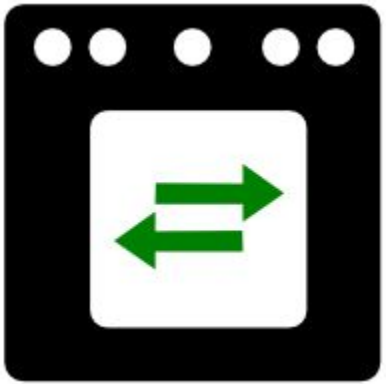
OVN:

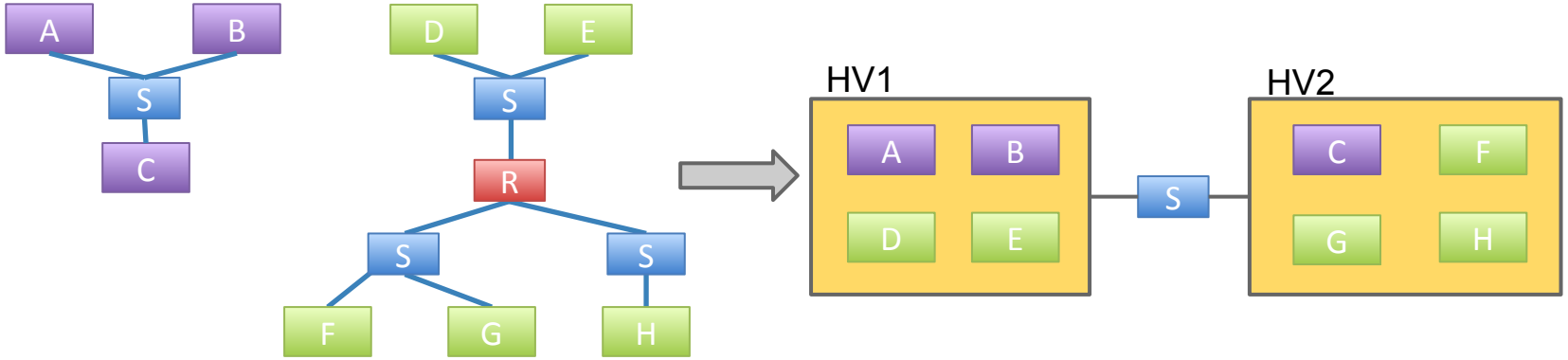
Scaleable Virtual Networking for Open vSwitch

Kyle Mestery (@mestery)
Justin Pettit (@Justin_D_Pettit)



The Case for Network Virtualization

- Network provisioning needs to be self-service.
- Virtual networking needs to be abstracted from physical.
- Virtual networking needs same features as physical.



Legacy Physical Network

Cloud Physical Network

What is OVN?

- Open source L2/L3 network virtualization for Open vSwitch (OVS):
  - ✓ Logical switches
  - ✓ IPv4 and IPv6 logical routers
  - ✓ L2/L3/L4 ACLs (Security Groups)
  - ✓ Multiple tunnel overlays (Geneve, STT, and VXLAN)
  - ✓ Logical load-balancing
  - ✓ TOR-based L2 logical-physical gateways
  - ✓ Software-based L2/L3 logical-physical gateways
- Works on same platforms as OVS:
  - ✓ Linux
  - ✓ Containers
  - ✓ DPDK
- Integration with:
  - ✓ OpenStack Neutron
  - ✓ Docker Swarm
  - ✓ Kubernetes

