Evolution of the LHAASO Distributed Computing System based Cloud

Qiulan Huang[1,*], *Haibo* Li[1] , *Yaodong* Cheng[1] , *Jingyan* Shi[1] , *Wei* Zheng[1] , and *Qingbao* Hu[1]

[1]Computing Center, Institute of High Energy Physics, P.O.Box 918-7, 19B Yuquan Road, Beijing 100049, China

Abstract. In this paper we will describe the LHAASO distributed computing system based on virtualization and cloud computing technologies. Particularly, we discuss the key points of integrating distributed resources. A solution of integrating cross-domain resources is proposed, which adopt the Openstack+HTCondor to make the distributed resources work as a whole resource pool. A flexible resource scheduling strategy and a job scheduling policy are presented to realize the resource expansion on demand and the efficient job scheduling to remote sites transparently, so as to improve the overall resource utilization. We will also introduce the deployment of the computing system located in Daocheng, the LHAASO observation base using cloud-based architecture, which greatly helps to reduce the operation and maintenance cost as well as to make sure the system availability and stability. Finally, we will show running status of the system.

1 Introduction

The LHAASO(Large High Altitude Air Shower Observatory) experiment[1][2] of IHEP is located in Daocheng, Sichuan province (at the altitude of 4410 m). The main scientific goals of LHAASO are searching for galactic cosmic ray origins by extensive spectroscopy investigations of gamma ray sources above 30TeV. To accomplish these goals, LHAASO contains four detector arrays, which generates huge amounts of data and requires mass storage and high performance computing system. And the dedicated computing resource of LHAASO locates in Beijing, Daocheng and Chengdu as well as resources from collaborated organizations. How to establish a distributed computing system making the distributed resources work together and provide a good computing service for LHAASO is very important and urgent. However, it faces high operation and maintenance costs, system instability and other issues especially from remote sites. In this paper, we will propose a solution of integrating distributed resources and make them work as a resource pool.
This paper is organized as follows. The rest of Section 1 gives an overview of LHAASO computing requirements and dataflow. Section 2 illustrates the system architecture and implementation as well as discussing the key technical points. Section 3 introduces the

* Corresponding author: huangql@ihep.ac.cn

deployment of LHAASO Daocheng site. Section 4 shows the running status of this system. And we conclude the paper in Section 5.

1.1 LHAASO Computing Requirements

The LHAASO experiment generates about 6PB per year. In order to meet the requirements in the aspect of data storage and processing, it requires mass storage, high performance computing system as well as dedicated network, and the detailed requirements are shown in Table 1.

Table 1. Computing requirements of the LHAASO.

	Requirement	Description
Disk Storage	~20PB	Storing raw data and all historical reconstruction data
Tape Storage	120PB (6PB*10*2)	2 replica for 10 years
Computing	1000+ CPU cores, FPGA	Simulation, reconstruction and analysis
Dedicated Network	>2.5Gbps	~2PB per year to be transferred from Daocheng to Beijing

1.2 LHAASO Dataflow

Figure 1 shows the LHAASO dataflow. The LHAASO DAQ system is designed to read out large amounts of data from the front-end detectors and record valid data on permanent storage devices. In order to decrease the consumption of network bandwidth, the data will be compressed and quick reconstruction will be performed prior to the transfer from Daocheng to Beijing. From this purpose, we constructed an on-site data center at Haizi Mountain. The small data center contains 2000 CPU cores and 700TB disk storage managed by the EOS file system[3]. The processed data will be transferred and stored in large offline data center at IHEP in Beijing, which includes about 5000 CPU cores, 4PB disk and 20PB tape storage. The distributed computing system is an important part of the LHAASO offline data processing. When there are no available computing resources on site, jobs will be scheduled to the distributed computing system. Therefore, the data has replicas in some remote sites.

