

oVirt SR-IOV support

Barak Azulay
Senior Manager, Software Engineering

Credits: Alona Kaplan, Martin Polednik, Ido Barkan

Red Hat
21/08/15

- SR-IOV basics (what, how, limitations)
- Ovirt Networking basics
- Ovirt Implementation of SR-IOV support
- Future improvements

specification that allows a PCIe device to appear to be multiple separate physical PCIe devices.

Full PCIe device that includes the SR-IOV capabilities.

'lightweight' PCIe functions that contain the resources necessary for data movement but have a carefully minimized set of configuration resources.

oVirt SR-IOV basics - how to add VFs

Before

```
[root@nari04 ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: enp2s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master ovirtmgmt state UP mode DEFAULT qlen 1000
    link/ether 78:e7:d1:e4:8f:16 brd ff:ff:ff:ff:ff:ff
4: enp2s0f1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP mode DEFAULT qlen 1000
    link/ether 78:e7:d1:e4:8f:17 brd ff:ff:ff:ff:ff:ff
```

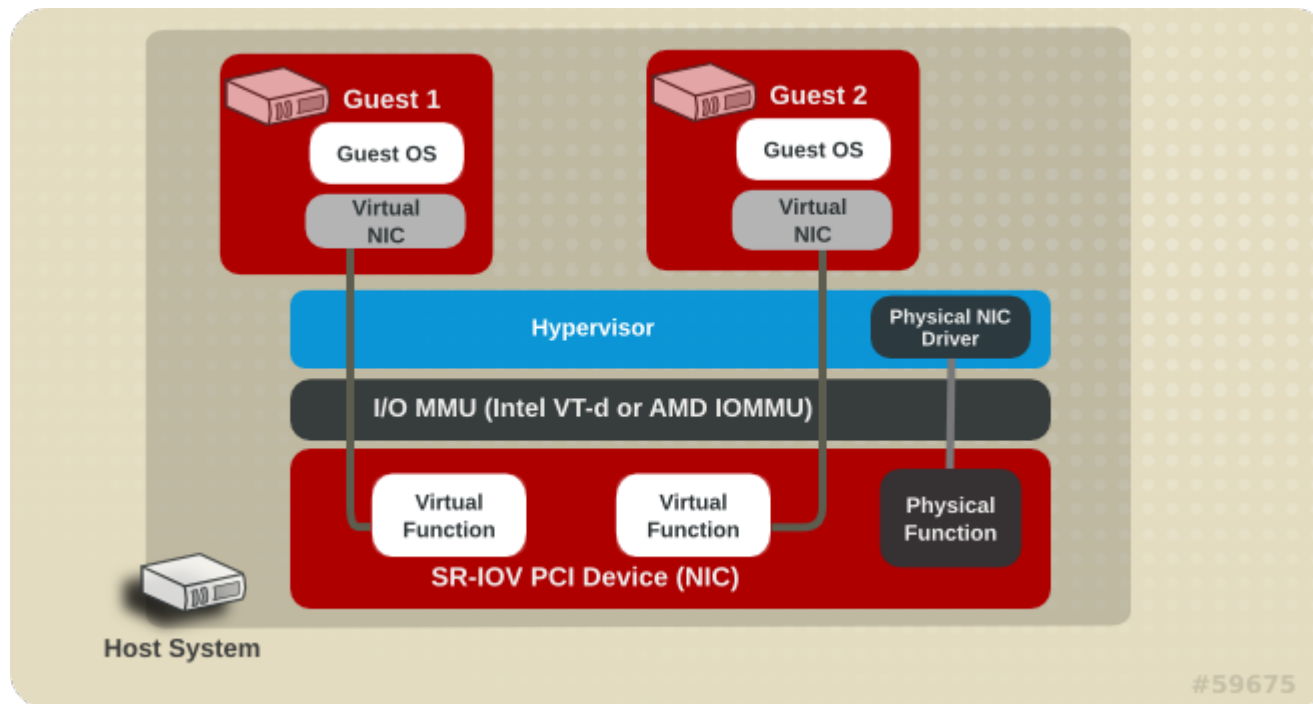
Add VFs

```
[root@nari04 ~]# echo 4 > /sys/class/net/enp2s0f0/device/sriov_numvfs
```

