delay: 24 (ms), last: 21 (ms)

}, q=-1--1, 5000 kb/s, 29.97 fps, 90k tbn, 29.97 tbc (default)
Metadata:
  encoder = L AVC56.60.100 libvpx

Stream mapping:
Stream #0:0 -> #0:0 (vp8 (native) -> vp8 (libvpx))

Press [q] to stop, [?] for help
Frame= 150 fps= 29 q=0.0 size= N/A time=00:00:05.00 bitrate=N/A

[root@localhost ~]# stress --vm 1 --vm-keep --vm-bytes $(1500*1024*1024)
stress: info: [1673] dispatching hogs: 0 cpu, 0 io, 1 vm, 0 hdd

host $ time virsh migrate --live --timeout 60 --timeout-suspend fedora23 qemu+ssh://ept/system
Live migration total time

- autoconverge
- postcopy
Live migration max UDP guest delivery latency

<table>
<thead>
<tr>
<th>Max UDP latency</th>
<th>precopy timeout</th>
<th>autoconverge</th>
<th>postcopy</th>
</tr>
</thead>
<tbody>
<tr>
<td>secs</td>
<td>0</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
Everything Else
“Meet The Experts” - Thurs 5:45-7 PM Free - Soda/Beer/Wine