The Particulars

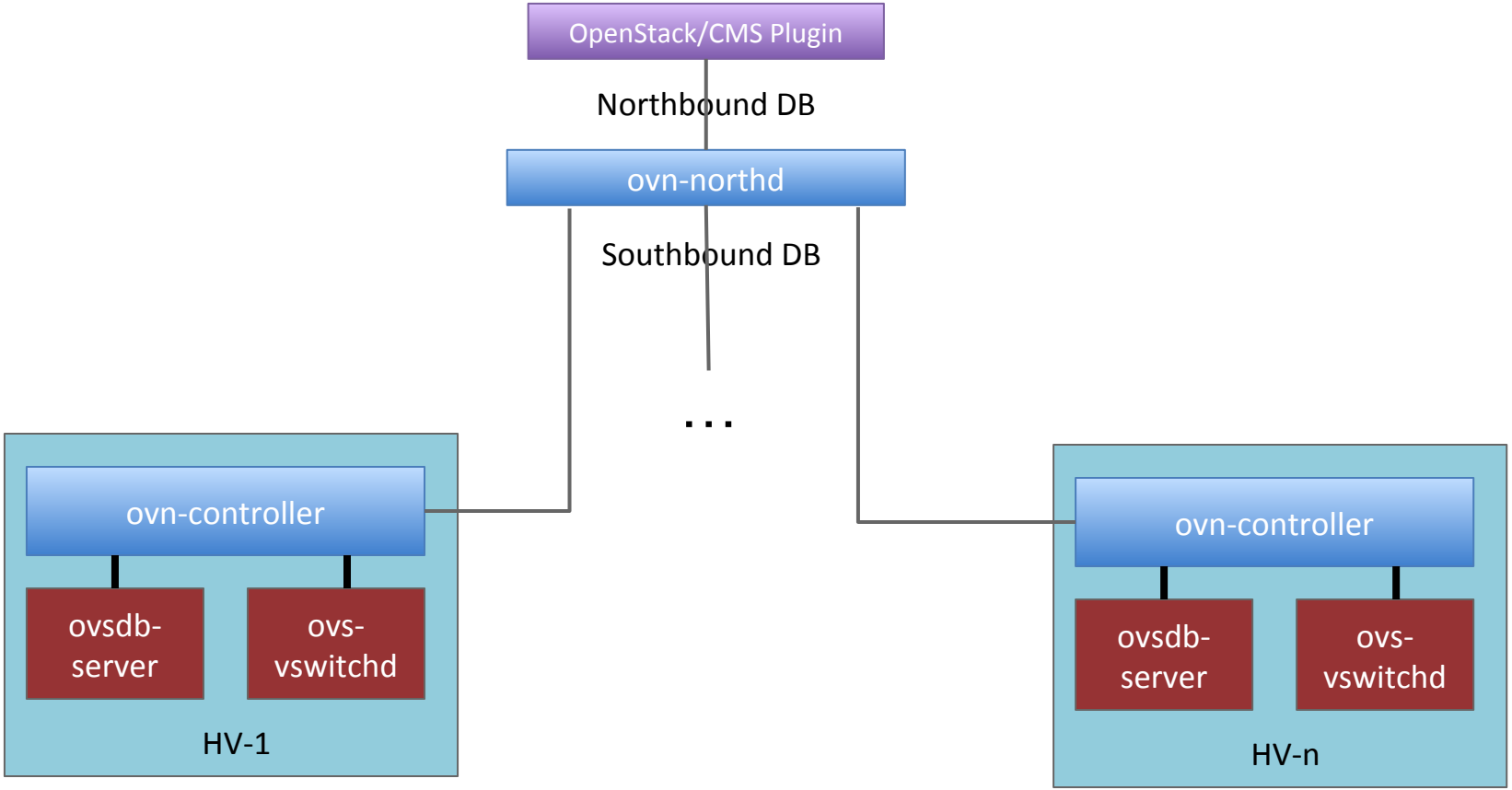
- Developed by the same community as Open vSwitch
- Vendor-neutral
- Design and implementation all occur in public
- Developed under the Apache license

Goals

- Production-quality
- Straightforward design
- Scale to 1000s of hypervisors (each with many VMs/containers)
- Scale to 100s of thousands of ports

How is OVN Different?