After

```
[root@nari04 ~]# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
3: enp2s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq master ovirtmgmt state UP mode DEFAULT qlen 1000
    link/ether 78:e7:d1:e4:8f:16 brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 00:00:00:00:00:00, spoof checking on, link-state auto
    vf 1 MAC 00:00:00:00:00:00, spoof checking on, link-state auto
    vf 2 MAC 00:00:00:00:00:00, spoof checking on, link-state auto
    vf 3 MAC 00:00:00:00:00:00, spoof checking on, link-state auto
4: enp2s0f1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc mq master bond0 state UP mode DEFAULT qlen 1000
    link/ether 78:e7:d1:e4:8f:17 brd ff:ff:ff:ff:ff:ff
35: enp2s16: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 4a:2f:20:98:fa:14 brd ff:ff:ff:ff:ff:ff
36: enp2s16f2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether fe:0c:29:cc:b5:fa brd ff:ff:ff:ff:ff:ff
37: enp2s16f4: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether 4a:c3:8f:6d:6e:40 brd ff:ff:ff:ff:ff:ff
38: enp2s16f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT qlen 1000
    link/ether b2:32:2a:82:4d:fd brd ff:ff:ff:ff:ff:ff
```

oVirt SR-IOV basics – Hypervisor view



- ✓ VFs have near-native **performance**.
- ✓ low **latency**.
- ✓ **scalability** of the host is improved (more CPU available to apps in VMs).
- ✓ VM has **direct** access to the hardware.
- ✓ **Guest protection/isolation**
- ✓ VMs can **share** a single physical port.

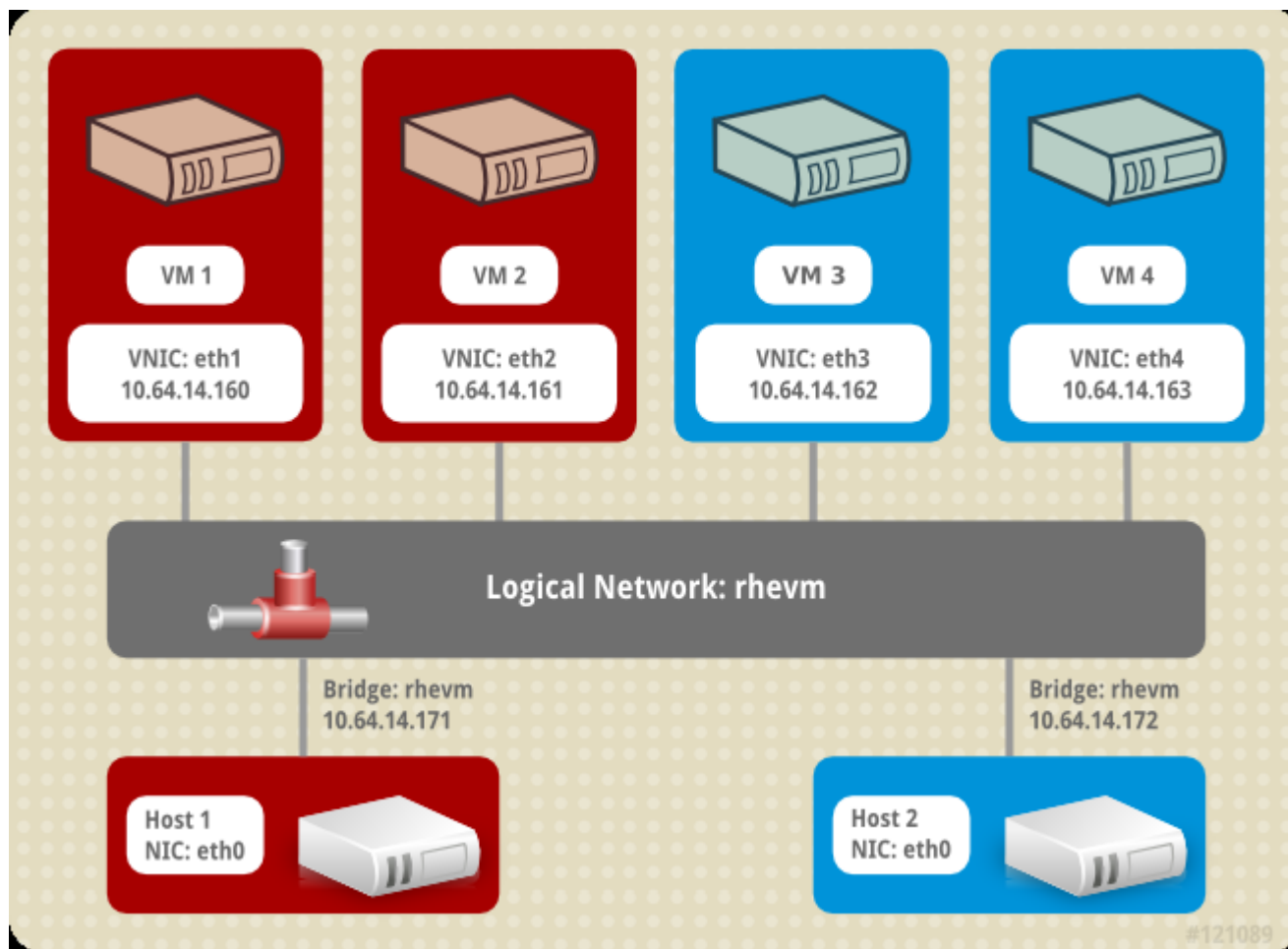
- ✗ Vfs number is limited by the device hardware.
- ✗ realistic 'num of VFs' should be set manually.
- ✗ VFs have limited configuration functions.
- ✗ live migration.

