RED HAT SUMMIT

Hyper-converged OpenStack and Ceph

Deployment, Resource Isolation, and NFV Performance

John Fulton
Senior Software Engineer

Andrew Theurer
Senior Principal Software Engineer

May 2, 2017

Agenda

- What is Hyper-converged Infrastructure (HCI)?
- HCI Automation and Deployment
- HCI Resource Isolation
- HCI and NFV

What is Hyper-converged Infrastructure (HCI)?

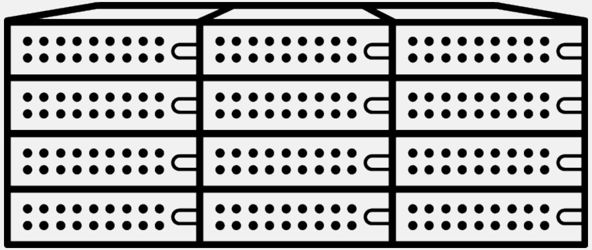
- A server which runs both compute and storage processes is a hyper-converged node

- Today’s focus is on running OpenStack Nova Compute and Ceph Object Storage Daemon services on the same server

- HCI is not a step backwards to local storage because it does not introduce a single point of failure

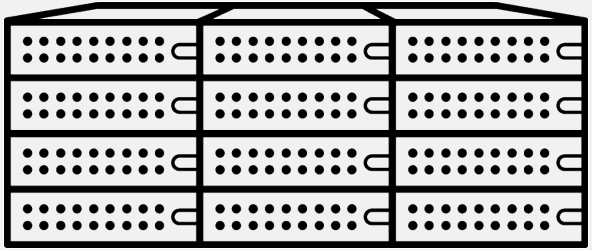
Local Compute/Storage: Single Point of Failure

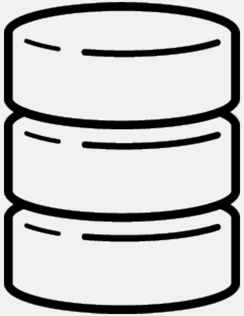
Each server only uses its local disks



Separate Storage/Compute: No SPoF

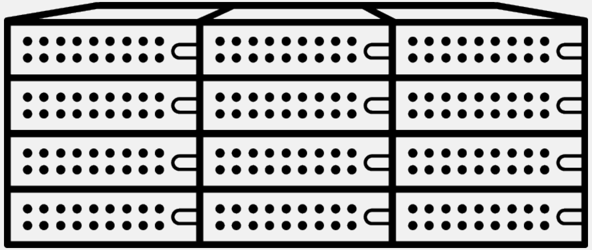
Servers store their data on separate storage cluster





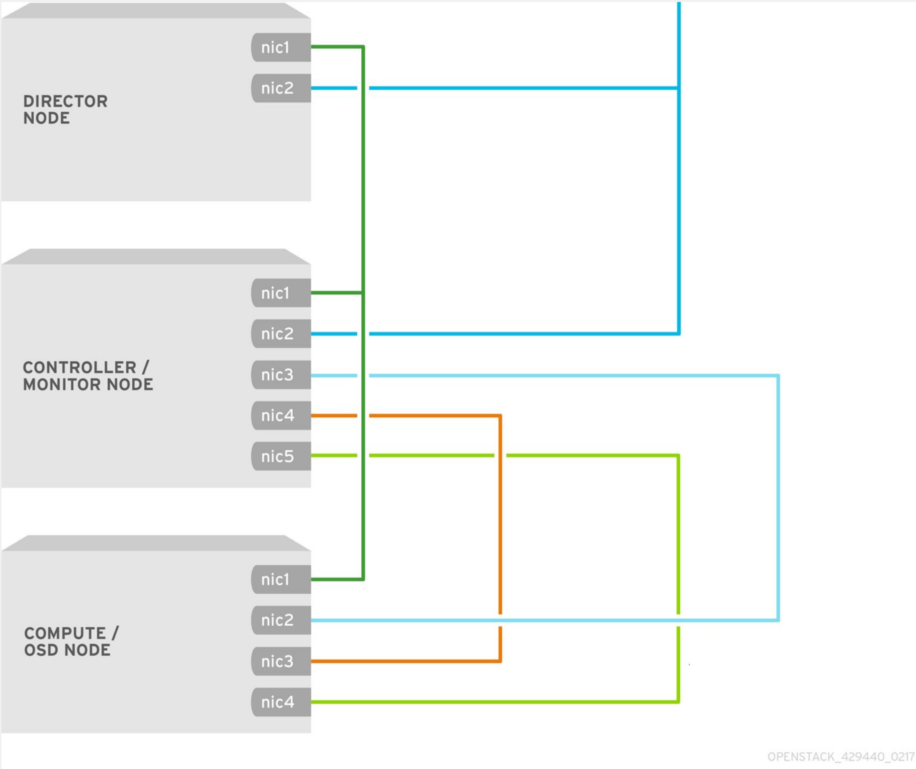
HCI: Local Storage without SPoF

Each server adds to Compute and Storage resources without being a single point of failure



Two clusters, Compute and Storage, but both are using resources from the same servers

OpenStack/Ceph HCI Architecture



Why is there demand for HCI?

- HCI lets us deploy a *smaller footprint*
  - HCI: 6 nodes for HA (+ director)
    - 3 controllers/monitors + 3 computes/OSDs
  - Non-HCI: 9 nodes for HA (+ director)
    - 3 controllers/monitors + 3 computes + 3 OSDs

- Further standardization of hardware
  - Hardware vendors may offer a discount for more of the same type of server
  - Fewer server types simplify operations
  - May enable more efficient use of hardware resources

Small Footprint Benefits

- An implication of NFV is putting more power on the edge of the network
- Devices put on the edge of the network should be small and dense
- PoC requires a smaller initial investment

What are the tradeoffs?

