Operating Neutron at Scale in HP Public Cloud

Jack McCann

Neutron tech lead, HP Public Cloud

May 15, 2014

HP Public Cloud

1st generation compute service

nova-network

FlatDHCP multi-host with HP extensions

2nd generation compute service

Neutron

Today

2011 2012 2013 2014

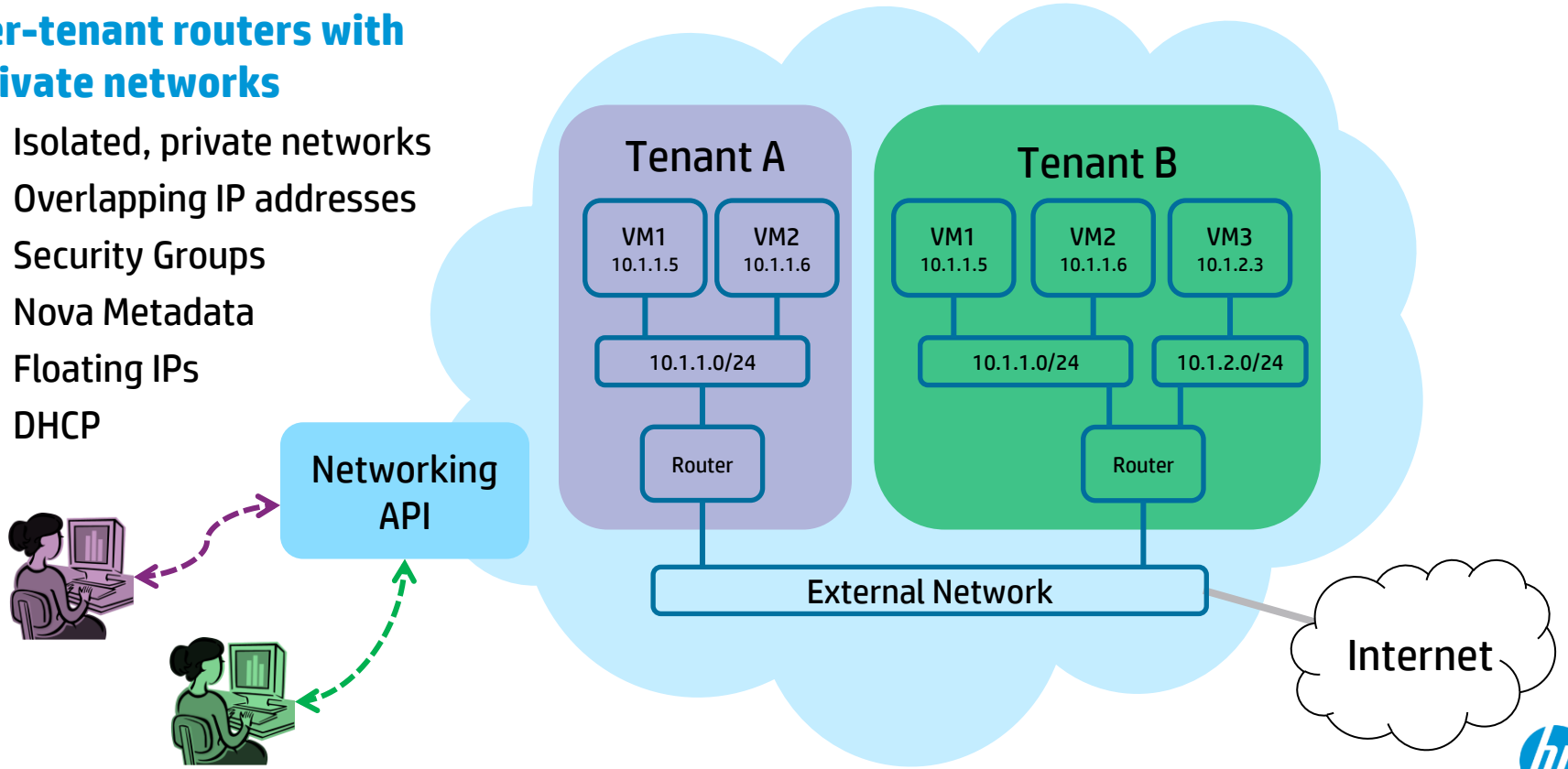
HP learnings and contributions to Neutron to improve stability, performance and scalability