OVN Architecture



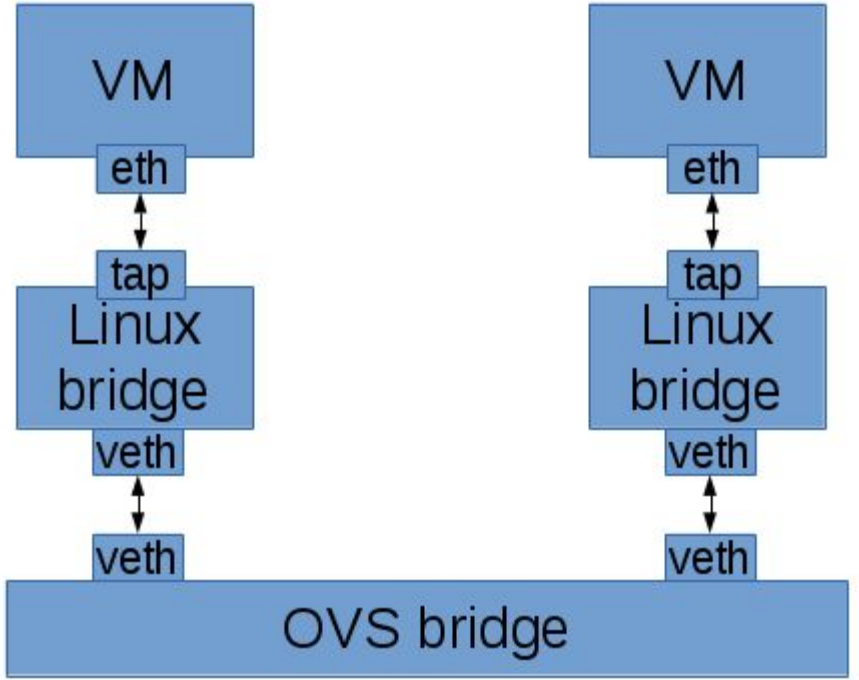
Architecture

- Configuration coordinated through databases
- Logical flows, don’t worry about physical topology
- Local controller converts logical flow state into physical flow state
- Desired state clearly separated from run-time state
- Based on the architecture we wanted from seeing a number of others using OVS

Data Plane Scale

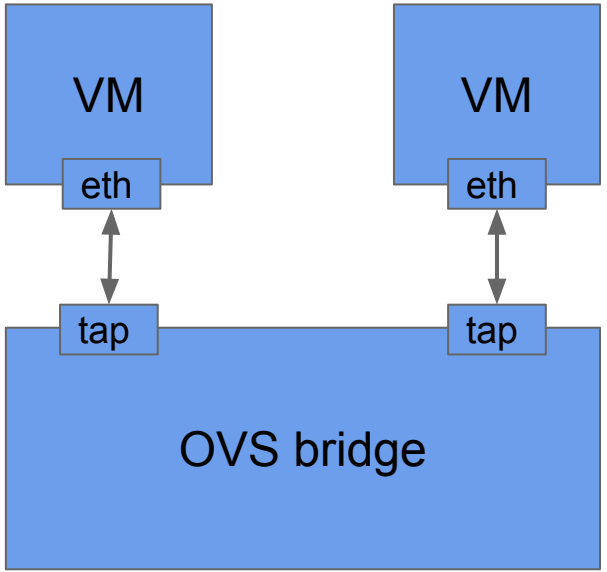
Common Approach to Security Groups

- OpenFlow
  - Not truly stateful
  - Possibly bad performance
- OpenStack
  - Required extra linux bridge and veth pair **per VM**
  - Uses iptables