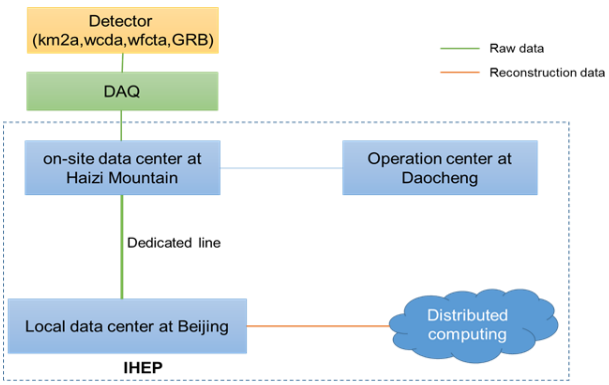


Figure 1. LHAASO dataflow

2 System Architecture

The system aims to provide a logical resource pool by integrating cross-regional computing resources for the upper-layer application. We adopt the Openstack[4]+HTCondor[5] to make the distributed resource work together and be transparent to jobs running in remote sites, so as to improve the overall resource utilization. Considering the resources (dedicated or opportunistic) for the LHAASO distributed in different locations, we define different site layers for the distributed system, which is showed in Figure 2.

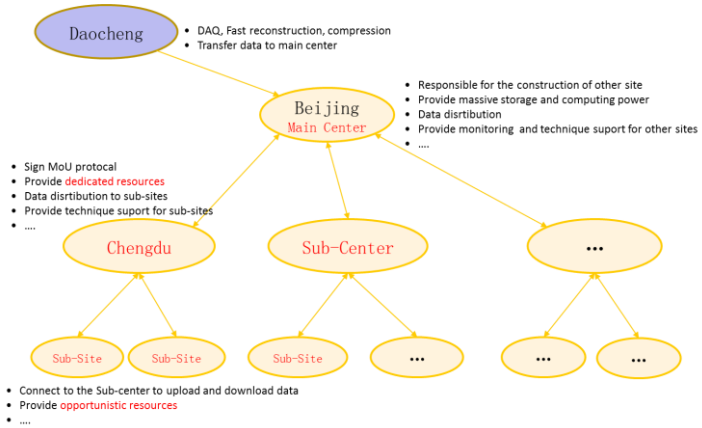


Figure 2. The schema of LHAASO distributed computing system

The Main center is the offline computing center at IHEP, located in Beijing, which provides massive storage and computing resources and should be responsible for the deployment and administration of other sites including data distribution, monitoring and technical support for other sites. Sub-Centers should provide dedicated resources, dispatch data to their sub-sites as well as they should provide technical support for their sub-sites. For Example, Chendu site is a sub-center, it has some sub-sites from universities. Sub-site connects to its superior site for the direct upload and download of data. It provides opportunistic resources. In order to manage each of sites better, we have each sub-Center sign MoU protocol with Main-Center.

Under this scheme, we introduce virtualization and cloud computing technologies into this system. On one hand, we use virtualization technology to hide the underling details instead of requiring the unified hardware configuration. On the other hand we can greatly reduce the high cost of operation and maintenance especially in some sites having shortage of experienced administrators.

The system design and implementation are based cloud computing models. The system architecture is shown in Figure 3. Some key technical points are needed to be addressed. (1) Providing unified distributed resource management; (2) Scheduling jobs across regions transparently; (3) Dynamic resource provision to meet the peak demand; (4) Distributed monitoring and automated deployment; (4) Security certification. The first key point we need to figure out is how to integrate distributed resource and provide unified resource management, which is the preliminary of this system. And we don't want to let user do any changes to use the system, so we have to make jobs scheduled to distributed sites transparently. Users don't need to care about where their jobs actually are running. Also we have to have distributed monitoring, which is incredibly useful for active investigation and troubleshooting. The monitoring system collects all the meaningful metrics (CPU, memory, load and so on) of physical machines and virtual machines and

makes them visualized in Grafana. Security certification is the key technical point. In this system, users having AFS account of IHEP can access it. And the AFS account authentication is based Kerberos 5, which makes system security.