Today's focus

Tenant-facing network model

Per-tenant routers with private networks

- Isolated, private networks
- Overlapping IP addresses
- Security Groups
- Nova Metadata
- Floating IPs
- DHCP

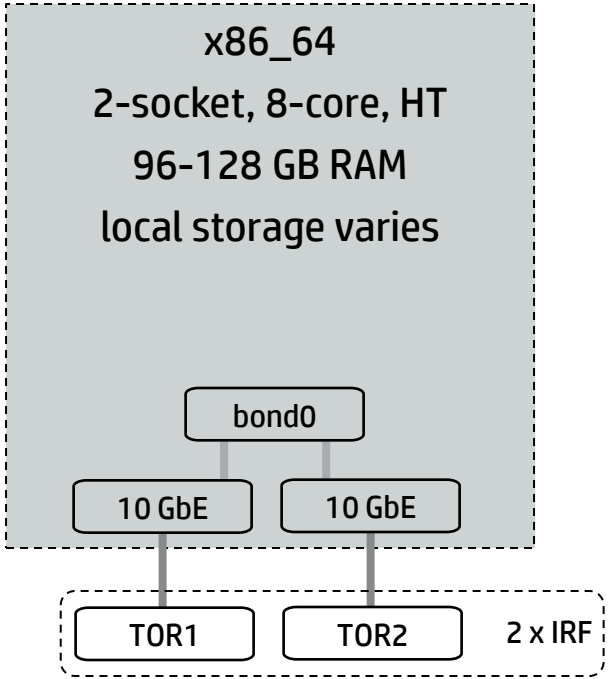


Typical server building block

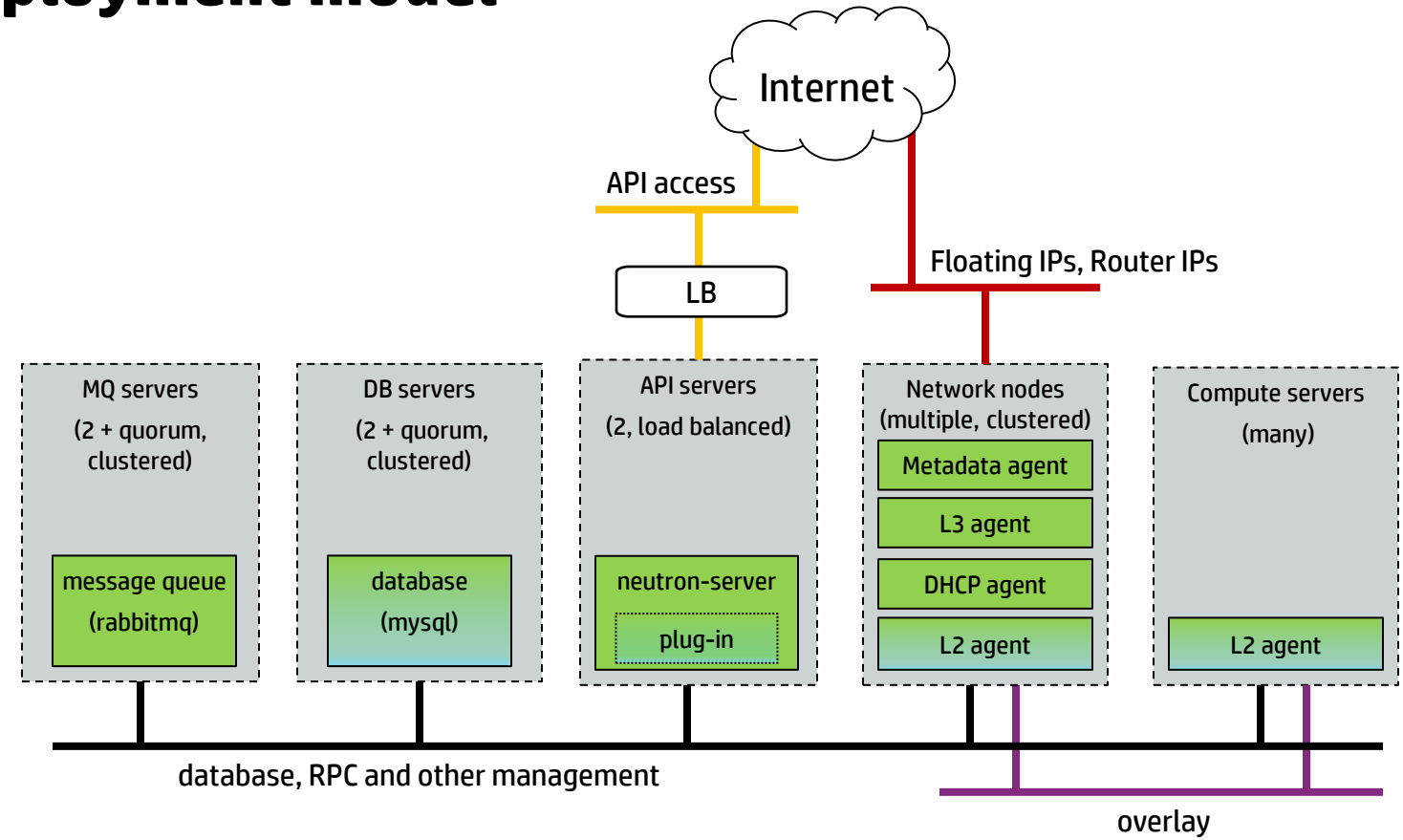
HP Servers

HP Networking 59xx series

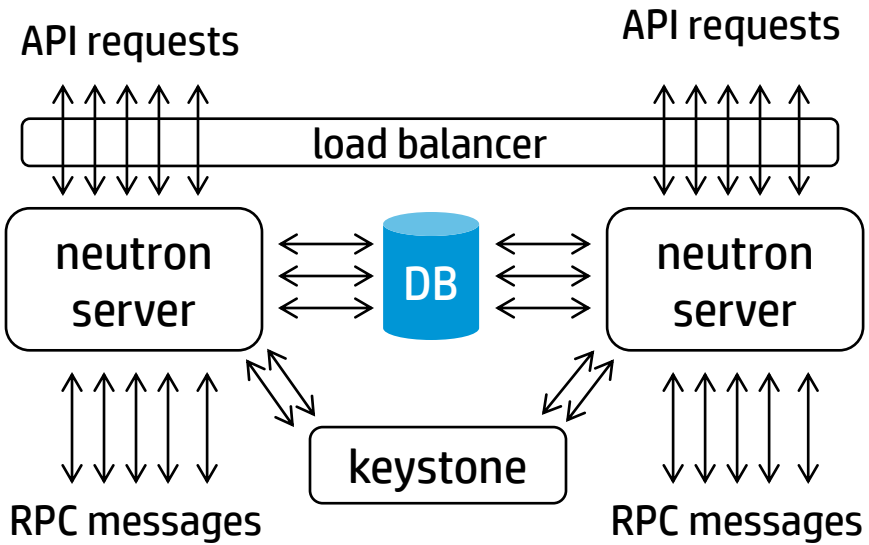
Intelligent Resilient Framework

