Comparison of control plane deployment architectures in the scope of hyperconverged OpenStack infrastructure

Miroslav Halas
Lenovo Cloud Technology Center
May 2017
About me

- Director of SW Architecture, Cloud Infrastructure
- SW Engineer, Architect and Leader
  - Designed, implemented and operated Desktop, Mobile, SaaS, Cloud Applications and Large Private Cloud Platforms
  - 10 years of engineering leadership for information security, public and private cloud for one of top 3 financials in US
  - Holds patents in security, private and public cloud areas
- Carries picture of his home server in his pocket

linkedin.com/in/miroslavhalas/
Outline

- Introduction to Lenovo Cloud Technology Center
- What It Takes to Deploy and Operate SDDC
- What Is Control Plane and Why Does It Matter
- Resilient Converged Infrastructure Control Plane
- Evaluation and Testing using Rally and Phoronix Test Suite
Standing on the Shoulders of Giants

Lenovo DCG Research & Technology
• Focused research in Systems, Storage, Cloud, AI, Big Data, etc.

Lenovo System Technology Innovation Center
• Co-innovating with partners and customers to deliver Next-Next ideas such as Abstract hybrid data centers, Dynamic reconfiguration of IT and Deployment of IIoT

Lenovo Cloud Technology Center
• Pioneering cloud advancements, deployment and management experiences with community, partners and customers

Lenovo DCG Product & Development
• Engineer and support production ready enterprise solutions
**Lenovo Cloud Technology Center (LCTC)**

**Mission:** Align and Drive Lenovo entry and leadership in Enterprise Cloud Infrastructure, building core capabilities and partnerships for a complete portfolio of Solutions – focus on open source

**Key Activities**

- **Customer and Market Engagement**
  - Deep customer engagement and support
  - Business/Partner development, Technical Product evaluations
  - Open source ecosystem visibility and credibility

- **Portfolio Development**
  - OpenStack and Open Systems development – distros, containers and integrations with solutions
  - SDDC – development of software defined storage, networking
  - Professional and Managed services

- **Center of Competence**
  - Cloud architecture and advisory board
  - e2e Cloud roadmap

**Long Term Goal**

- **Credible Cloud Market Presence**
- **Complete Cloud Solutions Portfolio**
- **Depth and Alignment of Expertise**

**MAJOR RESEARCH CENTERS IN RALEIGH, BEIJING, SHANGHAI**
OpenStack Upstream Engagement

Reviews

- cinder: 18%
- ironic: 5%
- kolla-kubernetes: 4%
- os-brick: 4%
- training-guides: 3%
- python-cinderclient: 3%
- cinder-specs: 3%
- ironic-ui: 3%
- openstack-helm: 3%
- others: 5%

Commits

- training-guides: 32%
- pyghmi: 9%
- kolla-kubernetes: 6%
- cinder: 12%
- valence: 6%
- ironic-ui: 5%
- zun: 4%
- python-zunclient: 4%
- valence-specs: 3%
- others: 5%

2017 Lenovo. All rights reserved.
ManageIQ Upstream Engagement

- Initial provider of ManageIQ Physical Infrastructure
- Features/Use cases
  - Physical Server Data Model
  - New XClarity Provider
  - XClarity Provider Summary View
  - Physical Server Inventory via REST
    - Vital Product Data (VPD)
    - Firmware Levels
    - Server to Platform Host Relationships
- Outstanding development support from Red Hat ManageIQ / CloudForms team
- Open Source Contributions Summary
  - ManageIQ Lenovo Provider
    - https://github.com/ManageIQ/manageiq-providers-lenovo
  - Lenovo XClarity Client (Ruby)
    - https://github.com/lenovo/xclarity_client
What It Takes to Deploy and Operate SDDC

- Application Platform: Applications, Workloads, SaaS Platform
- Enablement Platform: PaaS Platform, Orch. & Mgmt, Govern, Workflow
- IaaS Platform: openstack, VMware, Microsoft Azure, Public Cloud
- Virtual Infrastructure: Bare Metal, Virtual Server, Virtual Storage, Virtual Network
- Physical Infrastructure: Servers, Storage, Network, Solution

