ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, Vol 8(S8), 582-587, April 2015
DOI: 10.17485/ijst/2015/v8iS8/71712

mvcluster: Multiple Experiments Supportable Virtual Cluster System

Seo-Young Noh[1] and Sangho Ha[2*]

[1]National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information Daejeon, 305-806, South Korea,
[2]Department of Computer Science and Engineering, Soonchunhyang University, Asan, Chungnam, 336-745, South Korea; hsh@sch.ac.kr

Abstract

Cloud computing is emerging in industry because of its flexible resource management. Many private companies have started their new businesses that are reselling their unused computing powers. Such a trend is gradually impacting the traditional computing facilities used in scientific domains. In the high energy physics domain, many computing resources are required in general. Therefore it is always required to seek sharing resources and find outside available resources. The one of most challenges is how to stably share a queue among heterogeneous experiments. Sharing a queue is at risk of losing services because malicious work can deeply impact the entire system. In cloud computing, such a problem can be resolved with the feature of virtual machine isolation. We can utilize such a feature to support multiple experiments in a single computing farm. In this paper, we introduce *mvcluster* which is a multi-queue enabled virtual cluster system, which extends the functionalities of the single queue-based *mvcluster*. Our proposed approach can achieve service stability and flexible computing capacity increasing resource utilization.

Keywords: Mvcluster, Multiple Experiments Support, Virtual Cluster System

1. Introduction

Cloud computing provides many advantages to computing intensive scientific domain in the viewpoint of resource management. The feature of flexibility can make it possible to expand computing capability to the unlimited computing boundary in theoretical[1,2]. This feature is attractive to scientific domains such as the High Energy Physics (HEP), which requires huge amount of computing resources in general.

In HEP, it is well known issue to utilize unused and available computing resources geographically distributed in the world. Such willingness was the starting point that many new technologies in the domain have been developed; for example, grid is one of most successful examples and still actively used in real world[3].

Computing infrastructures operated in scientific laboratories like CERN[4], FNAL[5], BNL[6] have been moving toward cloud computing based infrastructures. The transition to cloud computing is because of the robustness against failures and resource utilization[7,8].

In traditional HEP computing environments, a computing farm is dedicated to a single experiment. Supporting multiple experiments in a single computing farm is a complex task because every experiment has its unique requirements. Such requirements can be conflict with the other experiments' requirements. The conflicts can affect system stability and result in system failures. Because of this high risk of failures, a single queue is used for a single computing farm. However, a single queue based computing farm has the issues of resource utilization because the computing farm is not always filled

*Author for correspondence

by jobs. Although there are available computing slots, it cannot support other experiments which are waiting for resource allocation. We might also consider multiple queue enabled system in a computing farm. However, a single system with multiple queues will face up the same problem of service stability.

In that sense, cloud computing can solve the traditional problem because of its robustness and flexibility. It is possible to enable multiple services in a single computing farm as well as in multiple computing farms. Virtualization technology in cloud computing makes it possible because it safely isolates virtual machines which are used for worker nodes in the traditional physical computing farms. In case virtual worker nodes go wrong, such failures are immediately removed by terminating the virtual machine. Restarting the virtual machine can simply resume its service without long delay of service readiness. This feature tremendously improves service stability. All virtual worker nodes are considered as independent machines. Therefore, it is possible that we can build multiple cluster systems in a single computing farm in order to support multiple experiments. In addition, it is also possible to create a virtually large single computing farm which consists of multiple physical farms geographically distributed.

In this paper, we will introduce a multi-queue enabled virtual cluster system called *mvcluster*. *mvcluster* is designed to support multiple experiments in a single computing farm which is built in cloud computing infrastructure[9]. The proposed virtual cluster system extends single queue-based vcluster architectur[10]. *mvcluster* can safely prevent single failure from destroying the entire computing farm by removing failed virtual machine. It can not only improve the utilization, but also guarantee the service stability. Such stability is nature in cloud computing because virtual machines in the virtual cluster systems are independent each other. They are not intervening in the service of each other. Therefore, interference between virtual machines is fundamentally prevented.

The rest of this paper is organized as follows: In Section 2, we will discuss the utilization problem in HEP domain with real examples. In Section 3, we will discuss the related work. In Section 4, we will revisit vcluster which is the base architecture of *mvcluster*. In Section 5, we will introduce *mvcluster* and show how it can effectively handle multiple experiments in a single system. In Section 6, we will summarize our work and future work.

