



How to Reduce the Performance Impact of Using Multicast Networking with KVM Virtual Machines

Author: Perry Myers
Editor: Allison Pranger
06/02/2011

OVERVIEW

Certain applications require the use of multicast networking in order to function properly. These applications can be used in environments with KVM virtual machines (VMs) present on the same network, and in some cases, they can be used directly on KVM VMs. However, when multicast is used in conjunction with KVM VMs, the requirement for additional memory copies will impact performance (for example, throughput and latency of multicast packets).

This document identifies best practices that will reduce the performance impact of using multicast networking in conjunction with KVM.

Environment

- Applications that use multicast:
 - Red Hat Enterprise Linux 5+ Advanced Platform (Clustering and GFS/GFS2)
 - Red Hat Enterprise Linux 6+ with the High Availability Add-On or Resilient Storage Add-On
 - MRG Messaging (MRG-M) used with the Clustering option
- Hypervisors that impact performance of multicast traffic:
 - Red Hat Enterprise Linux 5 KVM
 - Red Hat Enterprise Linux 6 KVM
 - Red Hat Enterprise Virtualization 2.2

RECOMMENDATIONS

The following are identified best practices for using multicast with KVM VMs. Some recommendations should be applied in all situations, but others might depend on the specific application(s) or other environmental factors.

Use the Para-Virtualized Network Driver

KVM VMs should always be configured to use the para-virtualized network driver (virtio-net) card instead of fully emulated network cards like Realtek or e1000.

NOTE: By default, guests managed in Red Hat Enterprise Virtualization 2.2 will use virtio-net as long as it is supported by the guest operating system. This includes Red Hat Enterprise Linux 5.4+ and Red Hat Enterprise Linux 6.0+ guests.

To check to see if a specific network interface is using the virtio-net driver, you can use the output of either **ethtool -i** or **lspci**. If the interface is using virtio-net, you will see the following:

```
Cannot get driver information: Operation not supported (ethtool -i ethx)  
00:05.0 Ethernet controller: Red Hat, Inc Virtio network device (lspci)
```



Use KVM Kernel Modules for Hardware Acceleration

Ensure that hosts are using the KVM kernel modules for hardware acceleration. If the KVM kernel modules are not being used, the guest will run fully emulated, drastically reducing performance. The following commands can help identify whether a host is using KVM hardware acceleration.

- **lsmod | grep kvm** will show if the KVM modules are loaded; either **kvm_intel** or **kvm_amd** must be present:

```
kvm_intel      41950    3
kvm            257132    1 kvm_intel
```

- **virsh capabilities** should show that **/usr/bin/qemu-kvm** is one of the available emulators:

```
<guest>
<os_type>hvm</os_type>
<arch name='x86_64'>
...
<domain type='kvm'>
<emulator>/usr/bin/qemu-kvm</emulator>
```

- **/var/log/libvirt/qemu** will contain a QEMU log file for each VM; check the command line used by QEMU to launch the VM to see if **-enable-kvm** is present.

Disable STP

Spanning Tree Protocol (STP) should be disabled on all software bridges. You can check STP status by using the **brctl show** command. Sample output is shown below.

```
bridge      name bridge id      STP enabled  interfaces
breth1      8000.001320f5f974  no           eth1
```

STP should also be disabled or properly configured on hardware switches used on the same network as multicast. Proper STP configuration varies among switch models. Consult with Red Hat Support to determine how to best configure your hardware switch to use STP.

Avoid NAT

Because multicast over network address translation (NAT) virtual networks is much slower than it is over bridged networks, you should avoid using NAT when using multicast with KVM VMs.

Disable Bridge Netfilter

Bridge netfilter should be disabled. You can check the status with the following command: **cat /proc/sys/net/bridge/bridge-nf-***.

Each of the **bridge-nf-*** files should contain a zero (0). If any contain one (1), use **echo 0 > /proc/sys/net/bridge/bridge-nf-*** to set them all to disabled.

Set VMs to Use Two or More vCPUs with OpenAIS/Corosync

There is a known performance issue with OpenAIS/Corosync when VMs are used with one vCPU. To work around this issue, set each VM to use at least two vCPUs, even if this would cause overcommit of vCPUs on



the host. This issue will be resolved in a future version of **Corosync**.

Set `txqueuelen` to 5000

Setting `txqueuelen` to 5000 can improve performance in some situations.

NOTE: For this change to be effective, you must increase `txqueuelen` on both the physical interface and the bridge interface.

There are pros and cons to using this setting. Additional memory will be used for the larger transmit queue, it can add additional latency, and if the guests cannot keep up with the volume of traffic, it just delays the inevitable. However, this setting can be useful if the multicast traffic is cyclic and has occasional spikes as it provides some additional buffer room.

To set an interface to use a different `txqueuelen`, use the following commands.

```
ifconfig eth0 txqueuelen 5000
ifconfig breth0 txqueuelen 5000
```

The default value for `txqueuelen` is 1000, but you can check the setting by examining the output of `ifconfig ethX` prior to modifying the `txqueuelen`.