Cloud Services

Integration, Placement, Config, Metering, Monitor, Alerts, Security

2017 Lenovo. All rights reserved.
Simple Deployment Pattern

Under cloud

All in One OSP

Overcloud

Controllers

Storage

Computes

Hyperconverged

Management and Operations

2017 Lenovo. All rights reserved.
Realistic Deployment Pattern
Large Deployment Pattern

- Very Large Central Site Cloud
- Mid Size Regional Cloud
- Small Local Cloud

Management Cloud
Size / Cost / Resource Optimized Cloud Infrastructure

- All deployment patterns have element that is sensitive to cost / space / power / skills / complexity
- Objectives achieved by compressing and collocating distinctive functions on the same set of HW and utilizing familiar concepts
- Converged Infrastructure Management Plane (CIMP)
  - OpenStack and non OpenStack management functions share the same HW resources
- Compressed Data Plane
  - Hyperconverged Compute Node collocating compute and storage on the same node
- Resource protection of shared resources
What Is Control Plane and Why Does It Matter
Monolithic Controller

Identical Controller Nodes

<table>
<thead>
<tr>
<th>Identity</th>
<th>Messaging</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bare Metal</td>
<td>Object Storage</td>
</tr>
<tr>
<td>Networking</td>
<td>Compute</td>
</tr>
<tr>
<td>Image Services</td>
<td>Mongo</td>
</tr>
<tr>
<td>Orchestration</td>
<td>MariaDB</td>
</tr>
<tr>
<td>Telemetry</td>
<td>OpenvSwitch</td>
</tr>
<tr>
<td>Workflow</td>
<td>HAProxy</td>
</tr>
<tr>
<td>Pacemaker*</td>
<td>Galera*</td>
</tr>
<tr>
<td>Block Storage*</td>
<td>Dashboard*</td>
</tr>
<tr>
<td>Clustering*</td>
<td>Shared Filesys*</td>
</tr>
</tbody>
</table>
Control Plane Beyond OpenStack Services

- Required to support full set of SDDC features
- Provide resiliency and autonomy for remote locations

- SDN – Contrail, NSX, Nuage
- Configuration Management – Ansible Tower, Pupper, SaltStack
- Logging – Logstash, Splunk
- Analytics – Elasticsearch
- Visualization – Kibana, Graphana, Prometheus
- Monitoring – Nagios, Zabbix, DataDog
- Performance Monitoring – Telegraf, CollectD, Graphite, InfluxDB
- Security – PowerBroker, ESM, CyberArk
- Capacity planning and optimization – Cirba, ManageIQ,
Disaggregated Controller

**Generic Services Controller**
- Identity
- Bare Metal
- Networking
- Image Services
- Orchestration
- Clustering
- Block Storage
  - Messaging
  - Object Storage
  - Shared Filesys
  - OpenvSwitch
  - HAPolicy
  - Workflow
  - Pacemaker

**Database Controller**
- Telemetry
- HAPolicy
- Pacemaker
- Mongo
- MariaDB
- Galera

**UI Services Controller**
- Dashboard
- HAPolicy

**Compute Scheduler Controller**
- Compute
- HAPolicy

---

OpenStack Performance Team, Barcelona Summit 2016
Hyperconverged Compute and Storage Node

- Compute Node
  - Compute
  - KVM
  - OpenvSwitch
  - Telemetry

- Block Storage Node
  - Block Storage
  - Telemetry
  - OpenvSwitch

- Object Storage Node
  - Object Storage
  - Ceph*
  - OpenvSwitch
  - Telemetry

Hyperconverged Compute and Storage Node

- Compute
  - KVM

- Telemetry
  - OpenvSwitch

- Block Storage
  - Ceph*
Hyperconverged Controller Compute and Storage Node

C+C+S Node
- Identity
- Networking
- Telemetry
- Compute
- Messaging
- OpenvSwitch
- MariaDB
- Compute / KVM
- Ceph OSD
- Swift

C+C+S Node
- Dashboard
- Networking
- Image Service
- Compute
- OpenvSwitch
- MariaDB
- Compute / KVM
- Ceph OSD
- Swift