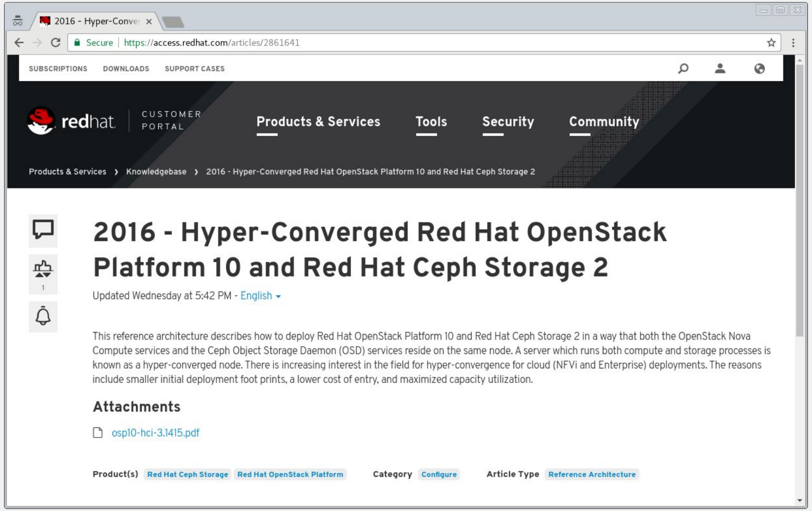
- Deployment Complexity
  - Deploy OpenStack and Ceph on the same server

- Resource Management
  - Compute and Storage are resource intensive and may contend

The above are my focus today and are covered in more detail in a Reference Architecture.

HCI Reference Architecture

https://access.redhat.com/articles/2861641


2016 - Hyper-Conve
Secure | https://access.redhat.com/articles/2861641
SUBSCRIPTIONS
DOWNLOADS
SUPPORT CASES
redhat. CUSTOMER PORTAL
Products & Services
Tools
Security
Community
Products & Services › Knowledgebase › 2016 - Hyper-Converged Red Hat OpenStack Platform 10 and Red Hat Ceph Storage 2
1

2016 - Hyper-Converged Red Hat OpenStack Platform 10 and Red Hat Ceph Storage 2

Updated Wednesday at 5:42 PM - English

This reference architecture describes how to deploy Red Hat OpenStack Platform 10 and Red Hat Ceph Storage 2 in a way that both the OpenStack Nova Compute services and the Ceph Object Storage Daemon (OSD) services reside on the same node. A server which runs both compute and storage processes is known as a hyper-converged node. There is increasing interest in the field for hyper-convergence for cloud (NFVi and Enterprise) deployments. The reasons include smaller initial deployment foot prints, a lower cost of entry, and maximized capacity utilization.

Attachments

osp10-hci-3.1415.pdf

Product(s) Red Hat Ceph Storage Red Hat OpenStack Platform **Category** Configure **Article Type** Reference Architecture

All Heat environment files and scripts available at https://github.com/RHsyseng/hci

HCI Support

- HCI as described today is in Technology Preview* for Red Hat OpenStack Platform 10
- Full support for HCI is targeted for Red Hat OpenStack Platform 11

* https://access.redhat.com/support/offerings/techpreview

It is possible to file a support exception for HCI in Red Hat OpenStack Platform 10

Deploying HCI with Director

Deploying Red Hat OpenStack Platform

- Red Hat OpenStack Platform should be deployed with director
    - By *deploy* I mean bare metal installation, scale up/down, and config management

- You cannot efficiently deploy an OpenStack cloud without automation

- Red Hat OpenStack Platform director is Red Hat’s solution to this problem

New environment file in v10

Red Hat OpenStack director 10 ships the following new environment file:

```
~/templates/environments/hyperconverged-ceph.yaml
```

Including the above in a deployment will result in Ceph OSDs and Nova Computes residing on the same node.

Code: hypercovnerged-ceph.yaml

```
resource_registry:
  OS::TripleO::Compute::Ports::StorageMgmtPort: ../network/ports/storage_mgmt.yaml

parameter_defaults:
  ComputeServices:
      - OS::TripleO::Services::CephOSD
      - OS::TripleO::Services::NovaCompute
      - OS::TripleO::Services::NovaLibvirt
      - OS::TripleO::Services::Timezone
      ...
```

The list above continues with all services found on a compute node

Explanation: hypercovnerged-ceph.yaml

- Add the storage management port to the compute node

- Add the Ceph OSD service to the compute node role

- Keep the list of standard services for a compute node

The above are now possible because of composable services

Can I mix converged and non-converged?

- Yes, Red Hat OpenStack Platform director’s composability is powerful and flexible

- Custom roles may be combined and a mixed-nodes scenario is available from the reference architecture’s GitHub site: https://github.com/RHsyseng/hci

- Scenario covered:
  - Standard Nova Compute
  - Standard Ceph Storage with 10 OSDs
  - Converged Compute/Storage with 12 OSDs
  - Converged Compute/Storage with 6 OSDs
- This is not ideal for balancing load but Red Hat OpenStack director is not the limit

Applying Resource Isolation with Director

Why Isolate Resources?

- Contention between Ceph and OpenStack could result in degradation of either service
- A spike in one service could negatively affect the other
- Neither service is aware of the other’s presence on the same physical host

Tuning Nova Compute for HCI

- Limit Nova’s memory and CPU resources so Ceph can use what it needs of them
- Appropriately change the following defaults in `/etc/nova/nova.conf`

```
reserved_host_memory_mb = 512
cpu_allocation_ratio = 16.0
```

Nova Reserved Memory

- The amount of memory to reserve for the host

- Should normally be tuned to maximize the number of guests while protecting host

- For HCI, it should maximize guests while protecting host *and Ceph*

- How much to reserve?
  - Reserve 3G of RAM per OSD
  - Reserve 0.5G of RAM overhead per guest for the host

  The above could be modified after testing but are a good starting point

Nova Reserved Memory for HCI

We can figure out the reserved memory with a formula

```
left_over_mem = mem - (GB_per_OSD * osds)


number_of_guests = int(left_over_mem /
                                (average_guest_size + GB_overhead_per_guest))


nova_reserved_mem_MB = MB_per_GB * (
                                          (GB_per_OSD * osds) +
                                          (number_of_guests * GB_overhead_per_guest) )
```

Nova CPU Allocation Ratio

- Used by Nova scheduler when choosing compute nodes for guests

- If the ratio has default of 16:1 and a node has 56 cores on a node, then the scheduler may schedule enough guests to consume 896 vCPUs before it considers the node full

- Nova scheduler does not know about Ceph so modify it to not allocate Ceph’s CPUs