2. Resource Utilization Problem

Data centers in the world are facing up resource utilization problems. According to an article shown in[8], computing resource utilization in general is very low. Because of this, many data centers are trying to share their computing recourses with other centers in order to overcome short computing resources as well as low utilization problem. Amazon EC2 service is most well-known cloud computing service reselling their computing resource to the public, increasing their profits.

There are many technologies introduced in last decade to improve resource utilization, but grid is one of the most popular technologies widely used in scientific domain especially in HEP[12]. Grid computing can be considered as a resource orchestration which collects resources and distributes jobs according to the resource availability. One of main differences compared to cloud computing is underlying utilization technologies. Grid computing focuses on idle resources distributed in multiple locations while cloud computing focuses on virtualization. Two technologies are different, but their main goal is the same in utilization viewpoint. In grid, the main approach is to open entire idle physical resources to the public. In contrast, cloud computing utilizes a physical system by generating virtual machines with hypervisors like KVM[13] and Xen[14].

Figure 1 to Figure 3 show how much CPU time has been used for CDF, STAR, and HCP experiments for 30 days in KISTI, respectively.

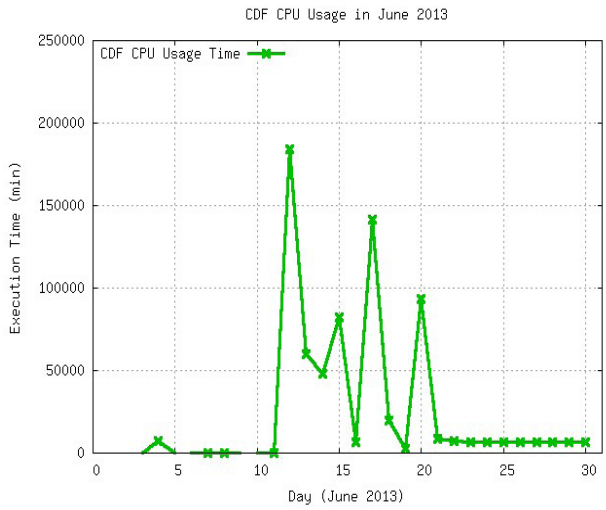

Figure 1. CDF CPU usage result in june 2013.

The three computing farms are supporting HEP experiments with 432, 1024 and 512 cores, respectively. The jobs performed in the farms are mixed with grid and local jobs.

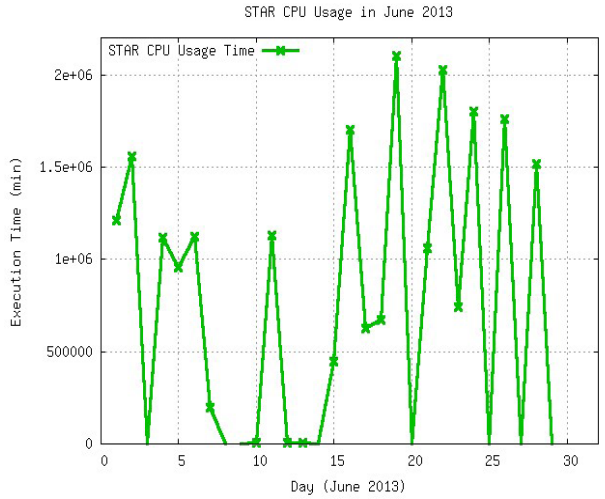

Figure 2. STAR CPU usage result in june 2013.

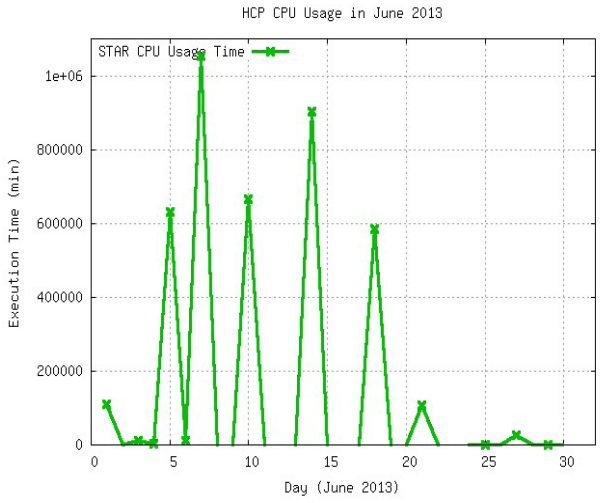

Figure 3. HCP CPU usage result in june 2013.

As shown in the figures, it is worth noting that the CPU usages of CDF for the first 10 days are relatively low compared to the other experiments. Therefore, we can expect the improvement of resource utilization if we can share the unused computing resources of CDF with HCP and STAR.