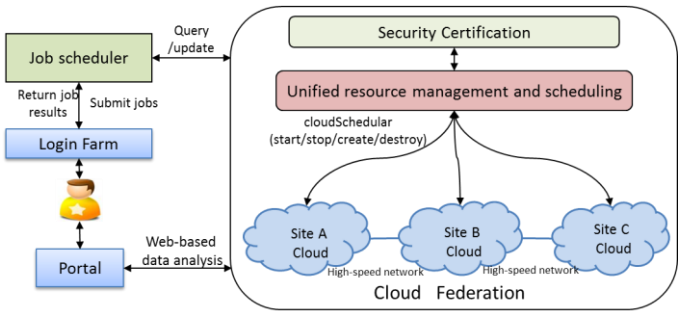


Figure 3. System architecture

2.1 Unified Distributed resource management

Principal considerations of distributed resource management are the efficient assignment of resources to application and the provisioning of unified management interface to application. In this system, we adopt the Openstack regions[6] to manage the distributed resources. Openstack regions are a good way to manage authentication and authorization for different clouds, with a shared identity. Keystone is the only service that is shared across regions. And other services such as Nova and Neutron are installed separately in each region and are not shared. In fact, the identification for LHAASO is shared across regions. All users accessing the distributed computing resources with the AFS[7] account registered in IHEP.

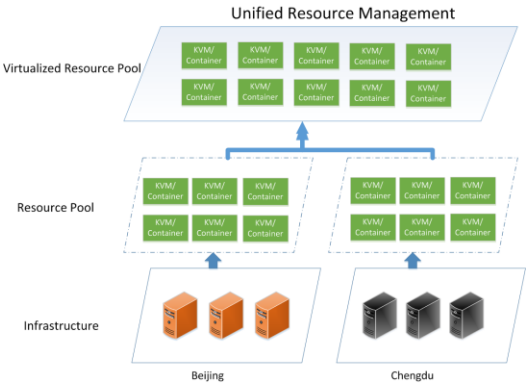


Figure 4. Unified resource management

Figure 4 shows the schematic diagram of unified resource management. A prototype is deployed and located in Beijing, Chengdu and Daocheng, which contains 1706 CPU cores and 278.6TB. The resources are listed by site in Table 2.

Table 2. Resources for the LHAASO distributed system.

	Beijing	Daocheng	Chengdu
CPU cores	1054	468	184
Storage capacity	62.6TB	20TB	196TB

2.2 Job Scheduler

The workflow of job scheduler is shown in Figure 5. The main idea of job scheduler is to schedule jobs to remote site via "Condor-C" model[8]. The jobs will be running in the local site by default. When the local site becomes busy with no more available computing resources, the jobs are scheduled to remote sites in a transparent manner which users (who submitted the jobs) do not need to care about where their jobs are running. The main components of jobs are listed as followings.

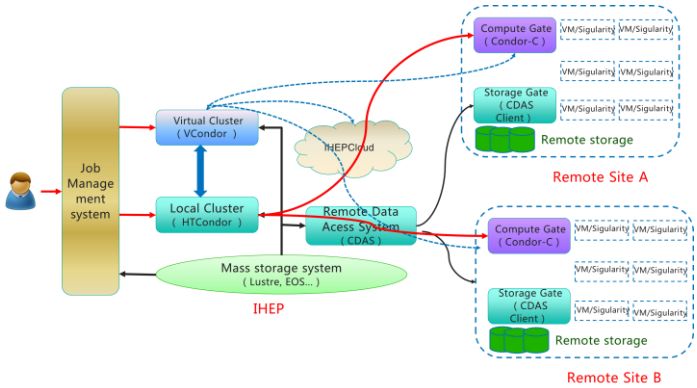


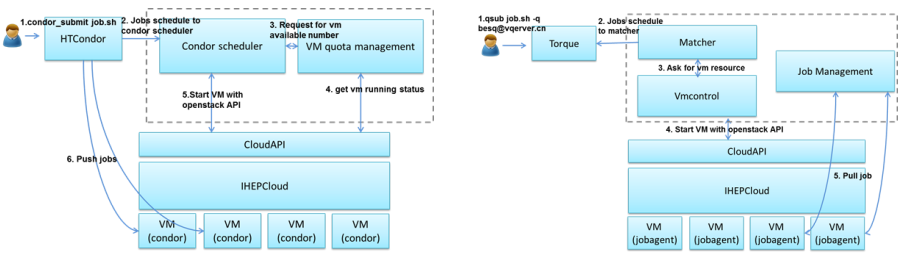
Figure 5. The workflow of Job Scheduler

(1) Job Management System

It is a toolkit developed based on HTCondor and it includes a set of commands such as hep_sub, hep_rm and hep_q for job submission, deletion and querying, respectively.

