OVN Changes 2h-2021

Glenn West

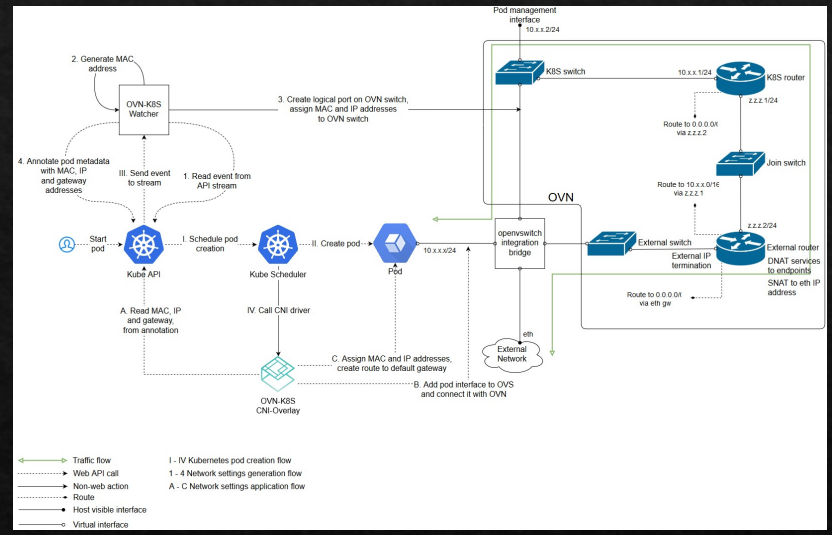
gwest@redhat.com

NorthD Results

- Logical Flow count on 120 node test database from 1.5m to 50k
- Southbound DB no longer a bottleneck @ 120 nodes
- 30% decrease in northd iteration times

Enhancement and Fixes

- 15 major Enhancements to NorthD
- 10 major enhancements to Database
- 10 major enhancements to ovn-controller
- Major Database Architecture change to Database



NorthD – ARP responding for LB Vips

- OVN builds too many lflows for ARP responding for load balancer VIPs
- In Test cluster with 250 nodes with ovn2.13-20
  - 850K flows are reduced to 350k flows
  - Reduced Database from 500MB to 200MB
- BZ 1945415

NorthD – refactor unreachable Ips lb flows

- Refactor code for unreachable IPS for vip load balancers by inverting the logic during the lb flow creation.
- Reduce the ovn-nortd loop time of +/- 3 seconds in a production environment
- BZ 1988554

NorthD – Use Distributed Gateway Port for ovn-controller scalability

- The following changes each show significant performance and scalability improvements:
  - Remove calculation of ha_ref_chassis
  - Removes unnecessary get_local_datapath
  - Don’t Flood fill local datapaths beyond DGP boundary
    - 90% reduction of CPU and Memory
- Multiple upstream patches

Northd: Rework and optimize reconciliation of datapath groups

- Remove the creation of a new identical datapath group. For a new flow.
- Don't check datapath groups in full if not needed – Saves multiple seconds in production size system for this loop
- Multiple upstream patches

NorthD – process load balancer defrag flows once for all routers

- Change to creating the matched strings for each LB VIP exactly once, instead of once per datapath, reducing CPU usage in the event processing loop.
- In a 120 node system this reduces event processing loop time in northd from 9.5 seconds to 8.5 seconds.

Northd - Don't merge ARP flood flows for all (unreachable) Ips

- Change to install individual flows, one per IP, for managing ARP/NO Traffic flows towards router owned Ips that are on the logical port connecting the switch.
- Reduced time to install all releveant openflows for a single pod from +/- 6 seconds to 100msec

NorthD – Fix extremely inefficient usage of lflow hash map

- Change the hash table size from a fixed 128 entries to max last seen entries.
- Can reduce startup time from > 15 Minutes to 11 seconds
- This also could reduce some possible failures due to the long processing time.

Northd Results

- Logical Flow count on 120 node test database reduced from 1.5m to 50k
- Southbound DB is no longer a bottleneck @ 120 nodes
- 30% decrease in northd iterations times

Database – RAFT heartbeats during compaction

- Frequent southbound DB leader elections in large scale scenario (250 nodes)
- Can occur in 100 node cluster
- Shows as connection dropped
- Note this is in 4.8.z
- Alternative fix is to do a controlled change to another master for compaction.
- BZ 1943631

Database – Avoid unnecessary re-connections with updating remove

- The current implementation can cause a storm of reconnects, as well as reloading database content creating high load on database servers.
- Corrected by saving if its still in list of remotes
- This should increase stability of cluster reducing load during recovery or network issues.
- Upstream patch

Database Results

- 50% memory usage reduction
- 75% CPU usage reduction
- RAFT cluster stability increases

Database: Two-tier/Relay Architecture

- Relay architecture
  - Multiple database server proxy for a Raft Cluster
  - Proxy forwards OVSDB transactions between RAFT Members and a subset of clients
  - Client load is distributed to a arbitrary number of cattle databases
  - Allows the HA members to be isolated from load
  - Increases Stability of Cluster
- Relay Service Model – see: https://docs.openvswitch.org/en/latest/ref/ovsdb.7/

Ovn-Controller – ovn-controller should update OF rules atomically

- Old methodology – Remove old rule, on next cycle add new rule
  - Caused app dataplane to be down
  - Caused Packet loss
- New methodology – Remove and add new in same cycle
- Avoids creating additional version of OF tables
- BZ 1947398

OvnController - Result

- Huge memory Usage Reducation in ovn-controller and ovs-vswitched
- Large reduction in CPU uage in ovn-controller and ovs-vswitched
- Large decrease in latency to install flows for OVS

For Further Info:

- https://docs.google.com/document/d/1c5eQM4rVTLjns6smkvD1-hpbbNPx5-jJTo8rD3Zz7G8/edit
- This document: https://github.com/glennswest/presentations/ovn-2h-2021.pdf