When combining resources to support multiple experiments, it is critical to maintain stable service. To address this problem, cloud computing can be adopted in service consolidation. To service multiple experiments in the consolidated farm, a virtual cluster system must handle multiple queues.

3. Related Work

It is not difficult to find many approaches to utilize cloud computing technologies in scientific domains. Such approaches are natural in computing intensive research fields. In HEP, Gable et. al[15] proposed a dynamic cloud batch system called Condor Scheduler which can combine multiple cloud sites. As noted in the name, their approach tried to extend the capability of HTCondor batch system.

Ivan Krsul et. al[16] introduced VMPlants and VMShop which can manage virtual machines for grid computing. VMPlants and VMShop are communicating each other in virtual machine management. VMShop essentially sends virtual machine related commands to VMPlants. VMPlants then executes the commands such as creation, querying, destroying of virtual machines. As users specify necessary services or packages, the proposed approach generates a Direct Aacyclic Graph (DAG) for the requests. Once the DAG is generated, then it tries to find a virtual image which includes all necessary packages or service expressed by DAG. Within this framework, we can expect a fast service if a proper image is already prepared.

Because of flexibility and service stability, cloud computing is attractive to many scientific researchers. Therefore, many research institutes have been building own private cloud facilities or moving to cloud computing based infrastructure.

4. Revisiting *vcluster*

vcluster is a virtual cluster system being developed in the joint project between KISTI and FNAL since 2011. It is able to create a virtual cluster system on top of multiple cloud platforms including public and private cloud solutions. The design concepts behind vcluster are including simplicity and agnostic approach to cloud and batch systems. vcluster provides a framework and control functionalities are implemented through plug-ins. Such an approach can reduce the complexity of working with multiple solutions of cloud and batch systems. As long as a proper plug-in is developed, any underlying cloud and batch system can be integrated to vcluster framework. Figure 5 shows the overall concept of vlcuster.

As shown in Figure 4, vcluster can communicate with cloud solutions through plug-ins. For example, a cloud solution using Open Nebula can communicate with vcluster through open nebula plug-in when creating, stopping, and terminating virtual machines. For Amazon EC2, amazon-ec2 plug-in is used to work with Amazon cloud service. In addition to cloud plug-ins, vcluster is also cooperating with underlying batch systems using plug-ins. In case of HTCondor[17], we use htcondor plug-in of vcluster while pbs plug-in for PBS batch system[18].

Plug-in based architecture makes it possible to simplify the interface between vcluster and complex underlying solutions and provide a homogeneous view to the system administrator. Actual users cannot distinguish where their jobs are executed because vcluster creates a virtual cluster which is shown as a single large cluster system to users.

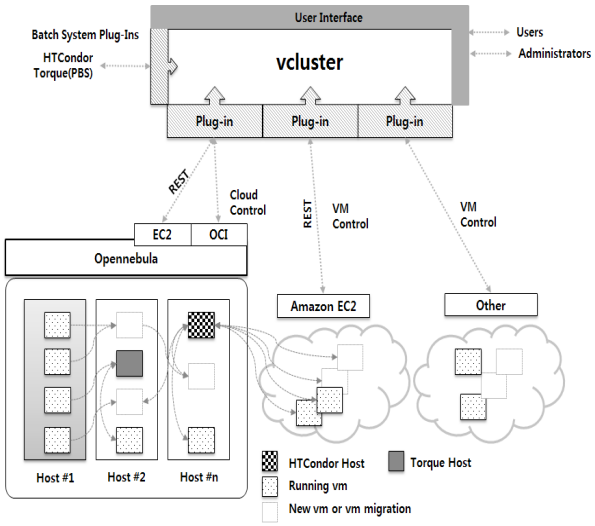

Figure 4. Conceptual diagram of *vcluster*[9].

vcluster is simply considered as a tool for management of virtual machines which construct a single logical large farm running over multiple cloud solutions. It can create virtual machines and the created virtual machines automatically join to the virtual cluster system. Therefore, the cluster system can be scaled up or down seamlessly without suspending services.

5. *mvcluster*: Multi-Queue Enabled *vcluster*

TIn this section, we will discuss *mvcluster* which is the extended version of vcluster in order to support multiple experiments in a single virtual cluster which is consisting of multiple cloud solutions.

vcluster can create a single virtual cluster system consisting of multiple virtual machines which may be initiated in different cloud platforms. One of short comings of vcluster is that it can only support a single queue which means it can only support a single experiment at the same time in HEP perspective. If we can launch multiple virtual cluster systems over a consolidated cloud platform, we can achieve not only stability, but also utilization. In order to mediate such a limitation, we need to enhance vcluster to multi-queue supportable one. In this paper, we call it *mvcluster*.