- hypervisor
 - hardware IOMMU support (AMD-Vi, Intel VT-d enabled in BIOS) .
 - kernel enabled IOMMU support (intel_iommu=on for Intel, amd_iommu=on for AMD in kernel cmdline) .
 - SR-IOV capable hardware.
 - RHEL7 or newer (kernel ≥ 3.6).
- SR-IOV support in the guest (driver).

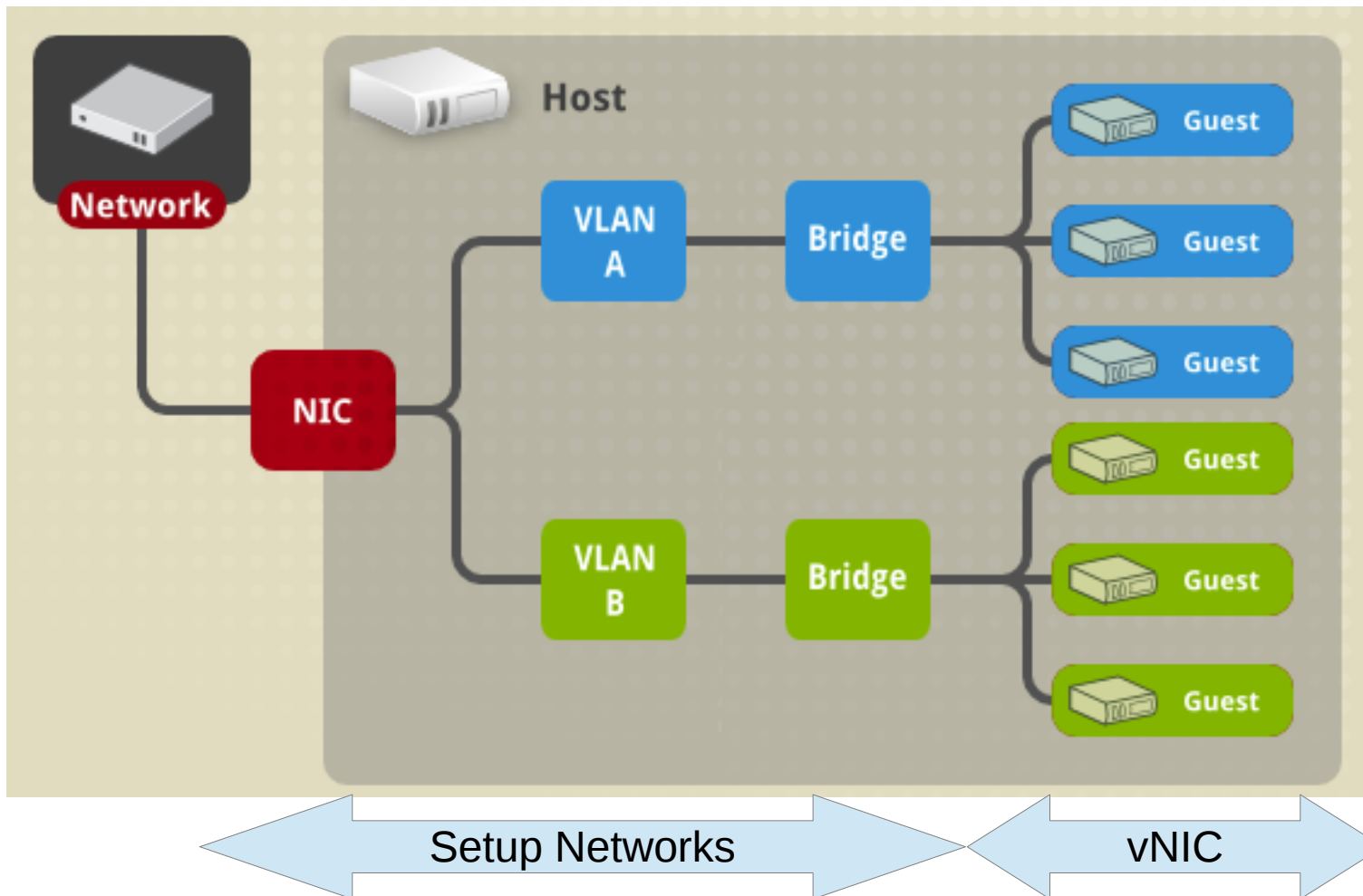
oVirt Networking

- Logical Network (VM, non-VM).
- Setup networks - Configuring the logical networks on the hypervisor
- VM Interface Profile (vNic profile).
- VM Interface (vNic).

Logical Network



Host Networks setup



oVirt Attaching VM to a new network demo

oVirt OPEN VIRTUALIZATION MANAGER

Network: [] [x] [☆] [Q]

Data Centers Clusters Hosts **Networks** Storage Disks Virtual Machines Pools Templates Volumes Users Events

New Import Edit Remove

Name	Comment	Data Center	Description	Role	VLAN tag	Label	Provider
ovirtmgmt		Default	Management Network	✓	-	-	
net-1		mb		✓	-	-	
net-10		mb		✓	-	-	
net111		mb		✓	-	-	
net-2		mb		✓	-	mb	
net-3		mb		✓	-	-	
net-4		mb		✓	-	-	
net-5		mb		✓	-	-	
net-6		mb		✓	-	-	
net-7		mb		✓	-	-	
net-8		mb		✓	-	-	
net-9		mb		✓	-	-	
ovirtmgmt		mb	Management Network	✓	-	-	

Last Message: ✓ 2015-Aug-06, 12:19 Network sr_lov_net1 was removed from Data Center: mb

Alerts (8) Events Tasks (10)

- ✓ 2015-Aug-06, 12:19 Network sr_lov_net1 was removed from Data Center: mb
- ✓ 2015-Aug-06, 12:18 Interface nic1 (VirtIO) was removed from VM vm_6_ (User: admin@internal)
- ✓ 2015-Aug-06, 12:17 Network changes were saved on host puma22.scl.ltv.redhat.com
- ✓ 2015-Aug-06, 12:15 Interface nic1 (VirtIO) was added to VM vm_6_ (User: admin@internal)