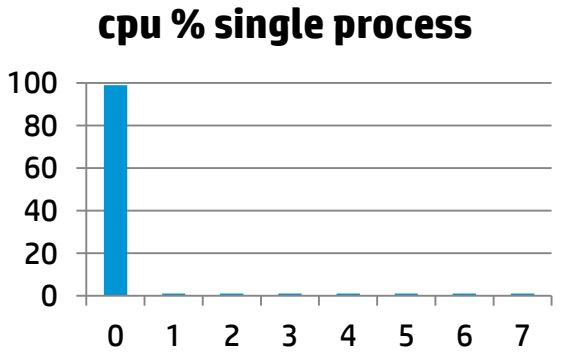


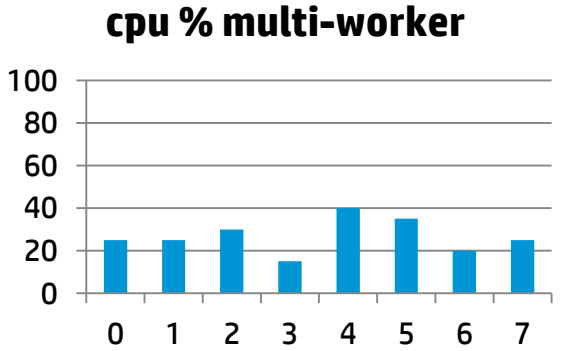
Deployment model



Neutron server







[Creates multiple worker processes for API server]

[Adds multiple RPC worker processes to neutron server]

neutron.conf

```
api_workers=4
rpc_workers=4
```

caveat: rpc_workers currently not compatible with qpid, zeromq

First you need an IP address

I can't ping my VM!