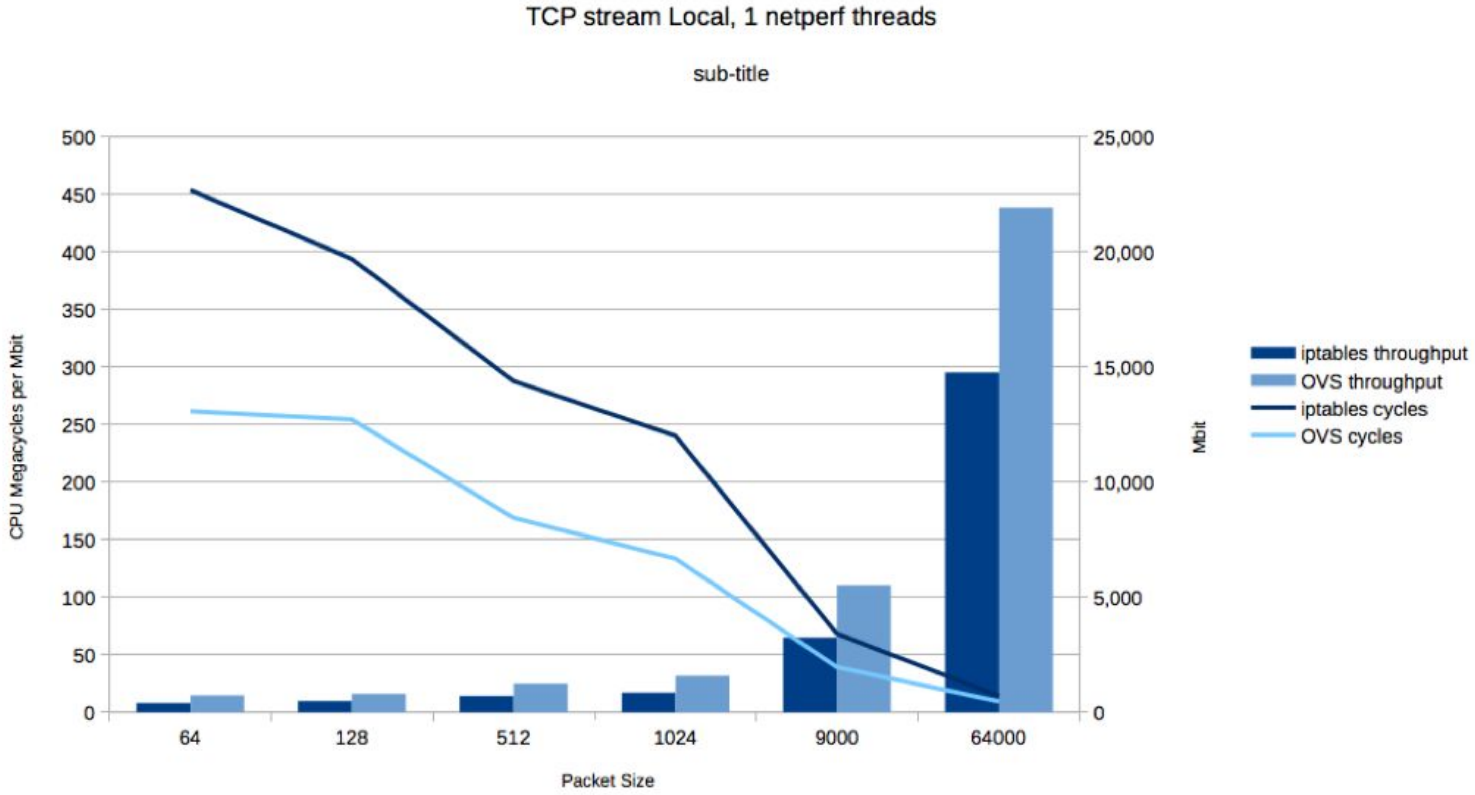
OVN Security Groups Design

- Uses kernel conntrack module directly from OVS
- Design benefits
  - No complicated pipeline
  - Faster* -- Fewer hops and veth ports



* http://blog.russellbryant.net/2015/10/22/openstack-security-groups-using-ovn-acls/

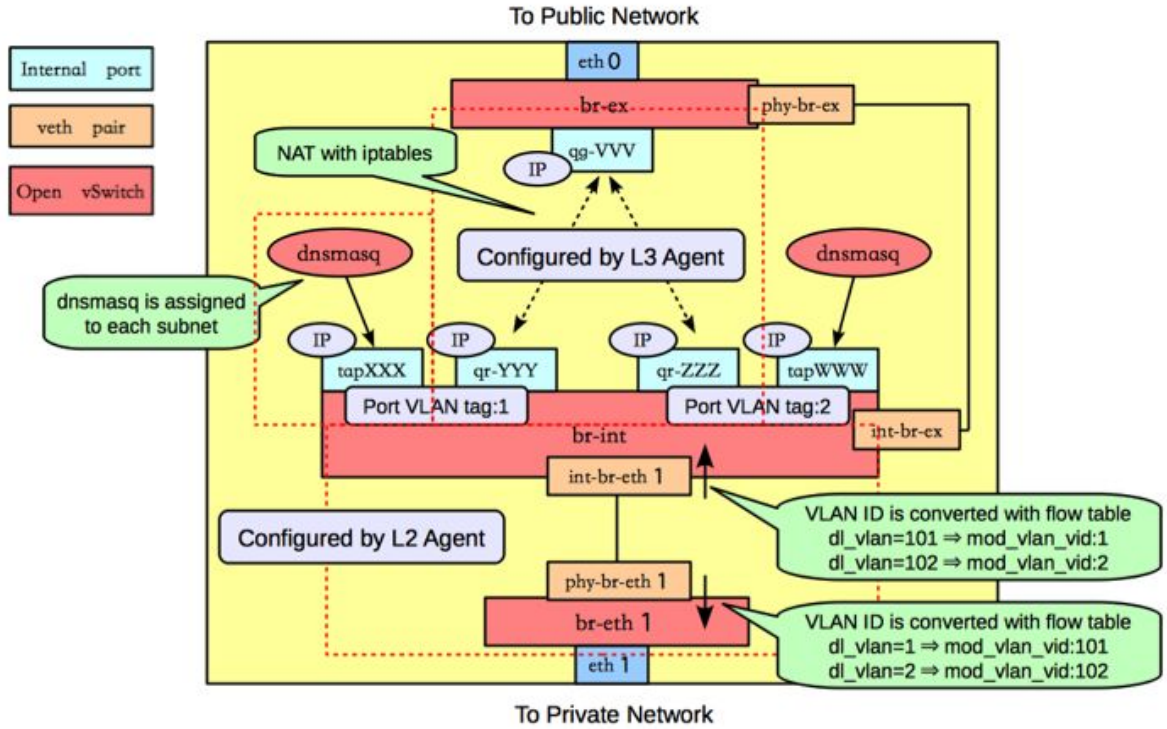
Security Group Throughput



Common Approach to L3

- Agent-based
- Use the Linux IP stack and iptables
  - Forwarding
  - NAT
- Overlapping IP address support using namespaces