Nova CPU Allocation Ratio for HCI

We can figure out the the CPU allocation ratio with a formula

```
cores_per_OSD = 1.0
average_guest_util = 0.1 # 10%
nonceph_cores = cores - (cores_per_OSD * osds)
guest_vCPUs = nonceph_cores / average_guest_util
cpu_allocation_ratio = guest_vCPUs / cores
```

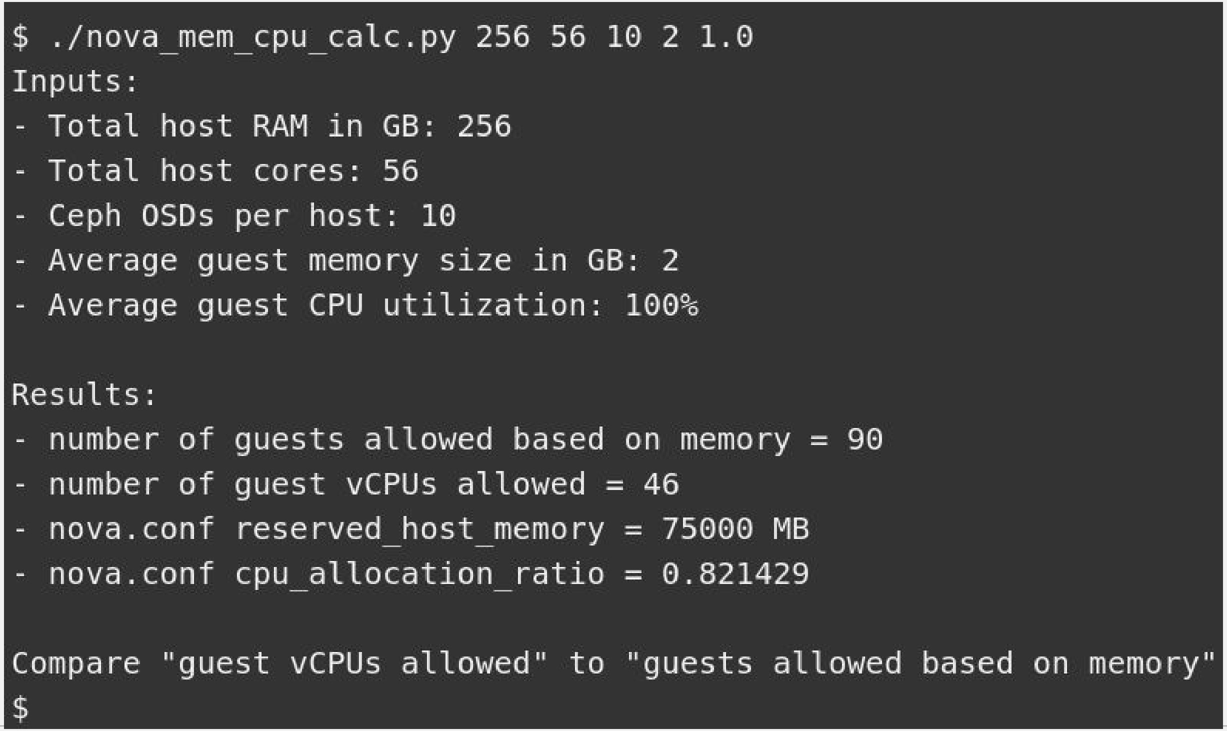
- The above is for rotational hard drives.
- Increase the core per OSD if you use an NVMe SSD to keep up with it

Nova Memory and CPU Calculator

- Takes the following inputs
  1. Total host RAM in GB
  2. Total host cores
  3. Ceph OSDs per server
  4. Average guest size in GB
  5. Average guest CPU utilization (0.0 to 1.0)

- Returns `nova.conf` `reserved_host_memory_mb`and `cpu_allocation_ratio`

Nova Memory and CPU Calculator: Busy VMs

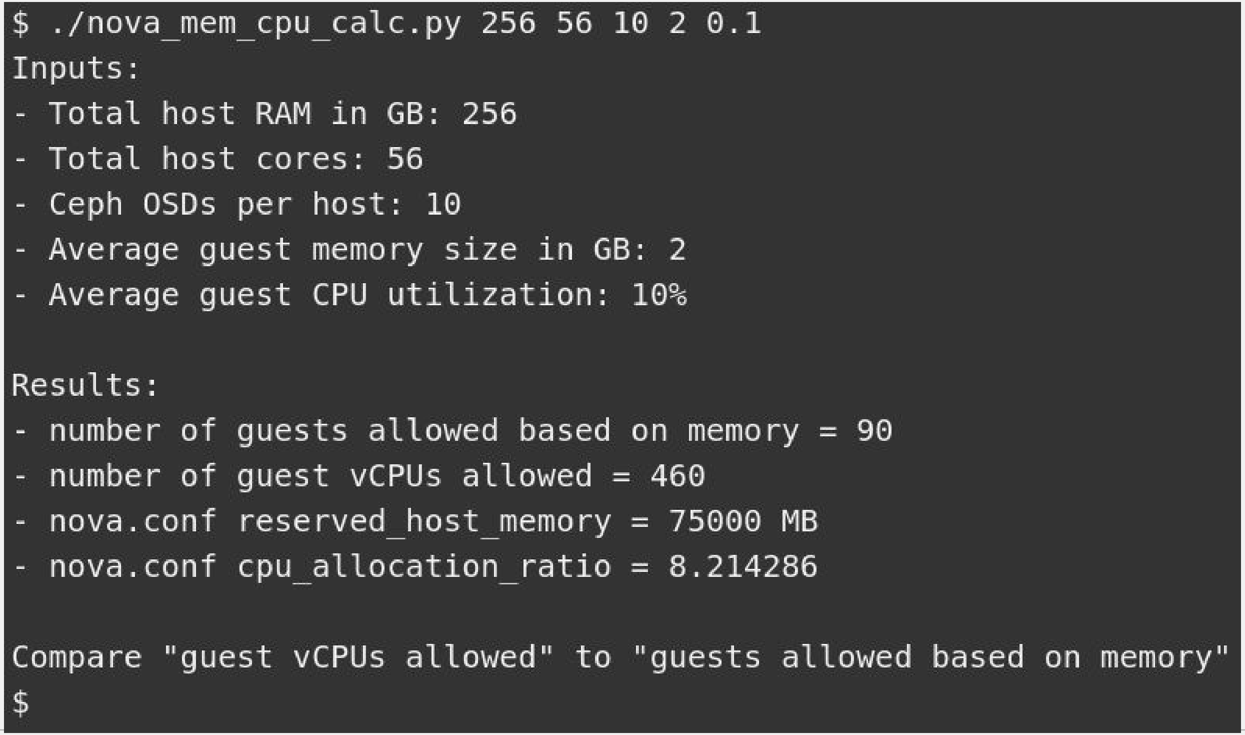


```
$ ./nova_mem_cpu_calc.py 256 56 10 2 1.0
Inputs:
- Total host RAM in GB: 256
- Total host cores: 56
- Ceph OSDs per host: 10
- Average guest memory size in GB: 2
- Average guest CPU utilization: 100%

Results:
- number of guests allowed based on memory = 90
- number of guest vCPUs allowed = 46
- nova.conf reserved_host_memory = 75000 MB
- nova.conf cpu_allocation_ratio = 0.821429

Compare "guest vCPUs allowed" to "guests allowed based on memory"
$
```