VM console log

```
        :
cloud-init-nonet gave up waiting for a network device.
ci-info: lo    : 1 127.0.0.1      255.0.0.0
ci-info: eth0  : 1                        fa:16:3e:d9:cb:2a
route_info failed
        :
```

DHCP agent improvements

Fixed in Havana:

Adds default route to DHCP namespace for upstream name resolution

- allow the default DNS server (dnsmasq) to reach other DNS servers outside the cloud

Fixed in Icehouse:

Dhcp agent sync_state may block or delay configuration of new networks.

- a case of one bad apple spoiling the bunch, sync_state would never get past a bad network

Use information from the dnsmasq hosts file to call dhcp_release

- keep dnsmasq in sync with agent cache

Change to improve dhcp-agent sync_state

- make sure one sync_state completes before another begins

Remove unnecessary call to get_dhcp_port from DeviceManager

- dhcp agent was making thousands of unnecessary RPC calls per day

Coming in Juno:

Provide way to reserve dhcp port during failovers

- if you move a network from one dhcp agent to another, its IP address changes, breaking default DNS server

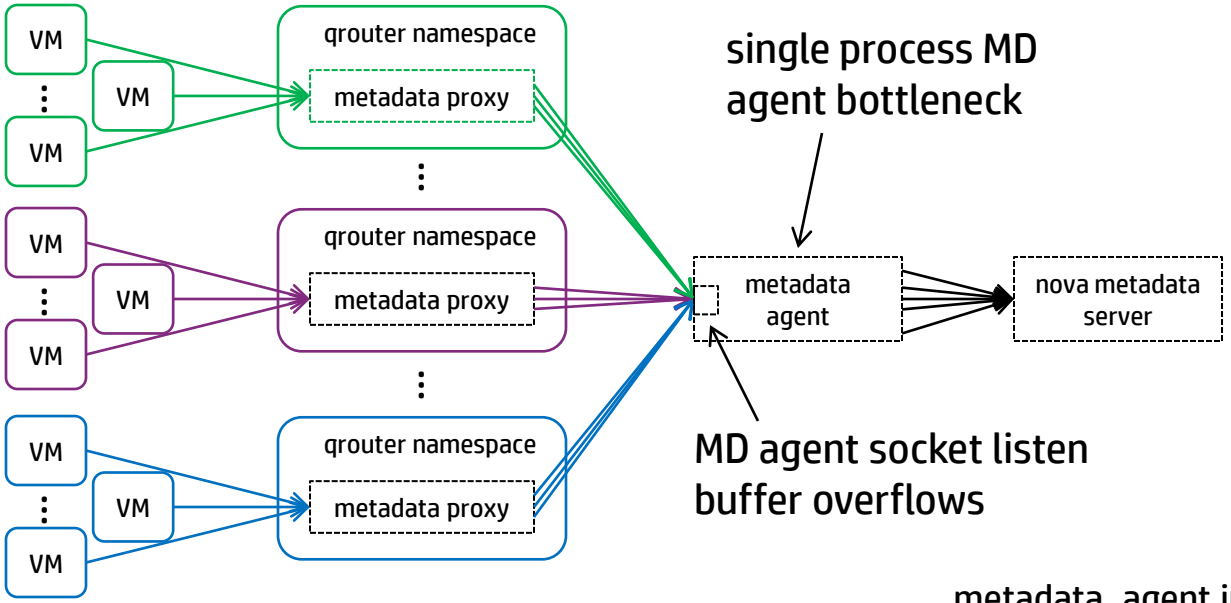
Got an IP, now you want your metadata

I can't ssh to my VM!

VM console log

```
        :
DataSourceEc2.py[WARNING]: 'http://169.254.169.254' failed: socket timeout [timed out]
DataSourceEc2.py[WARNING]: 'http://169.254.169.254' failed: socket timeout [timed out]
DataSourceEc2.py[CRITICAL]: giving up on md after 195 seconds
        :
```

Metadata agent



[Change metadata-agent to spawn multiple workers]