C+C+S Node
- Image Service
- Networking
- Telemetry
- Identity
- Messaging
- OpenvSwitch
- MariaDB
- Compute / KVM
- Ceph OSD
- Swift

C+C+S Node
- Identity
- Telemetry
- Dashboard
- Compute
- Image Service
- OpenvSwitch
- Mongo
- Compute / KVM
- Ceph OSD
- Swift
Containerized Controller

Kolla, OpenStack Helm, OpenStack LOCI

https://docs.openstack.org/developer/kolla-kubernetes/deployment-guide.html

helm install kolla-kubernetes/helm/service/mariadb --name mariadb
helm install kolla-kubernetes/helm/service/rabbitmq --name rabbitmq --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/memcached --name memcached --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/keystone --name keystone --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/glance --name glance --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/cinder-control --name cinder-control --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/nova-control --name nova-control --values ./cloud.yaml
helm install kolla-kubernetes/helm/service/nova-compute --name nova-compute --values ./cloud.yaml
helm install kolla-kubernetes/helm/microservice/nova-cell0-create-db-job --name nova-cell0-create-db-job --values ./cloud.yaml
helm install kolla-kubernetes/helm/microservice/nova-api-create-
helm install kolla-kubernetes/helm/service/cinder-volume-lvm
watch -d -n 5 -c kubectl get pods --all-namespaces
Containerized OpenStack Infrastructure

DCOS Node

OpenStack Services

DCOS Node

OpenStack Services

DCOS Node

OpenStack Services

DCOS (K8S, Docker, Mesos)

DCOS Management components (etcd, K8S Master, etc.) provisioned, operated and managed with production SLAs
Resilient Converged Infrastructure Control Plane

- Use familiar and production ready technologies to provide cost / space / power / skills / complexity effective control plane required to support full set of SDDC features
- Provide resiliency at all levels for lights out operations
  - Network
    - Redundant management and data plane switches
    - Redundant dual port NICs
  - Servers and storage
    - Redundant PDUs and power supplies
    - 3+ node bare metal server cluster sizes
  - Software
    - Off the shelf, commodity software
    - Live Migration, Shared storage, Snapshots, Backup
Control Plane for Resource Constrained Deployments

- Standard Lenovo servers and network switches
- Management components (XClarity, Undercloud, ManageIQ, etc.) deployed in virtualized fashion
- Distributed storage provided by Ceph
- Ceph deployed in hyper-converged mode alongside the KVM hypervisor
- Easy expansibility to host future management and operations functions

Rack

Converged Infrastructure Control Plane

3 Bare Metal Servers
Converged Infrastructure Control Plane

- HW Mgmt VM
- Ops VM 1-N

Each Compute / Storage Bare Metal Server

- Undercloud VM
- 3 x Controller VMs

Compute/Storage Hosting Plane

- Nova-compute process
- Ceph OSD process
Virtualized Control Plane Structure

- Hypervisor
  - KVM, Libvirt
  - Live Migration
  - Snapshots

- Networking
  - Linux Bridge
  - VLANs

- Storage
  - GlusterFS
  - Shared storage
  - File system replication
  - HA
Lenovo Integrated HW Platform in 2 Form Factors

42U Front
- 2x RackSwitch G7052 (Management network)
- Redundant power supply for G7052 (Optional)
- 2x RackSwitch G8272 (VM/storage network)
- Up to 13x System x3650 M5 hyperconverged nodes (#4-16: Optional)
- Up to 4x PDUs (3-5 nodes: 2x PDUs; 6-16 nodes: 4x PDUs)
- 3x System x3650 M5 hyperconverged nodes (#1-3: Required)
- 3x System x3550 M5 management nodes

42U Rear

10U Front
- 14 compute node bays
- Half or Full Width Compute node
- Information panel

10U Rear
- Six hot-swap power supplies
- Eight hot-swap 80 mm fan modules
- Two hot-swap 40 mm fan modules
- Two hot-swap Chassis Management Modules
- Four hot-swap switch bays
- Information panel