Nova Memory and CPU Calculator: Many VMs



```
$ ./nova_mem_cpu_calc.py 256 56 10 2 0.1
Inputs:
- Total host RAM in GB: 256
- Total host cores: 56
- Ceph OSDs per host: 10
- Average guest memory size in GB: 2
- Average guest CPU utilization: 10%

Results:
- number of guests allowed based on memory = 90
- number of guest vCPUs allowed = 460
- nova.conf reserved_host_memory = 75000 MB
- nova.conf cpu_allocation_ratio = 8.214286

Compare "guest vCPUs allowed" to "guests allowed based on memory"
$
```

Pass Nova Tunings to Director

The following could be added to a Heat environment template and passed to a deployment

```
parameter_defaults:
    ExtraConfig:
      nova::compute::reserved_host_memory: 75000
      nova::cpu_allocation_ratio: 8.2
```

Nova Memory and CPU Calculator Download

https://github.com/RHsyseng/hci/blob/master/scripts/nova_mem_cpu_calc.py

Tuning Ceph OSDs for HCI

- We limited Nova’s memory and CPU resources so Ceph and the OS can use them
- We now want to use `numactl` to pin the Ceph process to a NUMA node
- The socket to which Ceph should be pinned is the one that has the network IRQ
  - If a workload is network intensive and not storage intensive, this may not true.

NUMA pinning Ceph

- RHCS2's systemd unit file has a `$cmd` variable for the osd binary to start the cluster
- The unit file can be modified to redefine `$cmd` with the `numactl` command like so:

Before:

```
$cmd --cluster $cluster -f
```

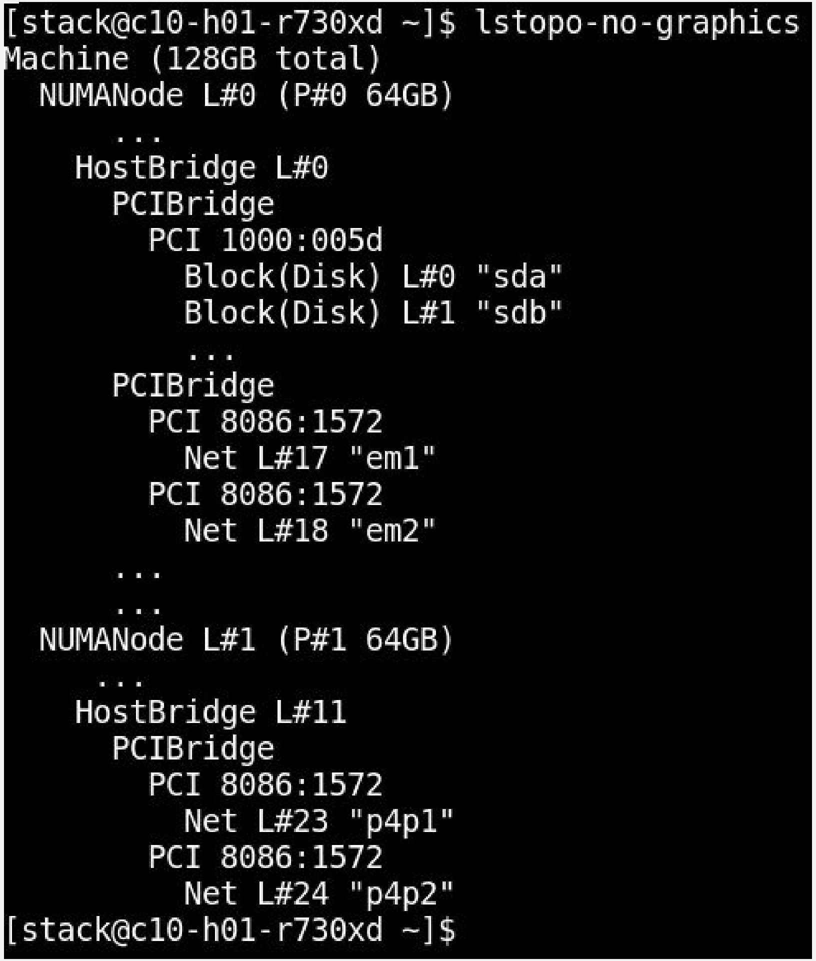
After:

```
numactl -N $numasocket --preferred=$numasocket $cmd --cluster $cluster -f
```

Automating the NUMA Change

A Red Hat OpenStack director post-deploy Heat environment file modifies the systemd unit file

```
ExtraConfig:
  type: OS::Heat::SoftwareConfig
  properties:
    group: script
    inputs:
      - name: OSD_NUMA_INTERFACE
    config: {get_file: numa-systemd-osd.sh}
ExtraDeployments:
  type: OS::Heat::SoftwareDeployments
  properties:
    servers: {get_param: servers}
    config: {get_resource: ExtraConfig}
    input_values:
      OSD_NUMA_INTERFACE: 'em2'
    actions: ['CREATE']
```



```
[stack@c10-h01-r730xd ~]$ lstopo-no-graphics
Machine (128GB total)
  NUMANode L#0 (P#0 64GB)
      ...
    HostBridge L#0
      PCIBridge
        PCI 1000:005d
          Block(Disk) L#0 "sda"
          Block(Disk) L#1 "sdb"
          ...
      PCIBridge
        PCI 8086:1572
          Net L#17 "em1"
        PCI 8086:1572
          Net L#18 "em2"
      ...
      ...
  NUMANode L#1 (P#1 64GB)
     ...
    HostBridge L#11
      PCIBridge
        PCI 8086:1572
          Net L#23 "p4p1"
        PCI 8086:1572
          Net L#24 "p4p2"
[stack@c10-h01-r730xd ~]$
```