Example OpenStack L3



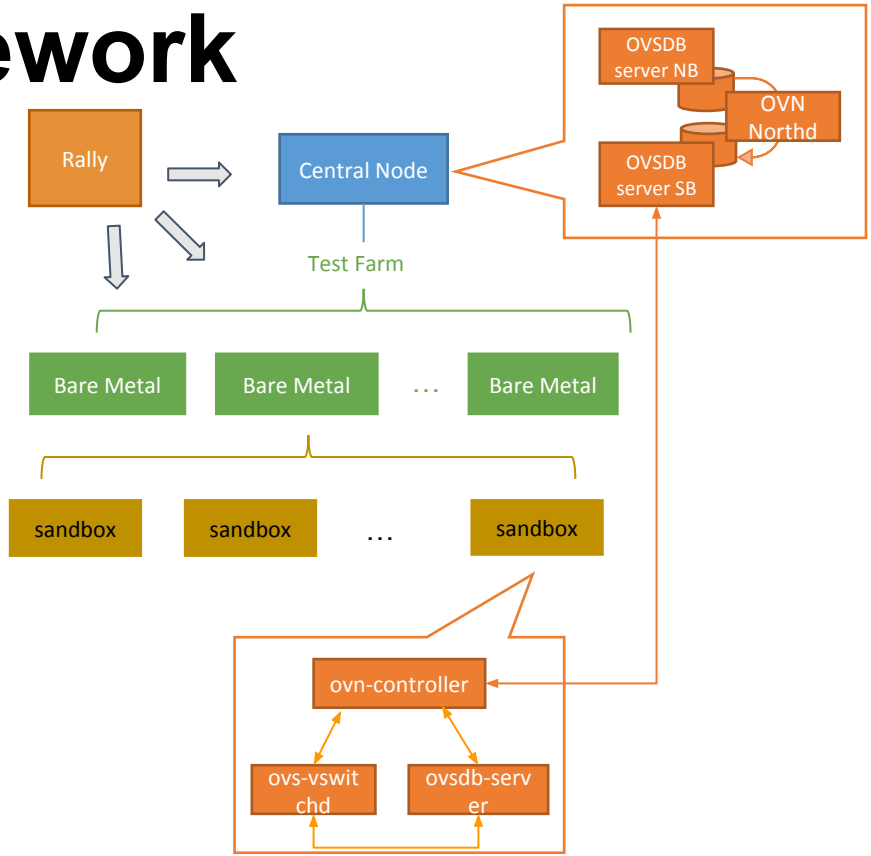
OVN L3 Design

- Native support for IPv4 and IPv6
- Distributed
- ARP/ND suppression
- Flow caching improves performance
  - Without OVN: multiple per-packet routing layers
  - With OVN: cache sets dest mac, decrements TTL
- No CMS-specific L3 agent