(2) Cloud Scheduler

This component is to integrate HTCondor/Torque PBS with IHEPCloud (a private cloud service developed by IHEP based on Openstack), aiming to provide elastic resource allocation service in this system. It contains VCondor and VPBS. VCondor is to provide a bridge between HTCondor and IHEPCloud. VPBS is the bridge between Torque PBS and IHEPCloud. The cloud scheduler allocates and reclaims VMs dynamically according to the status of HTCondor/Torque PBS queues. When users submit jobs to the cloud, the cloud scheduler requests the available cloud resources, and it boots the VMs to receive the jobs running there. Figure 6(a) and Figure 6(b) show the workflows of VCondor and VPBS respectively. VCondor works in push mode. When one VM starts, it will join in available slots automatically. VPBS behaves in pull-push mode. When a job comes, the Matcher service will request a specific VM slot. Once the VM is available, it will pull job and make it running. Once a job finishes, if there are no more jobs in the queue, the cloud scheduler will shut down the VM.



(a) VCondor (b) VPBS

Figure 6. The workflow of VCondor and VPBS

To control VMs remotely, we developed a CloudAPI [9] shown in Table 3, which contains a set of interfaces such as create, start, sop, delete and so on for VM orchestration. We also implemented a pilot-job scheme dispatching jobagents into VMs prior to the actual workloads running. The jobagent is responsible to pull jobs and monitor the VM status.

Table 3. A set of methods of CloudAPI

Method	Description
get_token(self)	Fetch tokens, return a string of tokens
list_image(self)	List images
list_flavor(self)	List flavors
get_active_vm_num(self)	Get the total number of active VMs
create_flavor(self,cpu,memory,disk)	Create new flavor with the input parameters(cpu,memory and disk size), return flavor ID
list_network_instance(self,serverid)	List network information of a VM instance
get_instance_from_ip(self,ip)	Fetch a VM instance ID by instance ID
create_vm(self,sname,imageid,flavorid)	Create a VM with the image ID and flavor ID
start_vm(self,serverid)	Start a VM
stop_vm(self,serverid)	Stop a VM
delete_vm(self,serverid)	Delete a VM
get_vmstat(self,serverid)	Get the VM status

(3) Remote Data Access System(CDAS) [10]

CDAS is a data sharing system based on streaming and cache, which provides cross-domain data access service in this system. It implements a unified data management module and an efficient data access especially in WAN environment.

3 LHAASO Daocheng Site

The Daocheng site is located Hai Zi mountain at an altitude of 4410 m. Considering there is no expertise in such a harsh environment, we constructed the computing system there using cloud-based architecture, which greatly helps to reduce the cost of operation and maintenance as well as make sure of the system availability and stability. The architecture of the remote site is shown in Figure 7. The Login farm and administrator nodes are managed by Openstack and Kubernetes[11]. Login farm runs within Docker[12] to get load balanced and administrator nodes run in VMs. This site contains about 576 CPU cores. It started running in production in Jan 2019 and has been working well.

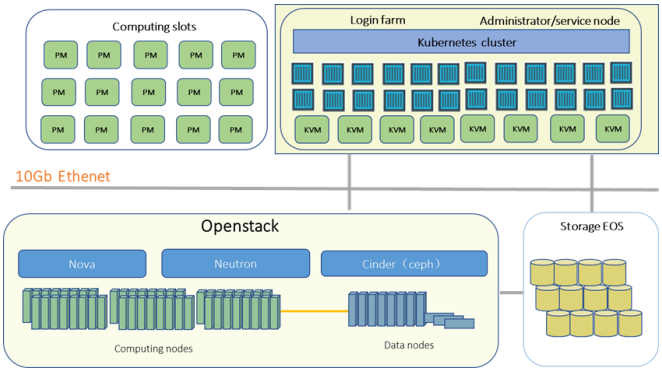


Figure 7. The architecture of HAASO Daocheng site

4 Running Status

The cloud-based computing system became in production in September 2014. The current resources of the system are shown in Table 2. This system runs well for the LHAASO experiment, which plays an important role in the LHAASO MC simulation and data generation in the early stage. From Oct 2017 to May 2019, the amount of completed jobs reached 2,439,090 and 5,305320 CPU hours were provided. We also developed an accounting system, which stores the data in Elastic Search and visualizes in Grafana. Figure 8 shows the completed jobs and CPU hours from May 28 to June 3, 2020.