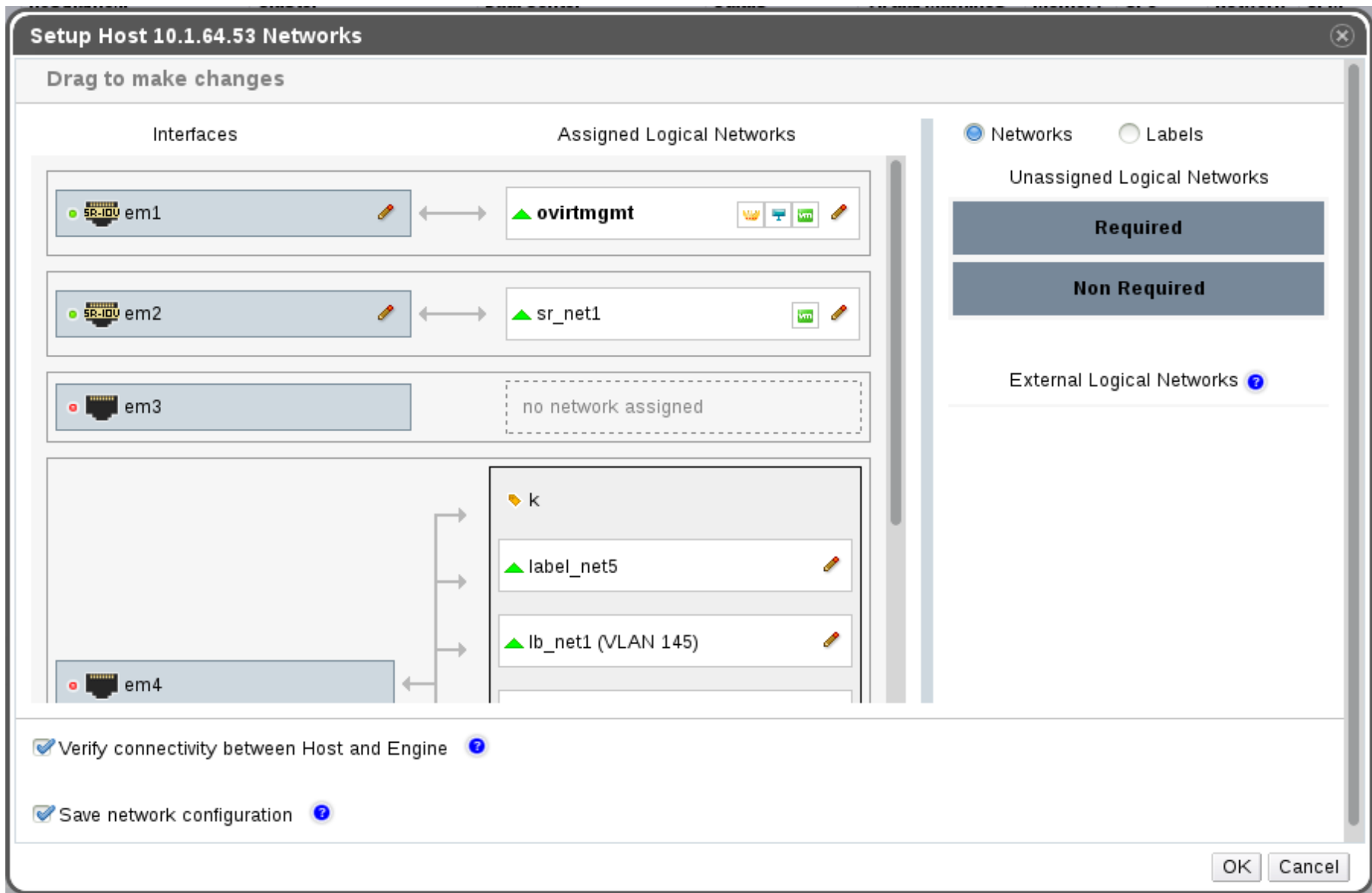
OVirt & SR-IOV

The problem:

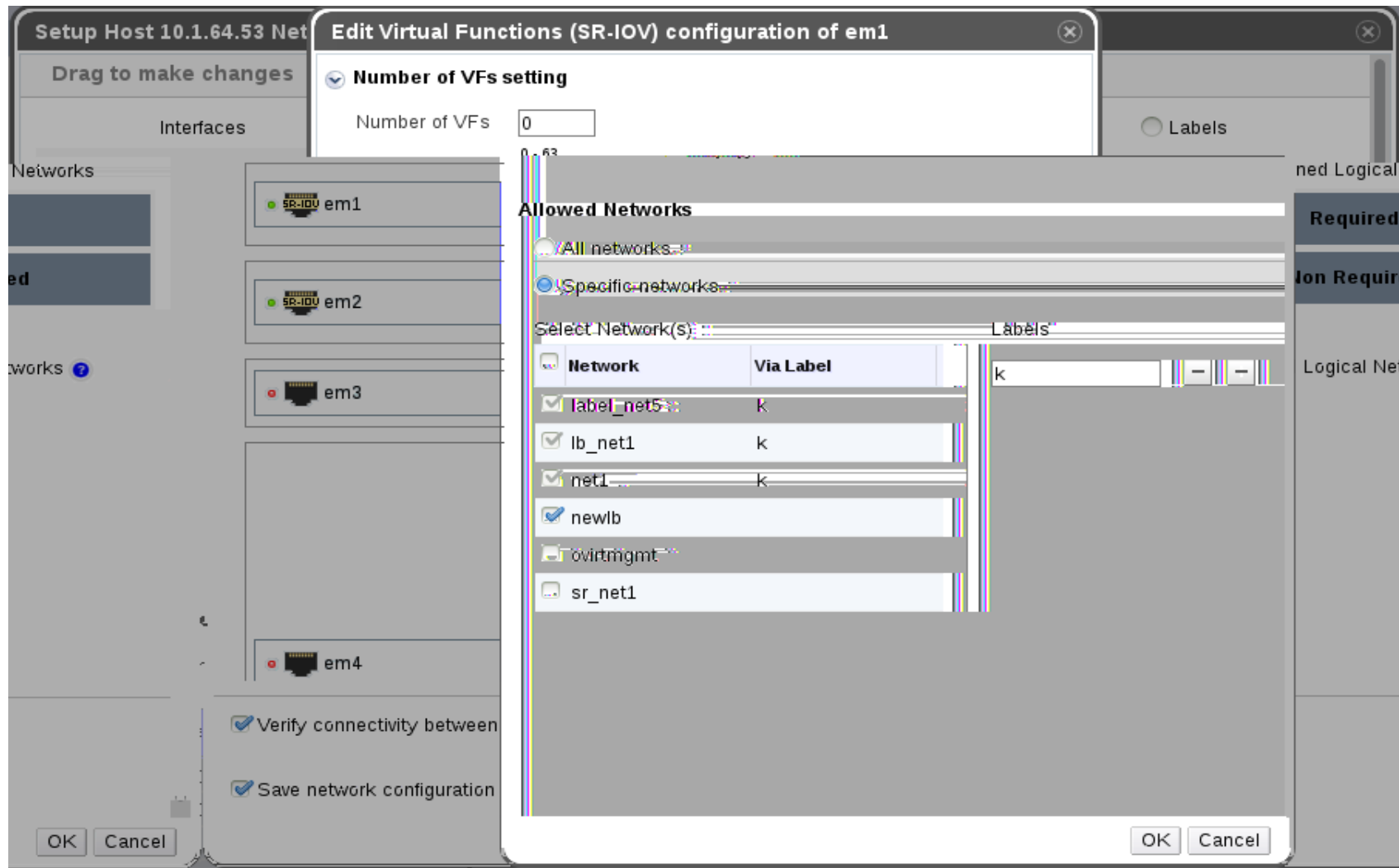
- SR-IOV passthrough belongs to the physical layer of Network
- It is not associated with logical network