Switch bays: 1 3 2 4
Lenovo HW Management Platform - XClarity
Network Topology

Management Network

Data Network
Virtualized Control Plane Networking Resiliency

CIMP-node

eth0  eth1  ens4f0  ens4f1  ens5f0  ens5f1

1G switch

native PXE (13), IMM (3), Internal Management (603),

HC-node

eth0  eth1  ens4f0  ens4f1  ens5f0  ens5f1

10G switch

cloud-public (103), cloud-tenant (600-700),

cloud-management (403), cloud-storage (303)
Physical Network Setup and Resiliency
Virtualized Control Plane Networking
Virtualized Control Plane Storage

1. GlusterFS Installation
   - `# yum update -y`
   - `# yum install glusterfs-server`
   - `# systemctl enable glusterd`
   - `# systemctl start glusterd`
   - `# systemctl status glusterd`
   - Configure firewall to enable traffic on ports used by gluster

2. Build XFS bricks
   - `# pvcreate /dev/vdb`
   - `# vgcreate vg_gluster /dev/vdb`
   - `# lvcreate -L 1000G -n brick1 vg_gluster`
   - `# mkfs.xfs /dev/vg_gluster/brick1`
   - `# mkdir -p /bricks/brick1`
   - `# mount /dev/vg_gluster/brick1 /var/bricks/images`
   - Add the following line at end of `/etc/fstab`:
     `/dev/vg_gluster/brick1 /bricks/brick1 xfs defaults 0 0`

3. Configure trusted pool
   - `# gluster peer probe cimp-node2`
   - `# gluster peer probe cimp-node3`
   - `# gluster peer status`

4. Create GlusterFS volumes
   - `# sudo gluster volume create vol-1 replica 3 cimp-node1:/var/bricks/running cimp-node2:/var/bricks/running cimp-node3:/var/bricks/running`
   - `# sudo gluster volume start vol-1`
   - Confirm the Gluster volume running:
   - `# sudo gluster volume info all`

5. Use the GlusterFS volume as a shared storage pool
   - `# sudo mkdir -p /var/images/running`
   - `# sudo mount -t glusterfs cimp-node1:/vol-1 /var/images/running`
   - `# echo "127.0.0.1:vol-1 /var/images/running glusterfs defaults,_netdev,noauto,x-systemd.automount 0 0" >> /etc/fstab`
   - `# sudo setsebool -P virt_use_fusefs 1`
Virtualized Controller VM Definition

```xml
<domain type="kvm" id="1">
<name>controller1</name>
<memory unit="GB">64</memory>
...
<os>
  <type machine="pc" arch="x86_64">hvm</type>
  <boot dev="network"/>
  <boot dev="hd"/>
...
<devices>
...
  <disk device="disk" type="file">
    <driver type="qcow2" name="qemu"/>
    <source file="/var/images/controller/controller1_os.qcow2"/>
  ...
  </disk>
  <disk device="disk" type="file">
    <driver type="qcow2" name="qemu"/>
    <source file="/var/images/controller/controller1_mongo.qcow2"/>
  ...
</disk>
</devices>
<interface type="bridge">
  <source bridge="br-pxe"/>
  <target dev="vnet0"/>
...
</interface>
<interface type="bridge">
  <source bridge="br-cloud1"/>
  <target dev="vnet1"/>
...
</interface>
<interface type="bridge">
  <source bridge="br-cloud2"/>
  <target dev="vnet2"/>
...
</interface>
</domain>
```

```
virsh create controller1.xml
```
Virtualized OpenStack Controllers Considerations

- **Power Control**
  - Undercloud to use `virsh` to control power management of other nodes
  - Pre Pike release `pxe_ipmitool => pxe_ssh`
  - Starting from Pike transition to `virtualbmc`

- **High Availability**
  - Core services – Galera, RabbitMQ, Redis
  - Active-Passive services – Cinder-Volume service
  - SystemD services – independent and able to withstand service interruption
  - Isolating a faulty node to protect a cluster and its resources
  - Pacemaker + Shoot-The-Other-Node-In-The-Head
  - Fencing agent – `fence_ipmilan / fence_xvm, fence_virt`
Hyperconverged Compute Node

