Practical OVN: Architecture, Deployment, and Scale of OpenStack Networking

Justin Pettit (@Justin_D_Pettit)
Ben Pfaff (@Ben_Pfaff)
Han Zhou (@zhouhanok)
Ryan Moats (@regxboi)
The Case for Network Virtualization

- Network provisioning needs to be self-service.
- Virtual network needs to be abstracted from physical.
- Virtual network needs same features as physical.
What is OVN?

- Open source L2/L3 network virtualization for Open vSwitch (OVS):
  - Logical switches
  - L2/L3/L4 ACLs (Security Groups)
  - Logical routers
  - Multiple tunnel overlays (Geneve, STT, and VXLAN)
  - TOR-based and software-based logical-physical gateways

- Works on same platforms as OVS:
  - Linux (KVM and Xen)
  - Containers (Docker)
  - DPDK
    - Hyper-V

- Integration with:
  - OpenStack Neutron
  - Other CMSes
The Particulars

- Developed by the same community as Open vSwitch
- Vendor-neutral
- Design and implementation all occur in public
- Developed under the Apache license
- Neutron plugin affiliation diversity in Mitaka release cycle
  - Top 5 reviewers from 5 companies
  - Top 4 committers from 4 companies
- Would qualify for OpenStack team:diverse-affiliation tag if it were an independent project (not part of Neutron)
Goals

• Production-quality
• Straightforward design
• Scale to 1000s of hypervisors (each with many VMs/containers)
• Improved performance and stability over existing OpenStack OVS plugin
• Become preferred method for OpenStack+OVS integration for the majority of use cases
Why Should OpenStack Care?

• Neutron’s primary job is to provide a cloud networking API abstraction
• OVN is a scope increase of OVS to implement many of the things Neutron needs
• If OVN succeeds in its mission, it reduces development burden on Neutron for OVS integration significantly
• Performance and scale improvements
How is OVN Different?
OVN Architecture

OVN Northbound DB

OVN Southbound DB

OVN-Controller

OVN-OVSDB-Server

OVN-OVS-vSwitchd

OpenStack/CMS Plugin
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 based on seeing a number of others using OVS
Data Plane Scale
Security Groups (The Original Way)

- Required extra `linux bridge` and `veth pair` per VM
- Uses `iptables`
Security Groups (OVN ACLs)

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

```plaintext
VM
  eth
 .tap

OVS bridge

VM
  eth
  .tap
```
Security Group Throughput

TCP stream Local, 1 netperf threads

Sub-title

CPU Megacycles per Mbit

Packet Size

Mbit

Iptables throughput
OVS throughput
Iptables cycles
OVS cycles
L3 (The Existing Way)

- Agent based
- Used the Linux IP stack and iptables
  - Forwarding
  - NAT
- Overlapping IP address support using namespaces
Current L3 Diagram
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 use of Neutron L3 agent
Control Plane Scale
Scale test framework

- Scalability test for OVN control-plane
- Simulated 2k HVs with 20 BMs
- Use Rally for deployment and test automation
- TODO:
  - Neutron integration
  - L3 test, ACL ...
- Contributions welcome! 😊
  - https://github.com/openvswitch/ovn-scale-test.git
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
Scale Improvements - Ongoing

- ovn-controller
  - Incremental Computation
  - Conditional Monitoring
- ovn-northd
  - Incremental Computation
- OVSDB
  - Multi-threading
- ACL
  - Address set
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)
Current Scale (w/OpenStack)

15 HV Deployment:
  > 250 routers and > 600 VMs

90 HV Deployment:
  > 450 routers and > 1500 VMs

Next step: 300 and 700 HV Deployments
Deployment
Deployment made easy

● No additional daemons to install on hypervisors beyond what comes with OVS
● Minimal host-level configuration
● Rolling upgrades
● Puppet OpenStack now supports OVN
● TripleO support posted for review
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
Status
Upcoming Release

• Production-ready for next OpenStack release (Newton)
• Features currently in development:
  • HA and multi-threading of ovsdb-server
  • L3 gateway with NAT support
  • IPv6 logical routing
  • Native DHCP and metadata proxy service
  • Address Set for ACL/Security group
  • Routed network support

The “Microwave” Release
Resources

• Architecture described in detail in ovn-architecture (5)
• Available in the “master” branch of the main OVS repo:
  – https://github.com/openvswitch/ovs
  – http://openvswitch.org/support/dist-docs/
• Neutron plugin in its own repo:
  – https://git.openstack.org/openstack/networking-ovn.git
• Neutron integration docs, including devstack instructions:
  – http://docs.openstack.org/developer/networking-ovn/
• 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:
  – #openvswitch on Freenode
• Neutron plugin for OVN is being developed here:
  – [http://git.openstack.org/openstack/networking-ovn.git](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)
Ben Pfaff (@Ben_Pfaff)
Han Zhou (@zhouhanok)
Ryan Moats (@regxboi)
Neutron Integration Status

• [http://docs.openstack.org/developer/networking-ovn/features.html](http://docs.openstack.org/developer/networking-ovn/features.html)

• Neutron plugin supports
  – L2 networks
  – Provider Networks
  – Security Groups
  – QoS API
  – 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