CSNS computing environment Based on OpenStack

To cite this article: Yakang Li et al 2017 J. Phys.: Conf. Ser. 898 082051

View the article online for updates and enhancements.

You may also like
- Platinum-nanoparticle-supported core–shell polymer nanospheres with unexpected water stability and facile further modification
  Conghui Yuan, Yiting Xu, Weiang Luo et al.
- Analysis of displacement damage effects on bipolar transistors irradiated by spallation neutrons
  Yan Liu, , Wei Chen et al.
- Beam loss and the stripping efficiency measurement for CSNS injection system
  Ming-Yang Huang, Sheng Wang and Shou-Yan Xu
CSNS computing environment Based on OpenStack

Yakang Li¹,², Fazhi Qi¹,², Gang Chen¹,², Yanming Wang¹,², Jianshu Hong¹,²
¹ Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
² Dongguan Neutron Science Center, Dongguan, China
Email:liyk@ihep.ac.cn

Abstract. Cloud computing can allow for more flexible configuration of IT resources and optimized hardware utilization, it also can provide computing service according to the real need. We are applying this computing mode to the China Spallation Neutron Source (CSNS) computing environment. So, firstly, CSNS experiment and its computing scenarios and requirements are introduced in this paper. Secondly, the design and practice of cloud computing platform based on OpenStack are mainly demonstrated from the aspects of cloud computing system framework, network, storage and so on. Thirdly, some improvements to openstack we made are discussed further. Finally, current status of CSNS cloud computing environment are summarized in the ending of this paper.

1. Introduction

CSNS[1] stands for the China Spallation Neutron Source. It is an accelerator-based neutron source, designed to provide multidiscipline research platforms with neutron scattering, now it’s operated by the institute of high energy physics(IHEP), under construction at Dongguan in Guangdong province of China. The project is supported by Chinese central government and guangdong province, civil engineering work has been completed on July of 2016, it’s for operation in 2018.

![Figure 1CSNS Data Flow](image-url)

The main target of CSNS is to offer research support for physics, life science, material science and other fields by performing data process and data analysis. As shown in Figure 1, the raw data comes from the spectrometers, will be firstly stored in central control room momentarily, then it’s transferred into both the central data storage region and the central backup region at the same time with the
special transfer system, that means the raw data is stored in two replication to ensure the data security. In the next stage, raw data will be processed into reconstructor data, the reconstructed data will also be backup to central backup region during the experimental period. Next, reconstructor data will be processed into the user data which open to the end experiment user to perform data analysis, visualization and so on.

CSNS will be actually used in broad range of fields, such as physics, material science, medicine, and so on. So from the software aspect, the experiment users may use different operating systems and softwares to perform their data analysis; From the hardware aspect, users in different fields may have different memory and cpu requirements, maybe some one need more memory and the other need more cpu cores; From the scalability aspect, in the near future, with the increasing application of CSNS, more spectrometers will be built, and more data will be generated, so more computing and storage resources will be needed. Fortunately, openstack can be a good solution to solve these problems[2].

2. Cloud Computing Environment Based on OpenStack
We are applying openstack[3] to the China Spallation Neutron Source(CSNS) computing environment, and a testbed based on openstack has been deployed, as shown in Figure 2.

![Deployment architecture of CSNS OpenStack](image)

In order to provide incessant services, we deploy two control nodes to provide basic control and network service, control service is the hot standby architecture and network service is the cold standby architecture. We apply kvm as the hypervisor, because it’s simple and easy to use in comparison to xen. The network, according to its function, is divided into the management network, the storage network, and service network. Vlan is used in the service network, trunk mode is needed for the switch to allow all data packages with different vlan tags to get through. Glusterfs[4] is used to provide storage service, which contains image storage, instance storage and cinder storage[5].

3. R & D
The first paragraph after a heading is not indented (Bodytext style).
Other paragraphs are indented (BodytextIndented style).