Control Plane Scale

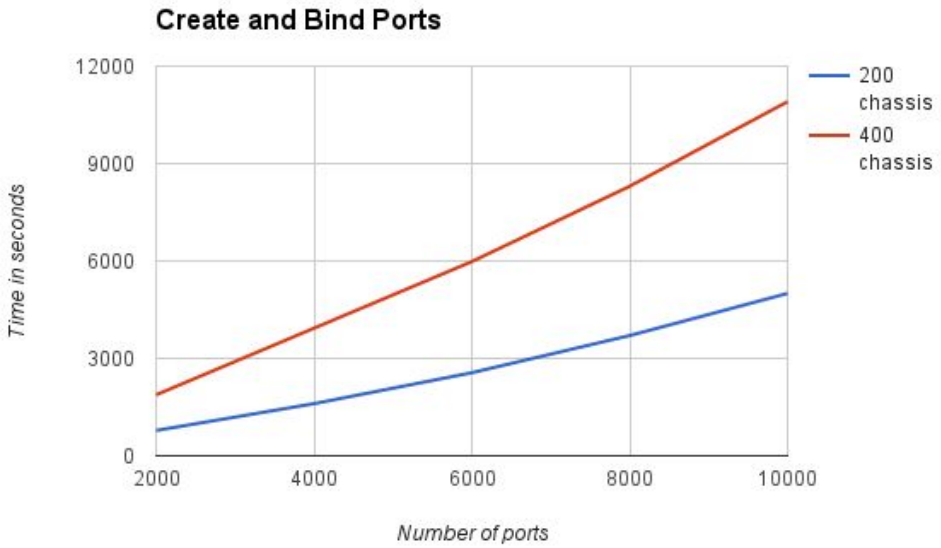
Scale Test Framework

- Scalability test for OVN control-plane
- Simulate an entire OVN deployment
  - Use Rally for deployment and test automation
- TODO:
  - Neutron integration
  - L3 test
  - Non-Rally test cases
- Contributions welcome! ☺
  - https://github.com/openvswitch/ovn-scale-test.git



Current Scale (Pure OVN)

- ovn-scale-test framework
  - 400 and 200 emulated chassis tests
  - 1 single network
  - 1 ACL/port
  - Creating and binding ports in increments of 2k
- NOTE:
  - OVN components ran on 2 physical hosts (48 threads and 256GB RAM)



Scale Improvements - Ongoing

- ovn-controller
  - Incremental Computation
  - Conditional Monitoring
- ovn-northd
  - Incremental Computation
- OVSDB
  - Evaluation of an alternative database

Deployment

Deployment made easy

- No additional daemons to install on hypervisors beyond what comes with OVS
- Minimal host-level configuration
- Rolling upgrades

Rolling Upgrades

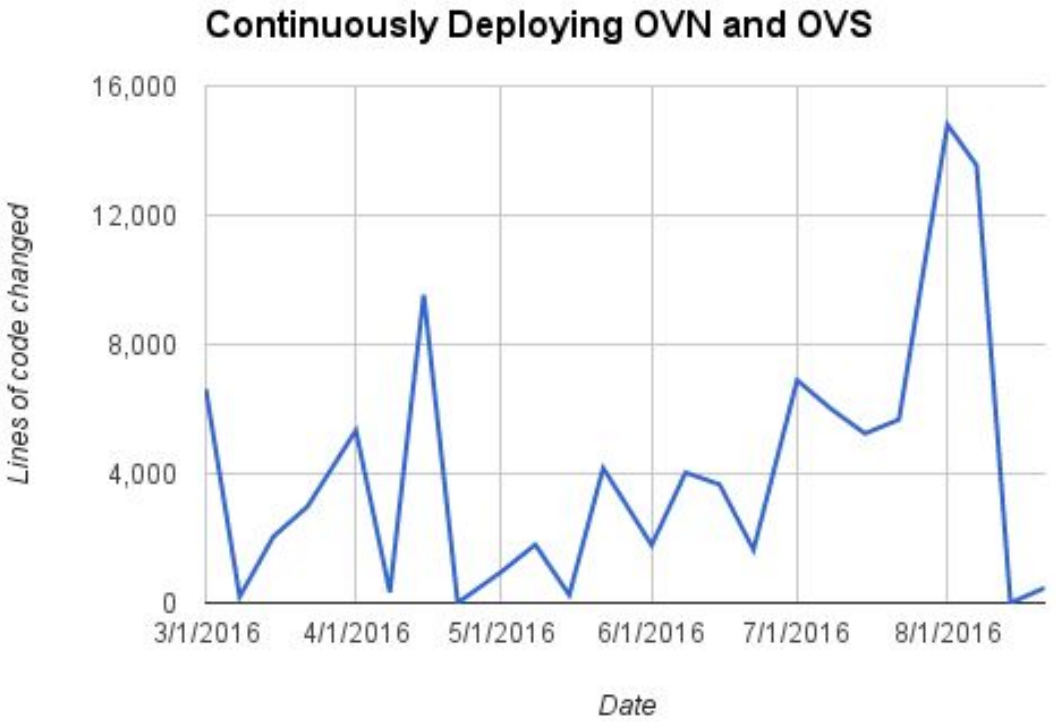
- OVSDB schema is versioned
- Changes to schema will be carefully managed to be backwards compatible
- Allows rolling upgrades
  - Update databases first
  - Roll through upgrades to ovn-controller
- Same strategy OVS itself has been using