Heat Environment Files

Templates used to deploy HCI/DPDK in Scale Lab:

https://github.com/redhat-performance/openstack-templates

Navigate to RDU-Scale/Newton/R730xdHciDpdk

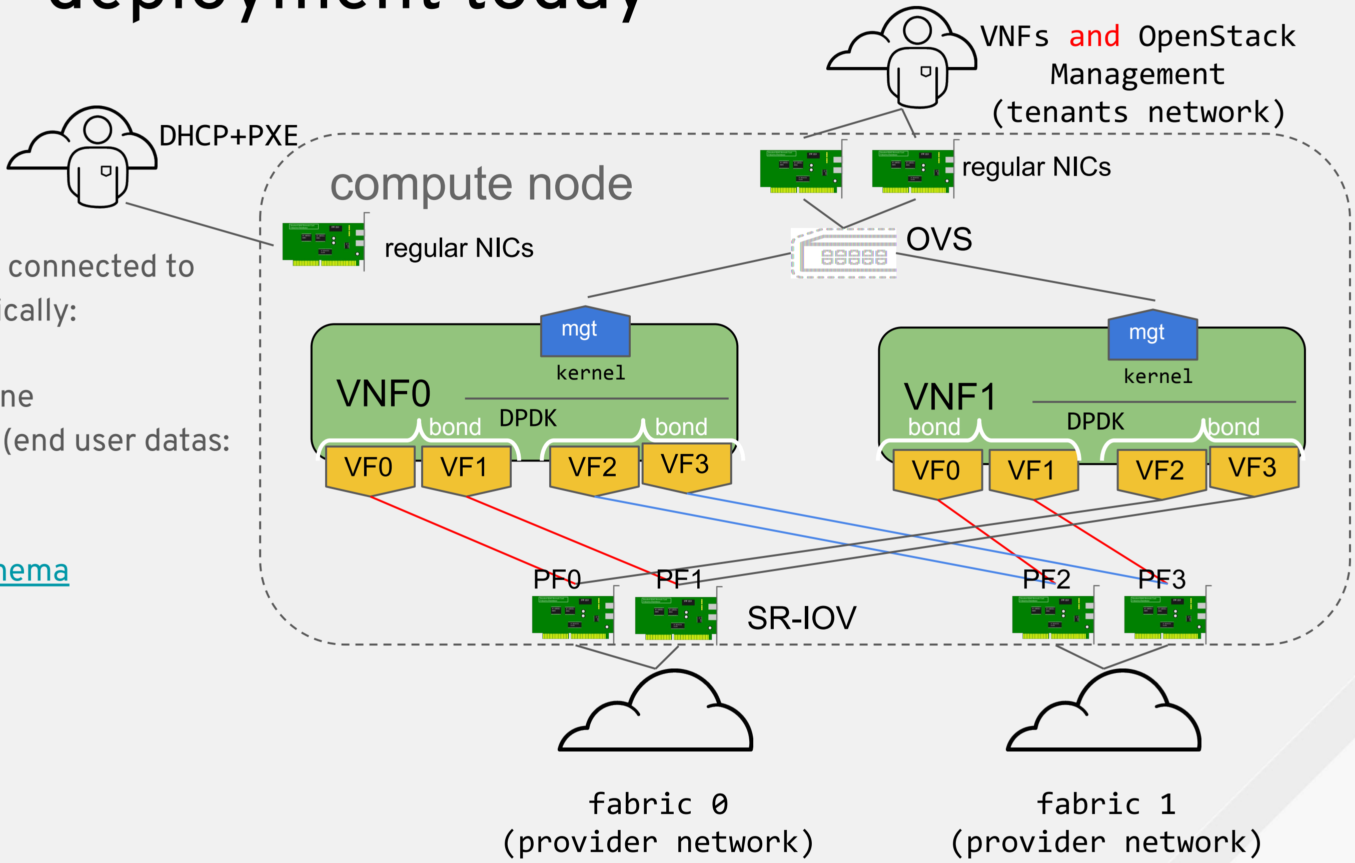
HCI for NFV

Typical VNF deployment today

Depending on the VNF, it will be connected to one or more fabric network, typically:

- One fabric for control plane
- One fabric for user plane (end user datas: mobile traffic, web, ...)

RHOSP director deployment schema



SR-IOV Host/VNFs guests resources partitioning

Typical 18 cores per node dual socket compute node (E5-2599 v3)

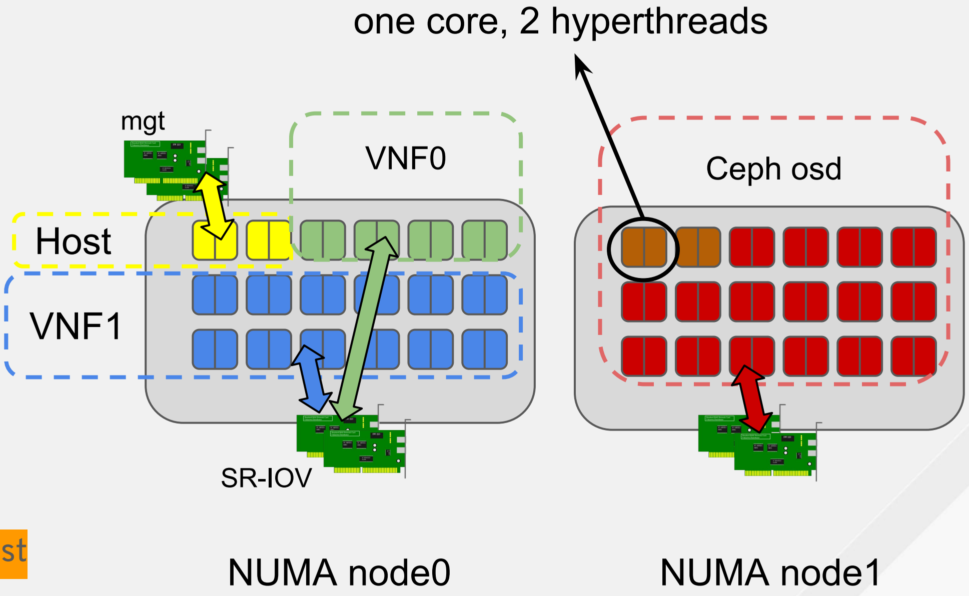
This looks like RT but is not RT, just partitioning