The solution:

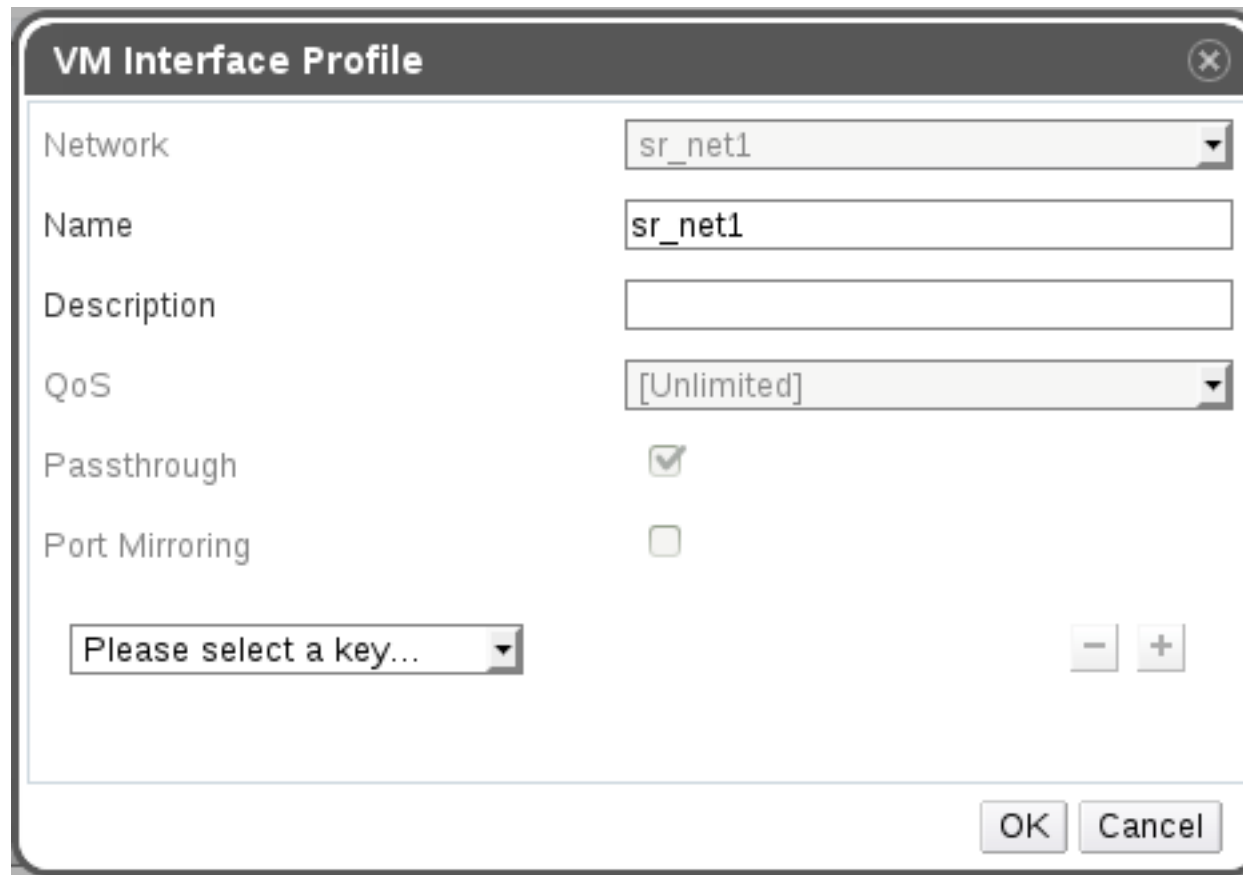
- Define in advance the networks list that could be used by the SR-IOV device (PF)
- Add specific vNIC profile type of passthrough
- Associate the vNIC to the passthrough vNIC profile