Continuously Delivering OVN

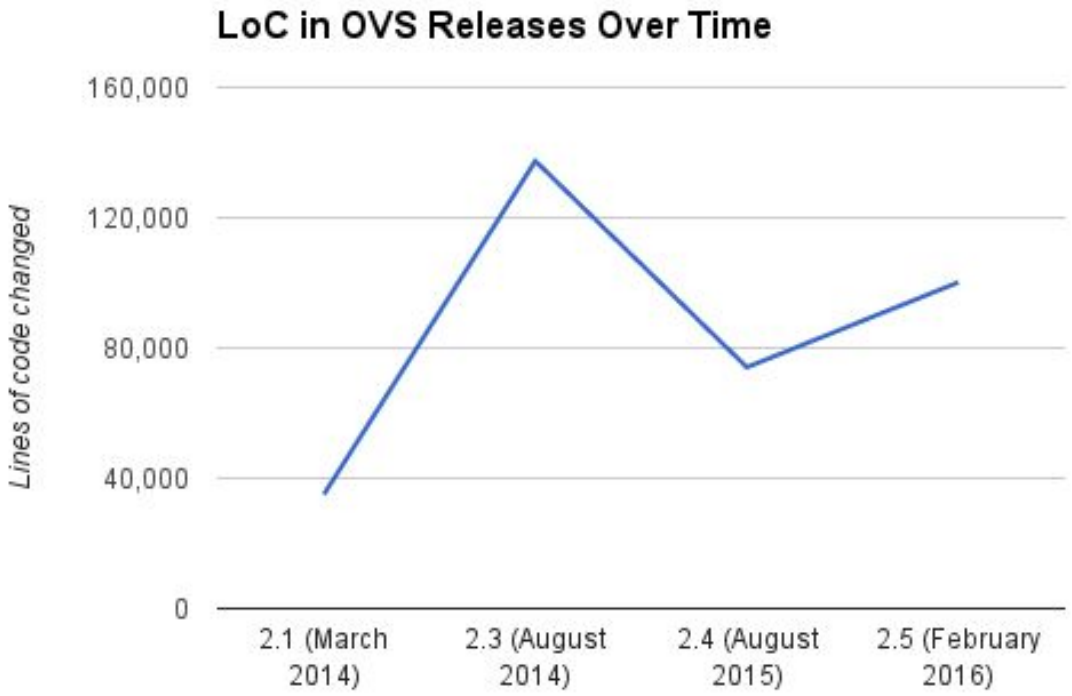
Why Continuous Delivery of OVN?

- 90+ active developers working on OVS/OVN
- Hundreds to thousands of lines of code added daily - travis-ci jobs running to test this
- At large scale, automated testing is a given
- Delivering upstream fast means developers can work upstream, reducing technical debt

Continuous Delivery of OVS/OVN



What About Delivering Releases?



One Way To Continuously Deliver

- Align with OpenStack CI/CD
  - Same tools upstream
    - Zuul (Pipeline management)
    - Nodepool (resource management)
    - Gerrit (code review)
  - Build our own packages
- Ability to carry local patches
  - Needed for security patches
  - Also for bugs and features not landed upstream yet

Status

Neutron Integration Status

- http://docs.openstack.org/developer/networking-ovn/features.html
- Neutron plugin supports
  - L2 networks
  - Provider Networks
  - Security Groups
  - QoS API
  - Native DHCP
  - Linux Kernel or DPDK datapaths
  - binding:profile for containers in VMs without another overlay
  - binding:profile for connecting vtep gateways to Neutron networks
- Can use OVN native L3 or Neutron L3 agent

OVN vs. OVS Python Agents

- Improved performance and stability over existing OpenStack OVS plugin
  - No more RabbitMQ usage for Neutron!
  - Uses OVSDB in place of RabbitMQ
- Become preferred method for OpenStack+OVS integration for the majority of use cases

OpenStack Deployment Options

- Full devstack support
- Puppet OpenStack now supports OVN
- TripleO support posted for review
- Kolla support being planned

Upcoming Release

- Non-experimental for next OpenStack release (Newton)
- Recently landed features:
  - L3 gateway with NAT and load-balancing support
  - IPv6 logical routing
  - Native DHCP service
  - Address Set for ACL/Security group
  - Kubernetes support

The “Microwave” Release

Future Work

- Better database clustering and HA
- Avoid complete recalculations with incremental computation
- Native DNS support
- Live migration support for ACLs
- Hitless upgrades

Resources

- Architecture described in detail in ovn-architecture (5)
- Available in the “master” and “branch-2.6” branches of the main OVS repo:
  - https://github.com/openvswitch/ovs
  - http://openvswitch.org/support/dist-docs/
- Neutron plugin:
  - https://git.openstack.org/openstack/networking-ovn.git
- Neutron integration docs, including devstack instructions:
  - http://docs.openstack.org/developer/networking-ovn/
- Kubernetes plugin and documentation:
  - https://github.com/openvswitch/ovn-kubernetes
- OVN scale test harness
  - https://github.com/openvswitch/ovn-scale-test.git