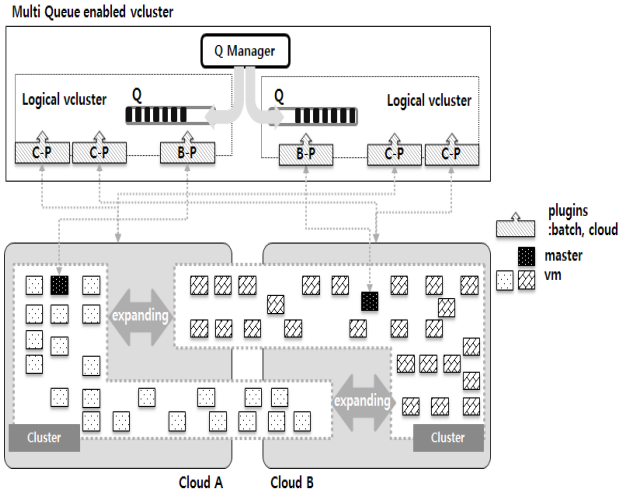

Figure 5. Multi queue enabled virtual cluster system[6].

Figure 5 shows the conceptual architecture of *mvcluster*. It can handle multiple queues in a single cluster system. Such a feature can improve resource utilization by launching virtual machines in resource available cloud computing platforms or migrating virtual machines from high to low utilized clouds. One major extension is the capability of managing multiple queues which safely handles the consolidated cloud computing resources, which can again provide better utilization in the viewpoint of system administration as well as better service resources if one of services is not fully utilized.

As shown in Figure 5, a virtual cluster in Cloud A can be expanded to Cloud B while a virtual cluster in Cloud B can also be expanded to Cloud A depending on resource availability. If only single queue enabled virtual cluster system is allowed in cloud platforms, it is hard to extend their capability to other cloud platforms when

multiple experiments are being supported because there is no central manager to handle requests from multiple clusters.

When supporting multiple experiments with multiple cluster systems which are managed with multiple queues, it is important to share multiple cloud platforms running in different geographical locations. If we only consider a single queue and a single experiment, we are not able to efficiently utilize available resources in other clouds. Therefore, efficient management of multiple queues in *mvcluster* is important to seamlessly handle multiple clouds with keeping high utilization and continuing stable service.

Algorithm 1 shows how *mvcluster* can effectively manage multiple virtual cluster systems by handling multiple queues.

Algorithm 1: Multi-Queue Management

```
Procedure mq-management
        /* iterate all queues in cluster systems */
        while queue←getNextQueue() do
            rate←getWaitingRate(queue)  /* waiting jobs? */
            if rate≤ threshold do
            continue
            end if
            cloud←get Available Cloud() /* find a cloud */
            if cloud == null do  /* no available cloud */
              waiting    /* wait until resource is available */
            end if
            /* launch virtual machine in a cloud */
            launch VM(cloud, queue)
        end while
end procedure
```

The algorithm first checks if there are idle jobs while looping all queues in the system. If there are waiting jobs in a queue, then it selects a queue to handle the jobs. In this point, it is important to not burst virtual machines just because there are waiting jobs. We have to consider if the waiting rate is over the threshold value which is a guide value when bursting virtual machines. If the rate is less than threshold, the queue is not busy. Therefore, we just continue checking the other queues. If there are too many jobs in the queue and the rate is over the threshold, then we have to find a cloud where to request to launch virtual machines. If there are no available clouds, then we have to wait until there is available cloud found. If we find a cloud to launch virtual machines, then we request the cloud to run virtual machines. Since the queue information is provided to a cloud, the cloud knows which types of virtual machine should be launched. The newly launched virtual machine will automatically join the proper cluster system which is managed with the queue.

6. Discussions and Conclusions

In this section, we will discuss *mvcluster* which is the extended version of vcluster in order to support multiple experiments in a single virtual cluster which is consisting of multiple cloud solutions.

The multi-queue enabled vcluster system introduced in this paper utilizes cloud technologies to support multiple experiments in a single logical large computing farm. Service stability and extendibility are always being sought from scientific research domains especially in HEP. In that sense, cloud computing effectively mitigates the pains by flexibly extending computing capacity as well as providing sable service by isolating virtual machines from physical systems.

