TungstenFabric (Contrail) at Scale in Workday

Mick McCarthy, Software Engineer @ Workday
David O’Brien, Software Engineer @ Workday
Introduction
Mick McCarthy

- Software Engineer @ Workday
- Providing Network services to the Workday Private Cloud based on OpenStack
David O’Brien

- Software Engineer @ Workday
- Providing Network services to the Workday Private Cloud based on OpenStack
Contrail at Scale in Workday

- Workday - Enterprise SaaS
  - HCM, Finance, Payroll
Contrail @ Workday
History

- Running Contrail in Production since early 2016
- Versions in Production
  - 2.21.x => single controller - non-HA
  - 3.2.x => 3 controller - HA
Use Cases

• Providing Networking Services for OpenStack based Private Cloud
  – Overlay Networking (MPLSoGRE)
  – DNS
  – DHCP
  – Segmentation
Contrail Architecture
Scale
Scale

- 35+ OpenStack/Contrail Clusters
- 300K+ Cores
- 4K+ Hypervisors
- 20K+ Virtual Machines (Immutable Images)
- 150K+ Contrail Network Policies
- 100+ Tenant Networks
- 15+ Critical Workday Services
High Availability
High Availability

Diagram showing the structure of a high availability system with HAProxy (Active) and HAProxy (Backup) connected to Contrail API, Neutron API, Config, and Control layers. Below these layers, RabbitMQ, Cassandra, and Zookeeper are shown.
1. Fault Tolerance

1. Throughput

1. ZDT upgrades
1. Operational Complexity

1. “24 x 7” availability

1. “HA” HAPerxy?

1. ZDT upgrades
High Availability - Lessons Learned

(1/4) Observability

(2/4) Orchestration

(3/4) Smaller clusters (more of them)
(4/4) Contrail DNS

- Hard to configure internal DNS delegations
- Contrail DNS keeps 2 out of 3 as active
Weekly Production Release Cycle
Immutable Images

- Workday Services packaged as VM images
- New service version is a new version of a VM image
- Weekly service deployments (tight patch window)
- 20K+ VM deletion and recreation
Control Plane Usage

- Number of POST /v2.0/ports.json per sec
Challenges

- **Duplicate IPs**
  - Contrail bug visible only under high control plane load
  - Contrail uses Zookeeper to figure out next available IP in a subnet
  - Caused by Zookeeper race condition

- **Delayed DHCP**
  - Contrail Control plane slowing down under high load
  - vRouter handling out short term incomplete DHCP leases
  - Freed up Contrail Control plane by adding memcache for Keystone (helped a bit)
  - Optimized client side to reduce Contrail API traffic (big relief)
Challenges

- **Contrail Schema Failover**
  - Schema busy processing requests (CPU intensive)
  - Schema gevent greenlet does not yield to Zookeeper heartbeats
  - Multi master schema (causes data corruption)
Data Plane Usage

- bps through the gateway routers (Juniper MX40)
Challenges

• Contrail Analytics overwhelmed with Flow data
  – Too much flow telemetry data
  – High CPU usage (all cores)
  – High IO/Disk usage on Cassandra
Lessons Learned

• Always have a Production like environment
  – End to End
  – Exactly as it happens in Production

• Test frequently
  – Ideally in CI to narrow down the changes

• Design fault tolerance with SLA in mind
  – Not enough redundancy from SLA point of view
  – Loss of one DNS would bring down DNS briefly and violate SLA

• Monitoring, Monitoring, Monitoring
Segmentation
Segmentation

- Fine Grained Network Policies
  - Layer 4 rules per service port
  - 150K+ rules
- Tenant Isolation
  - Subnet for each tenant
  - DNS subdomain per tenant
Challenges - Network Policies

• Size of compiled ACLs
  – Proportional to the number of rules across all policies applied to a virtual network.
  – Starts becoming bigger than the max HTTP request body size allowed by the Python Bottle web server

• Policy Updates
  – CPU intensive, needs to walk the policy, network graph
  – Schema transformer gets really busy
  – Processing happens in a greenlet, doesn’t yield to anything else
  – Removed a lot of east west policies in Dev clusters
Challenges - Tenant Isolation

- **IP Address Management**
  - Unique, non overlapping subnet per tenant
  - OpenStack custom Heat plugins integrated with internal IPAM system for creating and allocating subnets to tenant networks
- **Reverse DNS**
  - Could not get reverse DNS to work with Contrail
  - Simplifying the DNS stack
  - Moving to native Neutron extensions
Lessons Learned

• Is fine grained L4 isolation really required?
  – East West wide open within an environment
  – Mutual TLS
• Dedicated CIDRs per cluster
  – Easier to separate out gateway routers in more fine grained manner.
    Advertise relevant cluster prefix to the underlay.
  – IPv4 address space limitation
• Contrail’s concurrency model
• Automation
Conclusion

1. Contrail

1. Plan Ahead for Scaling

1. Smaller clusters (more of them)

1. DevOps mindset
Thank You

Q&A

mick.mccarthy@workday.com
david.obrien@workday.com