3.1. Unified Authentication
The OpenStack identity service supports integration with existing LDAP directories for authentication and authorization services. It requires all authentication and authorization data stored in ldap, but for the running LDAP service in some enterprises and organizations, it’s hard to modify the architecture of ldap to meet the requirement of keystone. So we apply a loosely coupled method to intergrate openstack authentication service to our local ldap service. Firstly we divide users into local users and common users, local user refers system users and admin users, and common user refers to the user who has unified authentication accounts. As show in Figure 3 ,for common users , only username and password are authencated by ldap service, other information is authenticated or authorized through
keystone local DB. For local users, all information are authenticated and authorized through keystone local DB. In this case, we implement unified authentication with no change to our ldap service.

![Authentication Flow Diagram](image)

**Figure 3 authentication flow**

### 3.2. Network

The adjustment of the network as shown in Figure 4, because of the bad stability and performance of L3-Agent, we disabled the L3-Agent, in this case, every virtual machine can be connected directly to the switch. This adjustment solved two problems: first, to ensure the performance and stability of the network; second, it can achieve seamless communication directly with local network.

![Network Architecture Diagram](image)

**Figure 4 Network Architecture**

### 3.3. Images & Instances

We use glusterfs ssd volume instead of hdd volume to provide image storage service, in this case, images can be copied to every computing node more faster, and to reduce the launching time. We use cloud-init script running on every new instance and initialize it according to a provided script or config file. Use cloud-init, at boot time, we can configuring instances, such as setting a instance hostname, generating instance ssh private keys, automatically registering in puppet, DNS, IPDB, and so on. We also configured the live migration, which means migrate the running instances from one OpenStack compute server to another compute server. We set all instances shared storage with glusterfs volume, so the instance can complete live migration within a few seconds and not stop in the migration process.

### 3.4. Distributed Messaging System

OpenStack uses a message queue to coordinate operations and status information among services, so it’s more dependent on the messaging system because of its flexibility of the deployment and loose coupling between modules. The performance of the message queue has a direct impact on openstack performance, so RPC messaging is critical for OpenStack. Its default messaging system is RabbitMQ, which has some problems, such as single point failure, difficult to scale out and obvious performance bottleneck and so on.

ZeroMQ\[^6\] is a high performance asynchronous messaging library aimed at used in scalable distributed or concurrent applications. Other than AMQP-based drivers, ZeroMQ doesn't have any
central brokers, instead, each host is both ZeroMQ client and server. As a result, each host needs to listen to a certain TCP port for incoming connections and directly connect to other hosts simultaneously. This picture Figure 5 shows that zeroMQ receiver running on every component in our openstack platform, such as nova-api, nova-compute and so on.

![Figure 5 zeroMQ receiver running on every component](image)

3.5. Dashboard
We developed a new portal for openstack to get a good user experience, as shown in Figure 6.

![Figure 6 Portal of CSNS Cloud](image)

4. Summary
CSNS is under construction, it need to be supported by the high performance and efficient computing environment in this process. Through investigation research and argumentation, we find that OpenStack and virtualization technology are good solution according to the computing scenarios and requirements of CSNS, and at present, a mini-production 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.

References
[1] Wei J, Chen H, Chen Y, et al. China Spallation Neutron Source: Design, R&D, and outlook[J]. Nuclear Instruments & Methods in Physics Research, 2009, 600(1):10-13.
[2] Li Y, Qi F, Wang Y, et al. Application of OpenStack in Chinese Spallation Neutron Source computing system[C]//World Automation Congress (WAC), 2016. IEEE, 2016: 1-5
[3] OpenStack web: http://www.openstack.org
[4] Developers G F S. The Gluster web site[J]. http://www.gluster.org, 2008
[5] Toor S, Osmani L, Erola P, et al. A scalable infrastructure for CMS data analysis based on OpenStack Cloud and Gluster file system[C]//Journal of Physics: Conference Series. IOP Publishing, 2014, 513(6): 062047
[6] Hintjens P. ZeroMQ: Messaging for Many Applications[J]. Oreilly Media, 2013