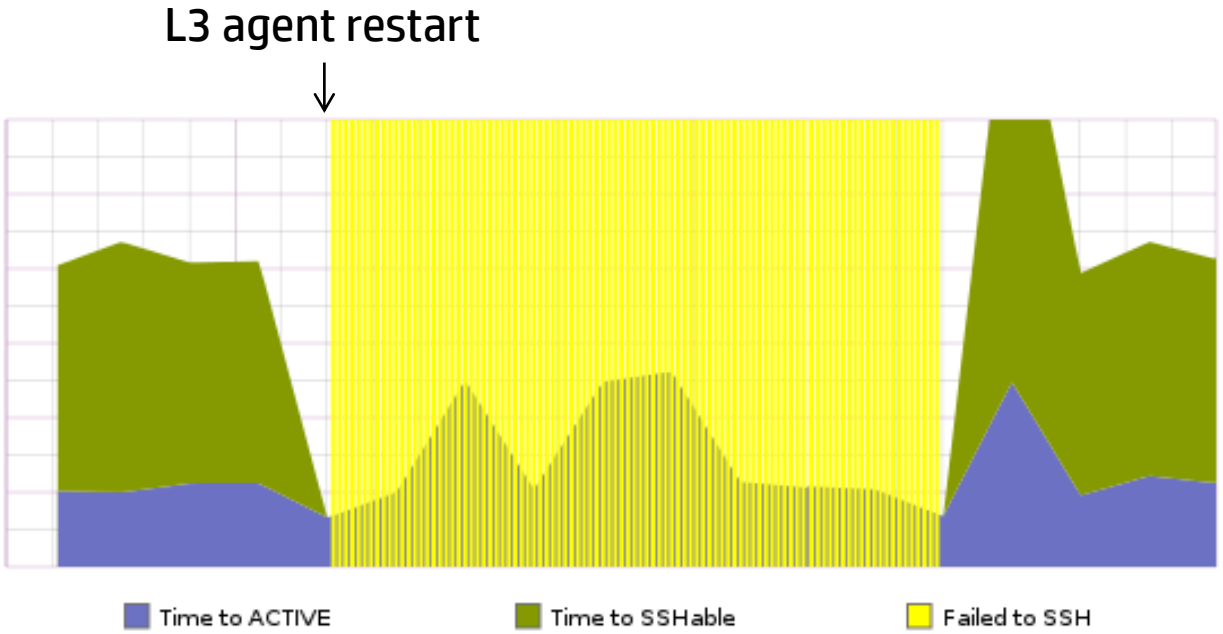
[Change metadata-agent to have a configurable backlog]

metadata_agent.ini

```
metadata_workers=10
metadata_backlog=2048
```

Got my IP, got my metadata, good to go?

I can't reach my VM!



L3 agent improvements

Fixed in Icehouse:

L3 Agent restart causes network outage

- l3 agent restart would tear down all network namespaces and rebuild them!

Preserve floating ips when initializing l3 gateway interface

- l3 agent restart would remove then add floating IPs causing temporary network outage

Spawn arping in thread to speed-up floating IP

- 'arping –c 3' takes 3 seconds per floating IP, do that in parallel rather than serialized

Use an independent iptables lock per namespace

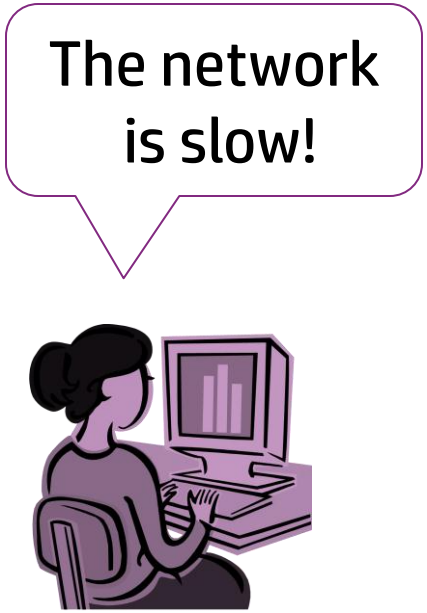
- agent can process multiple routers in parallel threads, but was serialized on iptables lock

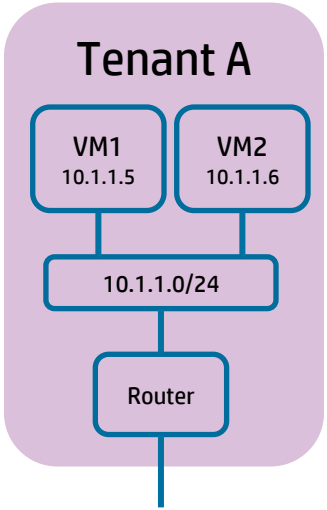
Coming in Juno:

L3 agent prefers RPC messages over full sync

- new work (e.g. floating IP add/delete) is delayed while agent runs a full sync

Got my IP, got my metadata, good to go?