SR-IOV interfaces bonding handled by the VNF

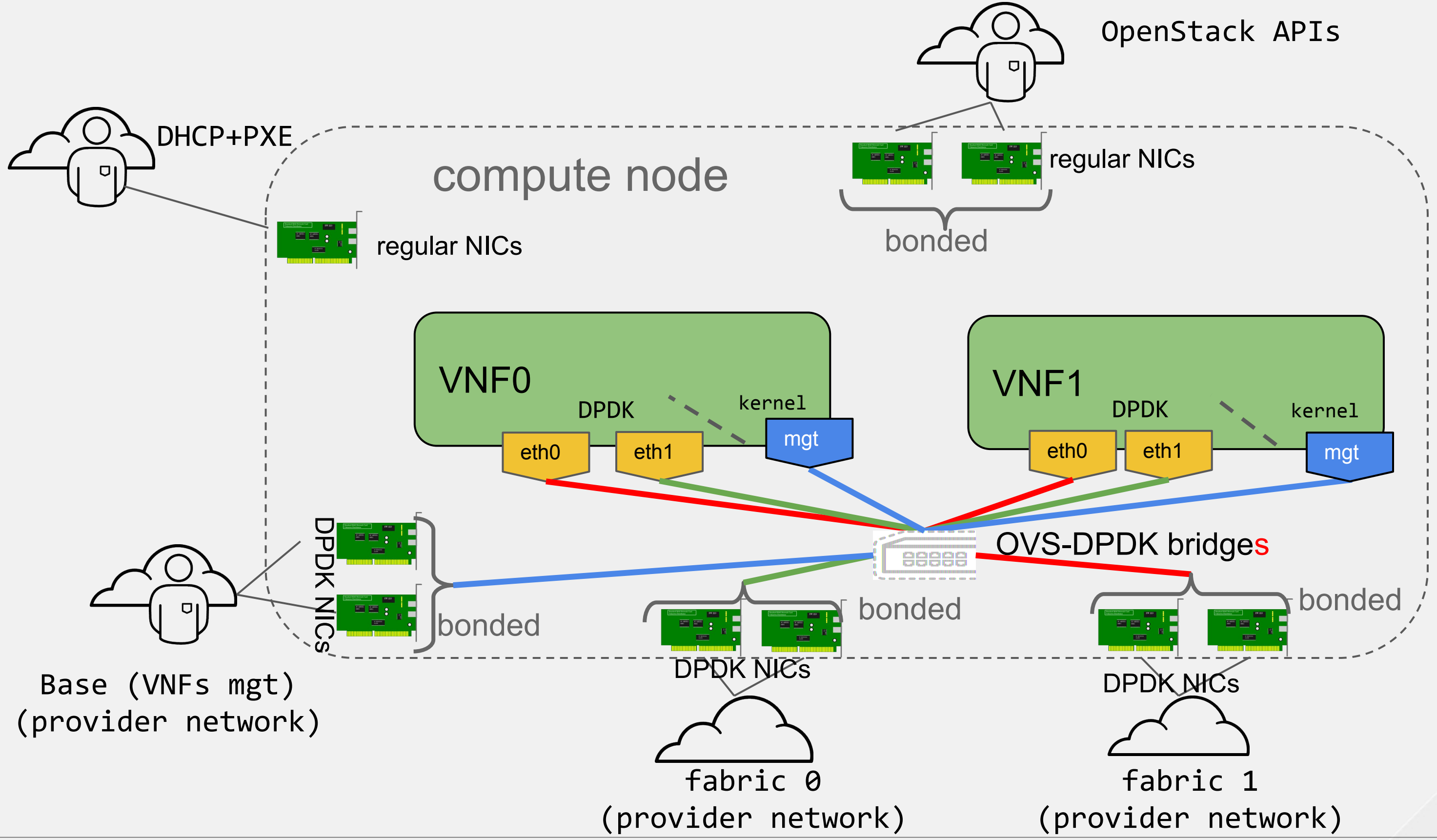
All host IRQs routed on host cores

All VNFx cores dedicated to VNFx

- Isolation from others VNFs
- Isolation from the host
- Isolation from Ceph Mon
- 21 Mpps/core with zero frame loss, 12 hours run (I/O bound due to Niantic PCIe x4 switch): equal to bare-metal performances
- The number above show that virtualization/SR-IOV cost is null, and that the VNF is not preempted/interrupted.



OVS-DPDK NFV deployment



RHOSP10 OVS-DPDK
Host/VNFs guests resources partitioning

Typical 18 cores per node dual socket compute node (E5-2599 v3)

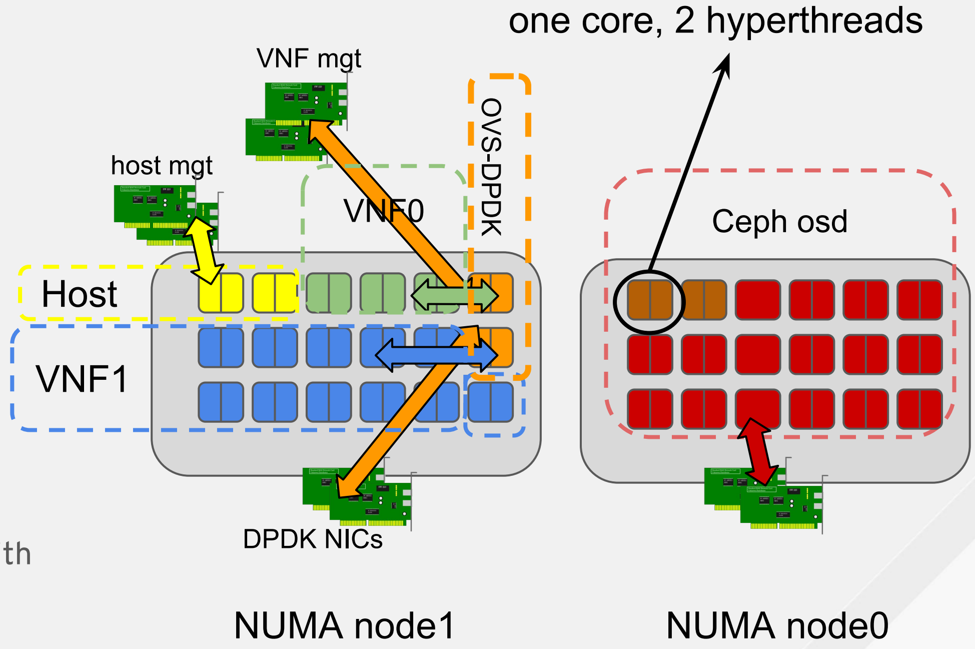
Same as SR-IOV, except of a 4th partition for OVS-DPDK