How you can help

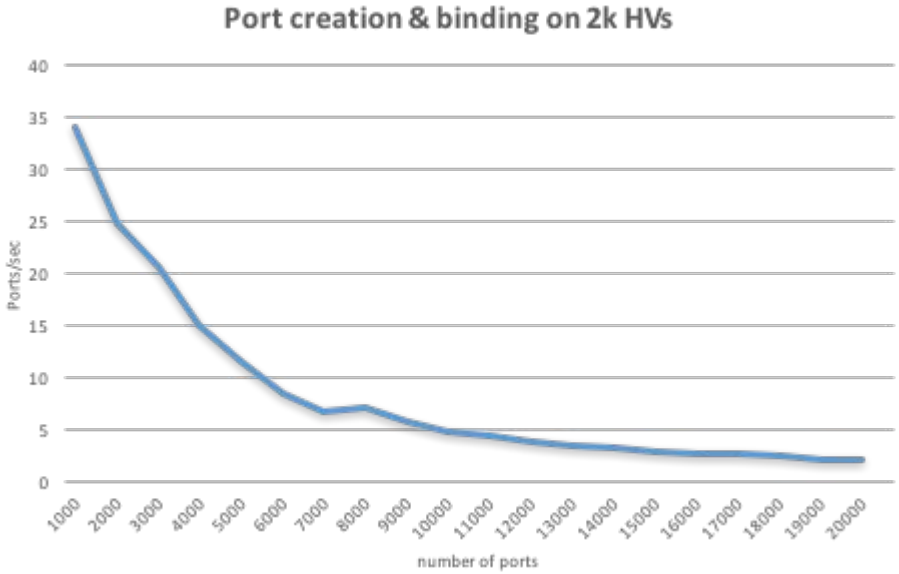
- Try it! Test it! Scale it! Report bugs! Write Code!
- Core OVN is being developed on ovs-dev mailing list:
  - http://openvswitch.org/pipermail/dev/
  - #openvswitch on Freenode
- Neutron plugin for OVN is being developed here:
  - http://git.openstack.org/openstack/networking-ovn.git
  - openstack-dev mailing list
  - #openstack-neutron-ovn on Freenode

Thank you! Questions?

Justin Pettit (@Justin_D_Pettit)
Kyle Mestery (@mestery)

Current Scale (Pure OVN)

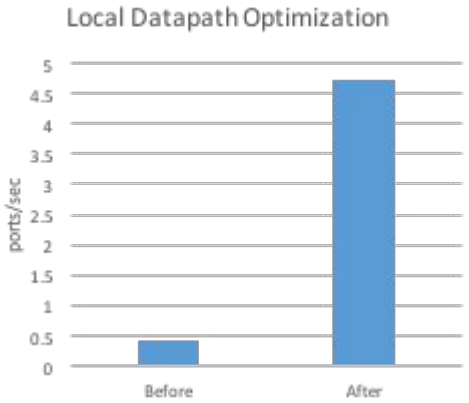
- L2
  - 2k HVs
  - 20k VIF ports (10 VIFs/HV)
  - 200 logical switches
    - Each lswitch spreading over 50 HVs
    - Each HV connected to 5 lswitches
- L3 – to be tested



@3k HVs, port create times becomes slow - improvements ongoing

Scale Improvements - Achieved

- Bottleneck 1: ovsdb north-bound memory leak fix
- Bottleneck 2: split ovsdb north-bound and south-bound into separate processes
- Bottleneck 3: ovsdb south-bound connections probe tuning
- Bottleneck 4: ovn-controller
  - Local datapath optimization
  - Micro optimizations on ovn-controller
    - Bit operations on logical flow processing
    - Dynamic memory optimization for lexer
    - Jemalloc
- Localnet improvement
  - Model change: reduced 50% # of logical ports



Current Scale (w/OpenStack)

15 HV Deployment:

> 250 routers and > 600 VMs

90 HV Deployment:

> 450 routers and >1500 VMs

400 HV Deployment:

1 provider network, 8k ports, 1 ACL/port

Neutron Plugin

- Speaks OVSDB to configure OVN via its Northbound database
- Goal: only run neutron API server, no agents
- No RabbitMQ, except for notifications (for Ceilometer, or a custom listener)