CSNS Computing Environment Based on OpenStack

Yakang Li
CSNS, Branch of IHEP
HEPiX 2016 Fall, LBNL
Outline

- About CSNS
- Scenarios & Requirements
- Computing Environment based on OpenStack
- R & D
- Summary
About CSNS

• Accelerator-based neutron source

• Designed to provide multidiscipline research platforms with neutron scattering

• Operated by the Institute of High Energy Physics, CAS

• Located at Dongguan in Guangdong province of China

• Planned for operation in 2018
About CSNS

- 80-MeV $^1\text{H}^-$ Linac
- 1.6-GeV proton rapid cycling synchrotron (RCS)
- 25 Hz repetition rate
- tungsten target station
- 3 initial spectrometers

Data Processing
CSNS Data Flow & Storage Policy

Instrument
- GPPD
  - DAQ
  - Control
  - Monitor
- SANS
  - DAQ
  - Control
  - Monitor
- MR
  - DAQ
  - Control
  - Monitor

Central Control Room
- Central Storage Region
- LogDB

Central Data Zone
- Raw Data Zone
- Reconstruction Zone
- User Data Zone

Compute Center Storage
- Central Backup Zone

backup
- nxs
- log
- cal
- result
- summary

Browser & Search
Analysis
Visualization
Scenarios & Requirements(1)

Software
- OS (windows & Linux)
- Diverse Analysis Softwares

Hardware
- Different Memory requirements
- Different CPU requirements

Scalability
- More spectrometers
- More data
Scenarios & Requirements(2)

- Different os, softwares
  - Diverse images

- Elastic hosts
  - virtualization

- Scalability
  - Scalable Resource pool
Computing Environment based on OpenStack

Backup Node
- MySQL
- RabbitMQ
- KeyStone
- Glance
- Nova
- Neutron

Master Node
- MySQL
- RabbitMQ
- KeyStone
- Glance
- Nova
- Neutron

Master Node
- nova-compute
- KVM
- GlusterFS

Node 01
- GlusterFS

Node 02
- GlusterFS

Storage cluster

Manager Network

Storage Network

public network

Trunk

Virtual Network

router
Research and development
Unified Authentication

- **Existing integration schema doesn't meet the requirement**
  - All stores in ldap
  - Too much change to ldap
- **Loosely coupled schema**
  - Local user and common user
  - For common users, only username and password are authenticated by ldap service
  - Other information will be authenticated and authorized through keystone local DB

```
keystone
```

```
Local account?
```

```
Yes
```

```
SQL authentication
```

```
success
```

```
No
```

```
LDAP
```

```
- - -
```

```
keystone
```

```
Local account?
```

```
Yes
```

```
SQL authentication
```

```
success
```

```
No
```

```
LDAP
```

```
- - -
```
Network

- **Virtual Network**
  - **Disable L3-agent**
  - Replace virtual router with physical gateway
  - VMs directly connect to the trunk mode switch

- To ensure the performance and stability of the network
- To achieve seamless communication directly with local network
Images & instances

• **Images storage**
  – Stored in glusterfs ssd volume

• **Cloud-init**
  – Initialize instances at boot time
  – Set an instance hostname
  – Generate instance ssh private keys
  – Automatically register in puppet, DNS, etc

• **Live Migration**
  – All instances shared storage with glusterfs volume
  – Completed within a few seconds
  – Instance will not stop in the migration process
Distributed Messaging System

- **RPC Messaging is critical for OpenStack**

- **Default Messaging System**
  - RabbitMQ

- **Problems**
  - single point failure
  - Difficult to scale out

How to implement a broker-less architecture for OpenStack RPC
ZeroMQ is a high performance asynchronous messaging library aimed at used in scalable distributed or concurrent applications.
Distributed Messaging System

Source: Going brokerless, the transition from qpid to 0mq.

Each host needs to listen to a certain TCP port for incoming connections and directly connect to other hosts simultaneously.

ZeroMQ Receiver running on every component.
Dashboard

Overview

Physical Resource
- CPU(Core)
  - 11.25%
  - 27/240
- Ram(GB)
  - 10.92%
  - 55/503
- Local Storage(GB)
  - 0%
- Floating IP
  - 0/0

Service Status (Error/Normal)
- Calc
  - 13.04%
  - 3/23
- Network
  - 35.71%
  - 5/14

Virtual Resource
- Tenant
  - Total: 10
- ERROR1
  - Virtual
    - NORMAL: 11
    - ERROR: 0
  - Switcher
    - NORMAL: 1
    - ERROR: 0
  - Router
    - NORMAL: 0
    - ERROR: 0
  - Loadbalancers
    - NORMAL: 0
    - ERROR: 0

Operate Logs

<table>
<thead>
<tr>
<th>User Name</th>
<th>Operator</th>
<th>Type</th>
<th>Result</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>test</td>
<td>Create</td>
<td>instances</td>
<td>Success</td>
<td>2016-09-28 17:29:18</td>
</tr>
<tr>
<td>test</td>
<td>Start</td>
<td>instances</td>
<td>Success</td>
<td>2016-09-27 14:06:42</td>
</tr>
<tr>
<td>test</td>
<td>Stop</td>
<td>Instances</td>
<td>Success</td>
<td>2016-09-27 14:05:57</td>
</tr>
</tbody>
</table>
Dashboard

- **RealTime Notification(polling mode) → WebSocket Push**
- Use socket.io running inside a NodeJS loop.
- A high performance websocket (RFC 6455) implementation has been added
- Using Redis as a message queue.
- Two instances of a uWSGI server:
  - one to handle normal HTTP requests for Django
  - one to handle WebSocket requests
Summary

- OpenStack and virtualization technology are good solution according to the computing scenarios and requirements of CSNS;
- Computing environment based on OpenStack is deployed and running well;
- Some R&D points are made from the aspects of unified authentication, network, messaging system, etc;
- More advices, suggestions and helps are strongly expected
Thank You!

liyk@ihep.ac.cn

CSNS