Kernel – version matters

We ran into several kernel performance and scaling issues along the way:

- panic during network namespace deletion (you really want to be able to delete namespaces)
- veth pairs found to be a network throughput bottleneck
- increasing number of network namespaces decreased ns entry/exit performance
- conntrack table full with UNREPLIED entries

The good news is each issue had already been addressed in newer kernels

issue	commit	kernel
namespace delete panic	665e205c16c1f902ac6763b8ce8a0a3a1dcefe59 32263dd1b43378b4f7d7796ed713f77e95f27e8a	3.8
veth performance	2681128f0ced8aa4e66f221197e183cc16d244fe 8093315a91340bca52549044975d8c7f673b28a1	3.9
network namespace performance	84d17192d2afd52aeba88c71ae4959a015f56a38	3.10
conntrack table full with UNREPLIED	6547a221871f139cc56328a38105d47c14874cbe	3.11

Commands - version matters

Problem:

As the number of network interfaces on a system increases, ‘sudo’ slows down.

With large numbers of network interfaces, the slowdown was considerable.

Solution:

sudo version 1.8.10 adds the ability to disable network interface probing

In sudo.conf: Set probe_interfaces false

Before

```
$ time sudo sleep 1
real    0m1.333s
user    0m0.036s
sys     0m0.292s
```

After

```
$ time sudo sleep 1
real    0m1.005s
user    0m0.004s
sys     0m0.000s
```

Got IP, metadata, and good network performance

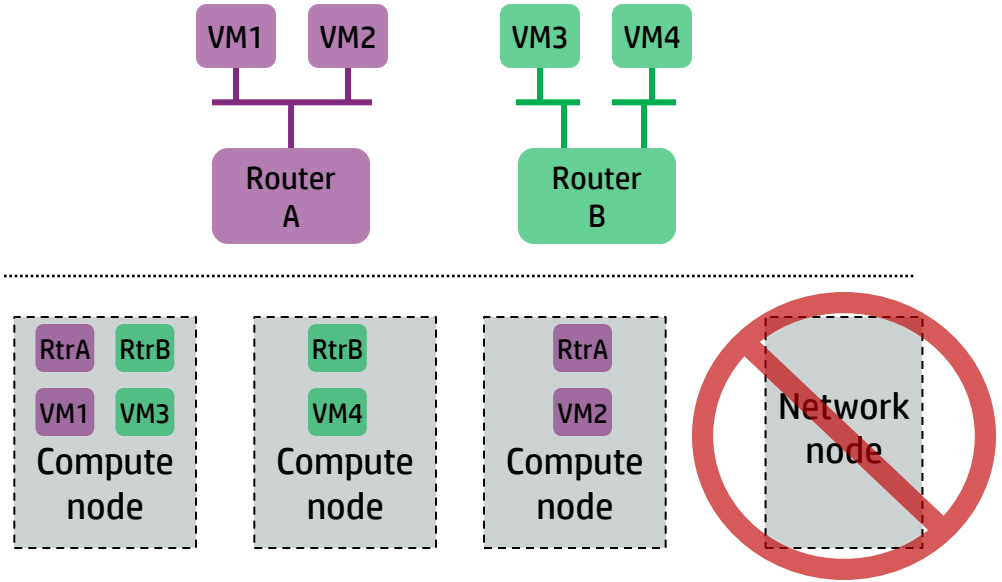
A look forward to DVR

2009 **HP Labs Diverter technology**

2011 **1st gen HP Public Cloud multi-host extensions**

Neutron Distributed Virtual Router for OVS

2014 **Juno**



Each compute node provides routing for local VMs.

Inter-subnet traffic flows directly between compute nodes.

Floating IP traffic flows directly to VM's compute node.

Summary and conclusions

Upgrade neutron (Icehouse is better than Havana is better than Grizzly…)

Make sure your neutron server is properly provisioned and tuned

Make sure the metadata agent is properly tuned

Upgrade your kernel (newer is generally better)

Make sure sudo is properly versioned and tuned

Expect improved stability, performance and scalability in Juno

Thank You

www.hpcloud.com