One of challenges when utilizing cloud solutions in HEP is how to efficiently handles multiple experiments. Since each experiment uses its own queue, it is difficult to extend its cluster system to other clouds or improve resource utilization. In order to resolve this problem, *mvcluster* proposed in this paper seamlessly manages multiple queues in virtual cluster systems which are running over multiple clouds. Enabling multiple queues is important to achieve maximum resource utilization in consolidated clouds. In our historical data discussed in Section 2, the three computing farms can be consolidated and provided better utilization as shown in Figure 6.

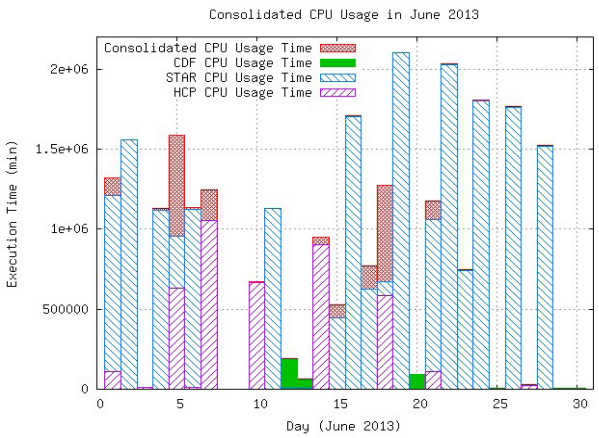

Figure 6. Consolidated resource for CDF, STAR, and HCP.

From the figure 6, we can conclude that less computing resource can safely service for CDF, STAR, and HCP simultaneously if we can manage these three computing farms depending on the number of jobs. Therefore, *mvcluster* can help HEP applications to provide a scalable service as well as stable service with continuing the supports of multiple experiments.

In future work, we are planning to apply *mvcluster* to real experiments introduced in CDF, STAR and HCP with the consolidated cloud computing farm.

7. Acknowledgement

This work was supported by the program of the Construction and Operation for Large-scale Science Data Center (K-15-L01-C05-S01).

8. References

1. Khaskheli M, Jamali A, Arain MA, Nizamani AH, Soomro AH, Arain HH. Chemical and sensory quality of indigenous milk based product 'Rabri'. Pakistan J Nutr. 2008; 7(1):133–36.
1. Sotomayor B, Montero RS, Foster I. Virtual Infrastructure Management in Private and Hybrid Clouds. IEEE Internet Computing. 2009; 13(5):14–22.
2. Kansal A, Zhao F, Liu J, Kothari N, Bhattacharya AA. Virtual machine power metering and provisioning. Proceedings of the 1st ACM symposium on Cloud Computing. Indiana, USA; 2010.
3. Foster I, Kacsu P. Editors' message. Journal of Grid Computing. 2001; 9(1):1–2.
4. CERN; 2015. Available from: http://www.cern.ch.
5. FNAL; 2015. Available from: http://fnal.gov.
6. BNL; 2015. Available from: http://bnl.gov.
7. Wu H, Ren S, Garzoglio G, Timm S, Bernabeu G, Kim W, Chadwick K, Jang H, Noh S-Y. Proceedings of International Conference on Parallel and Distributed Systems. Seoul, Korea; 2010.
8. Foster I, Zhao T, Raicu I, Lu S. In CoRR; 2015. Available from: http://arxiv.org/abs/0901.0131.
9. Noh S-Y, Jang H. Multi-queue enabled virtual cluster system in cloud computing platforms. Proceedings of PlatCon. Jeju, Korea; 2014.
10. Noh S-Y, Timm S, Jang H. *mvcluster*: a framework for auto scalable virtual cluster system in heterogeneous clouds. Cluster Comput. 2014; 17(3):741–9.
11. Pawlish M. Analyzing utilization rates in data centers for optimizing energy management. Proceedings of Green Computing Conference. San Jose, CA, USA; 2012.
12. Kesselman FC, editor. High performance computing: from grids and clouds to exascale. Cetraro, Italy; 2010 June.
13. KVM; 2015. Available from: http://www.linux-kvm.org.
14. Xen; 2015. Available from: http://xen.org.
15. Gable A, Agarwal M, Anderson P, Armstrong A, Charbonneau R, Desmarais K, Fransham D, Harris R, Impey C, Leavett-Brown M, Paterson D, Penfold-Brown W, Podaima R, Sobie M, Vliet. A batch system for HEP applications on a distributed IaaS cloud. J Phys. 2010; 6:331.
16. Krusl A, Zhang GJ, Fortes JAB, Figueiredo RJ. VMPlants: providing and managing virtual machine execution environments for grid computing. Proceedings of Supercomputing. Pittsburgh, PA, USA; 2004 Nov 6–12