- CPUs list dedicated to OVS-DPDK
- Huge Pages reserved for OVS-DPDK

Not mixing VNF management traffic and Telco

Traffic requires additional NICs as NICs cannot be shared between OVS-DPDK and the host:

- 3.5 Mpps/core for PVP configuration (OVS-DPDK 2.5) with zero packet loss



Host/VNFs guests resources partitioning

How do we partition VNF resources?

- Tuned cpu-partitioning reserves CPUs for exclusive use
  - Assign “isolated_cores” in /etc/tuned/cpu-partitioning-variables.conf
- By default all user processes and most kernel threads are excluded from using the CPUs configured with this profile
- Applications must move threads to these CPUs in order to use them:
  - Openvswitch:
    - Thread assignment: ovs-vsctl set Open_vSwitch . other_config:pmd-cpu-mask=
  - Nova:
    - Configuration: vcpu_pin_set in /etc/nova/nova.conf
    - Usage: hw:cpu_policy=dedicated for flavor

Performance Testing HCI with NFV

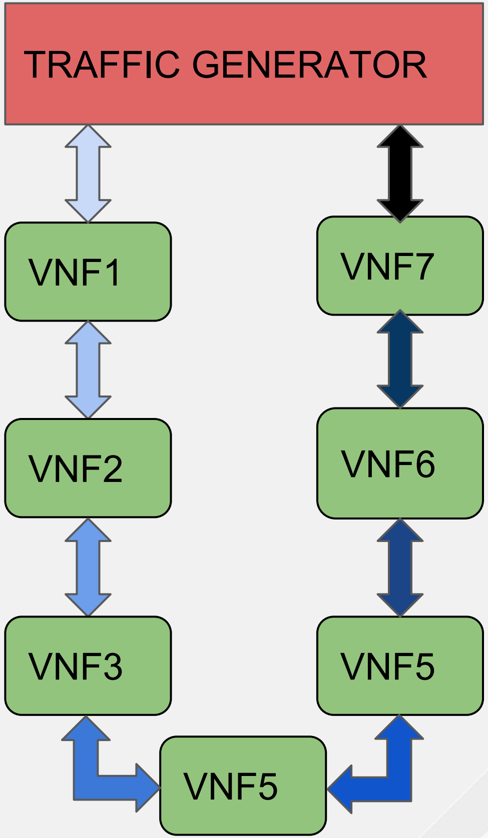
NFV performance is **highest priority**

- The network service of the VNF should never degrade
- VNFs should have adequate disk IO performance, even when at 100% network service utilization
- A test is conducted, having three phases:
  - Phase 1 has only VNF network activity in all VMs
  - Phase 2 has VNF and Disk activity in all VMs
  - Phase 3 returns to only network activity in the VMs
- During phase 2 there should be **no degrade** in VNF network performance

Performance Testing HCI with NFV

How do we test VNF performance?

- We have opted for a test which uses several VNFs across several compute nodes (1 per compute node)
- The VNFs are performing L3 routing (bidirectional) between two Openstack provider networks, using a DPDK application, VPP[1]
- VNFs are “chained” to form a series of routes that network traffic will traverse, and therefore, a degrade in throughput from any VNF will affect overall throughput
- The traffic generator transmits and receives on two interfaces (interface-1 Tx to interface-2, interface-2 Tx to interface-1)
- The traffic generator transmits 64-byte frames at sustained rate equal to maximum capability of the VNFs, in this case, 5.5 million packets per second per interface, for a total of 11 Mpps

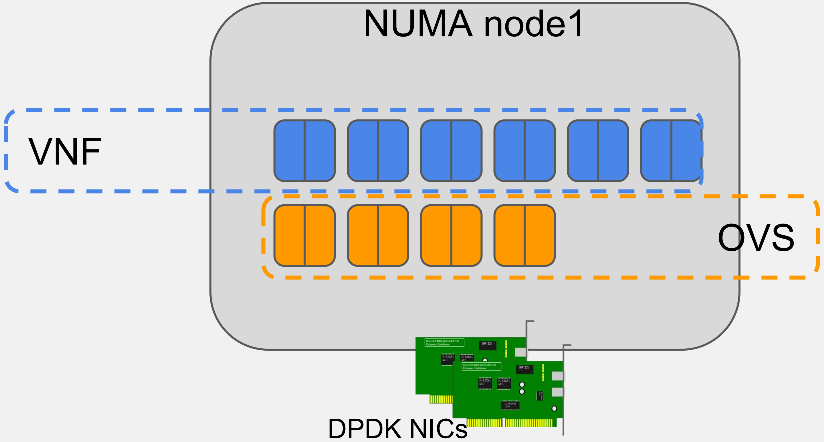


[1] https://wiki.fd.io/view/VPP

Performance Testing HCI with NFV

Configuration Details for NFV: compute node

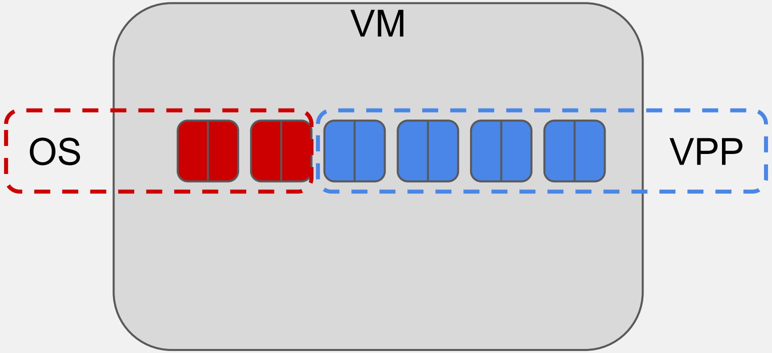
- Ceph on node0, NFV on node1
- Openvswitch with DPDK:
  - Enable 2-queue per device: options:n_rxq=2
  - Use 8 PMD threads on dedicated CPUs
- VM:
  - Dedicated 6 cores / 12 CPU-threads
  - Enable multi-queue for virtio-net: hw_vif_multiqueue_enabled=true