- Deploy Ceph OSD alongside nova-compute on the same node
- Tuning - Compute shares memory and CPU with Ceph OSD
  - Nova.conf – set aside CPU and memory for Ceph OSD
    - cpu_allocation_ratio / reserved_host_memory_mb
  - Ceph.conf – balance system resources needed for ceph recovery and rebalancing and guest workloads
    - Osd_recovery_op_priority, osd_recovery_max_active, osd_max_backfilles
- Tune overall system performance - throughput-performance
- NUMA Pinning of Ceph OSD processes <=> NIC PCIe Slots

Operations
- Nova / Compute operational workflows must account for Ceph OSD processes and vice versa
Testing using Rally

Tenant creation with user

- **Conditions**
  20-concurrency tests (create a tenant with user) and totally accomplish 100 times of tests

- **Overall Results**
  All of the tasks were successfully finished. Majority of them accomplished the tasks within 4 seconds and the maximum consumed 6 seconds
Testing using Rally

Snapshot creation and deletion

- Conditions
  10-concurrency tests (create a snapshot and delete it, set it to 10 since the snapshot limitation is 10) and totally accomplish 100 times of tests

- Overall Results
  All of the tasks were successfully finished. Majority of them accomplished the tasks within 17 seconds and the maximum consumed 23 seconds
Testing using Rally

Small instance creation and deletion

- **Conditions**
  1. Image: TestVM (around 30M)
  2. Flavor: 1 core + 512M RAM+ 20G Disk
  3. 20-concurrency tests (create an instance and then delete it) and totally accomplish 100 times of tests

- **Overall Results**
  All of the tasks were successfully finished. Majority of them accomplished the tasks within 28 seconds and the maximum consumed 30 seconds
Testing using Rally

Large instance creation and deletion

- **Conditions:**
  1. Image: Redhat 7.2 (around 4G)
  2. Flavor: 8 core + 19G RAM + 100G Disk
  3. 5-concurrency tests (create an instance and then delete it, set concurrency to 5 due to quota limitation) and totally accomplish 100 times of tests

- **Overall Results**
  Almost all of the tasks were successfully finished, some of them failed because the RAM of coexisted instances exceeded the quota limitation. Majority of them accomplished the tasks within 58 seconds and the maximum consumed almost 60 seconds.
Testing using Phoronix Test Suite

- Open source testing suites including test development framework, test runner, management and reporting
- Benchmark catalog at OpenBenchmarking.org is comprehensive 984 tests & suite
- Many real-world workloads wrapped for benchmarking
- Corpus of shared results for comparison / initial settings
- Easy to extend and share
Testing using Phoronix Test Suite

Average IOPS per Number of VMs (randread)

Number of VMs
1  pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 /
3  pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 /
6  pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 /
15 pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 10g /
30 pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 10g /
45 pts/fio-1.9.0 - randread libaio 0 1 8k 1 64 10g /
Testing using Phoronix Test Suite

Compilation time vs number of VMs

- Build Linux kernel
- Build Linux Kernel Min
- Build Linux kernel max
- Build Apache
- Build Image Magick

pts/build-apache-1.5.1 - pts/build-imagemagick-1.7.2 - pts/build-linux-kernel-1.7.0 - pts/build-php-1.3.1 -
Virtualized Control Plane Operations

- Similar concept as containers but ready to use today
- Reuse existing expertise in operating virtualized infrastructure
- Using the Host HA capabilities in addition to the app level HA
- Better resource utilization for resource limited deployments, can pack more components on the same HW
- VM snapshots before patching
- VM migration before HW upgrades
- VM resizing to provide additional resources
- VM restore after failure
  - Implication on VM structure and services deployments, “OS/services” stateless disk and data on separate disk
  - Should be combined with and is not replacement for backup strategy
Potential Future Work

- Reusable VM images for Undercloud and Overcloud
- Containerized Control Plane
Resources

- https://access.redhat.com/articles/2922421
- https://access.redhat.com/articles/2360321
- https://access.redhat.com/articles/2861641
- https://access.redhat.com/documentation/en-us/red_hat_openstack_platform/10/html-single/understanding_red_hat_openstack_platform_high_availability/
- http://docs.openstack.org/developer/performance-docs/
- http://tripleo.org/environments/virtualbmc.html
- https://docs.openstack.org/developer/kolla-kubernetes/deployment-guide.html
- https://www.phoronix-test-suite.com/
- https://openbenchmarking.org
thanks.