VFs configuration



oVirt Passthrough VM Interface profile



The image shows a screenshot of the "VM Interface Profile" dialog box in oVirt. The dialog has a title bar with a close button. It contains several fields and checkboxes:

- Network:** A dropdown menu showing "sr_net1".
- Name:** A text field containing "sr_net1".
- Description:** An empty text field.
- QoS:** A dropdown menu showing "[Unlimited]".
- Passthrough:** A checked checkbox.
- Port Mirroring:** An unchecked checkbox.
- Key Selection:** A dropdown menu showing "Please select a key..." with minus and plus buttons next to it.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

Edit Network Interface

Name

nic1

Profile

sr_net1/sr_net1

Type


PCI Passthrough


☐ Custom MAC address

00:1a:4a:16:01:51


Example: 00:14:4a:23:67:55


Link State

☒  Up

☐  Down

Card Status

☒  Plugged

☐  Unplugged

OK

Cancel

oVirt Run VM with passthrough vNic

The screenshot displays the oVirt Open Virtualization Manager web interface. The top navigation bar includes the oVirt logo, the title "OPEN VIRTUALIZATION MANAGER", and user options for "admin", "Configure", "Guide", "About", and "Feedback". A search bar labeled "Network:" is present. Below the navigation bar, a series of tabs allows switching between different system components: Data Centers, Clusters, Hosts, Networks (selected), Storage, Disks, Virtual Machines, Pools, Templates, Volumes, and Users. An "Events" button is also visible.

The main content area shows a table of network configurations. The table has columns for Name, Comment, Data Center, Description, Role, VLAN tag, Label, and Provider. The table lists several networks, including "ovirtmgmt" (Management Network) and "net-sr-iov1" (VLAN 162). The "net-sr-iov1" network is highlighted in blue.

Below the table, there are tabs for "General", "vNIC Profiles", "Clusters", "Hosts", "Virtual Machines", "Templates", and "Permissions". The "General" tab is active, showing details for the selected network "net-sr-iov1". The details include:

- Name: net-sr-iov1
- Id: 819dd182-3e79-446f-ad80-efd7487e0208
- Description:
- VM Network: true
- VLAN tag: 162
- MTU: Default (1500)

At the bottom of the interface, a status bar shows a "Quit" button, a system message icon, and a log of the last message: "2015-Aug-06, 16:49 VM vm_sriov was powered off ungracefully by admin@internal (Host: puma22.scl.lab.tlv.redhat.com) (Reason: Not Specified)". On the right side of the status bar, there are icons for "Alerts (8)", "Events", and "Tasks (10)".

- Change & persist number of VFs (sysfs) via ui.
- Managing PFs network connectivity white-list.
- Scheduling – no need to pin to a host
- Setting VLAN and MAC address on a VF.

SR-IOV capabilities- cont

- Mixed mode- bridged PF with VFs.
- Specifying boot order on Vfs (enableing booting VM with passthrough vNics from pxe).

- Hot plug/unplug passthrough vnics.
- Live Migration
- Opportunistic passthrough vnic.

VF missing functionality

- MTU (not supported)
- QoS (in/out- average link share, average upper limit, average real time).

Hardware issues

- VFs share the IOMMU group.
- IOMMU is not supported (under sysfs - the devices doesn't get iommu-group number).
- Hacks are needed
 - pci=realloc - 'igb <0000:02:00.1>: not enough MMIO resources for SR-IOV'
 - pci=assign-busses - 'igb <0000:06:10.0>: SR-IOV: bus number out of range'
 - vfio_iommu_type1.allow_unsafe_interrupts=1
 - On systems with broken interrupt remapping (problematic chipset)

Questions ?

THANK YOU!

<http://www.ovirt.org>
bazulay@redhat.com
[bazulay@irc.oftc.net#ovirt](irc://irc.oftc.net/#ovirt)