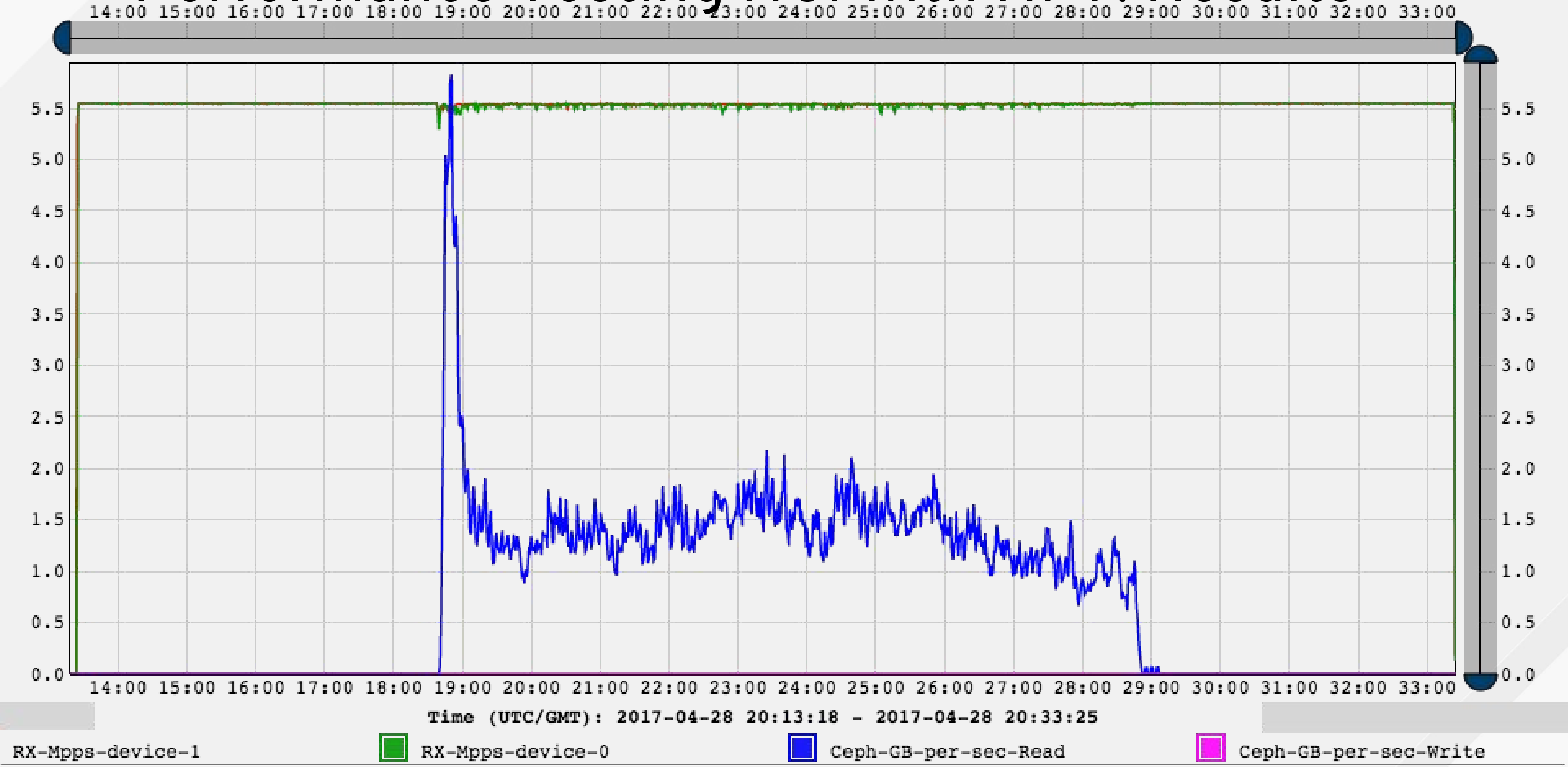
Performance Testing HCI with NFV

Configuration Details for NFV: VM

- Tuned cpu-partitioning used again, this time inside VM:
    - isolate last 4 cores for VPP
- VPP using 1 CPU-thread per core (from last 4 cores) for PMD threads
- First 2 cores used for OS, disk IO, and non-polling threads for VPP



Performance Testing HCI with NFV: Results



Performance Testing HCI with NFV: Results

Summary

- When disk I/O activity started, VNFs experienced a ~4% degrade in throughput momentarily, then experienced ~2% variation in throughput, but on average maintained 99% of the throughput without disk I/O
- Each VM was reading an average ~200MB/sec, lower I/O should reflect much lower impact to VNF
- Investigations into interactions between disk I/O and VNF effects underway @RH
- Our goal is to completely eliminate the disk I/O effects on the VNF performance

RED HAT
SUMMIT

THANK YOU

plus.google.com/+RedHat

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHatNews

RED HAT
SUMMIT

LEARN. NETWORK.
EXPERIENCE
OPEN SOURCE.

INTERACT | DISCUSS | DESIGN

ENABLE AGILITY WITH INFRASTRUCTURE AS CODE

Stephane Lefrere, Cloud Infrastructure Practice Lead
Julio Villarreal Pelegrino, Principal Infrastructure Architect

Tuesday, May 2, 4:30 PM - 5:15 PM

Located at the Consulting Discovery Zone at the Services Showcase in the Partner Pavilion

To learn more, **visit red.ht/discoveryzone**