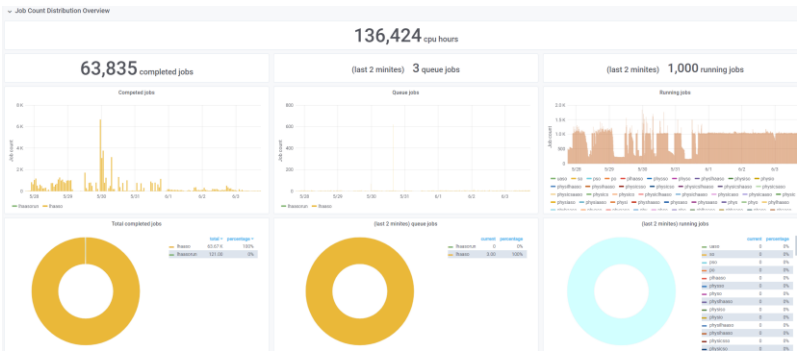


Figure 8. The completed jobs and consumed CPU hours from May 28 to June 3, 2020

5 Conclusions

In this paper, we presented the design and implementation of a cloud-based computing system for the LHAASO experiment. We introduce Openstack and HTCondor to integrate heterogeneous remote resources and make them work like one logical pool so that it can make job running in remote sites transparently. We also developed the cloud scheduler to provide resource provision dynamically so as to improve the overall resource utilization. With this solution, it's easy to integrate more resources from collaborated sites like universities, institutions and even the commercial cloud. The running status shows the system can support LHAASO experiment successfully. And it's easy to support other applications.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Contracts No. 11875283.

References

[1] LHAASO experiment：http://english.ihep.cas.cn/ic/ip/LHAASO.

[2] M. jun Chen, Z. Yao, B. Gao, B. Zhou, H. Wu, H. Li. R&D of LHAASO-WCDA. *32nd International Cosmic Ray Conference*, Beijing 2011.

[3] G.A. Adde, B. Chan, D. Duellmann, X. Espinal, A. Fiorot, J. Iven, L. Janyst, M. Lamannna, et al. Latest evolution of EOS filesystem. 2015 J. Phys.: Conf. Ser. 608 012009.

[4] Openstack : https://www.openstack.org/.

[5] HTCondor : https://research.cs.wisc.edu/htcondor/.

[6] Openstack multi-regions : https://docs.openstack.org/murano/pike/admin/appdev-guide/multi_region.html.

[7] Y.D Cheng, L. Wang, Q.L. Huang, J.Y. Shi. A Large Scale Data Management System for BESGrid. *The 9th International Conference on Grid and Cloud Computing (GCC 2010)*, Nanjing China.

[8] Condor-C model, The condor Grid Type.

https://research.cs.wisc.edu/htcondor/manual/v7.6/5_3Grid_Universe.html#SECTION00631000000000000000

[9] Q.L. Huang, H.B. Li, J.Y. Shi, S.Z Sun, W.J. Wen, Y.D. Cheng, Z.J Cheng. Openstack-based Virtualized Computing Cluster and Application for High Energy Physics. Computer Science,2017,**44**(10).

[10] Q. Xu, Z. Cheng, Y. Cheng, et al. Cross-domain Data Access System for Distributed Sites in HEP. Lecture Notes in Computer Science, 2019.

[11] Kubernetes : https://kubernetes.io/.

[12] W. Zheng, J.Y. Shi, Y.D. Cheng, Q.L Huang, X.F. Yan. Container Practice at IHEP. *International Symposium on Grids & Clouds 2019*(ISGC 2019).