COMMUNITY CLUSTER OR COMMUNITY CLOUD?
Utilizing our Own Bare-metal

Xin Fan, Yusuke Wada and Shigeru Kusakabe
Graduate School of Information Science and Electrical Engineering, Kyushu University
744, Motooka, Nishi-ku, Fukuoka city, 819-0395, Japan

Keywords: Cloud computing, Private cloud, Community cluster, Hadoop, MPI.

Abstract: The increasing availability of cloud computing technologies enables us to have an option we had not before: using private cloud as well as using public cloud. In this paper, we report our ongoing work on examining effectiveness of private cloud computing in an academic setting. Many researchers have examined the relative computational performance of commercially available public cloud computing offerings using HPC application benchmarks. As one of the driving forces in using cloud technologies is cost effectiveness, some researchers have examined public cloud offerings and their HPC environment, a community cluster, from a view point of cost-performance. Part of the conclusions indicates their community cluster may be favorable for typical community members. Due to the similar grounds of community cluster, we expect private (or community) cloud is promising in academic settings. Academic community members may also have interest in utilization of their resources with a configuration of less constraints compared to public cloud offerings while receiving benefit of cloud technologies. In this paper, we discuss the situation we are managing a number of bare-metals and we are deciding whether we configure the computing resource as a cluster of bare-metal nodes or as a cluster of virtual machines by using cloud computing technologies. According to our preliminary evaluation results, while we can easily reinstall and change the software framework on clusters in our private cloud, we must be ready for occurrence of unexpectedly severe performance degradation.

1 INTRODUCTION

Cloud computing has emerged as a new paradigm for using computing resources. We do not have the single definition of cloud computing so far, but most definitions share common characteristics(Armbrust et al., 2009):

1. The illusion of infinite computing resources available on demand, thereby eliminating the need for cloud computing users to plan far ahead for provisioning;
2. The elimination of an up-front commitment by cloud users, thereby allowing organizations to start small and increase hardware resources only when there is an increase in their needs; and
3. The ability to pay for use of computing resources on a short-term basis as needed and release them as unneeded, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

The on-demand and pay-as-you-go style seems to offer a flexible and cost-effective method to use computing resources.

From the view point of academic computing, many researchers have examined the relative computational performance of commercially available public cloud computing offerings using a number of standard benchmarks and HPC applications. Most studies used Amazon EC2 as the representative of commercially available could offerings(Jackson et al., 2010), while we have other options such as private cloud.

Since one of the driving forces in using cloud technologies is cost performance, some researchers have also examined public cloud offerings and their HPC environment, a community cluster, from a view point of cost-performance(Carlyle et al., 2010). A community cluster is a system obtained by a faculty group and centrally operated by an institution, maintained for the benefit of the many research groups that own the nodes in the cluster. Community cluster users gain peace of mind from the cluster's operation by professional IT staff; low overhead from centralized power, cooling, and data center space; and cost effectiveness from the combined purchasing power of all cluster owners and strategic sourcing of the cluster hard-

Fan X., Wada Y. and Kusakabe S..
COMMUNITY CLUSTER OR COMMUNITY CLOUD? - Utilizing our Own Bare-metal.
DOI: 10.5220/0003450501270130
In *Proceedings of the 1st International Conference on Cloud Computing and Services Science* (CLOSER-2011), pages 127-130
ISBN: 978-989-8425-52-2

ware. From the institutional perspective, community clusters are cost-effective way for faculty to obtain HPC resources. In the case-study, researchers at Purdue University tried to measure per node hour cost of cloud offering and the traditional HPC environments, their community cluster, in doing scientific computing. Part of the conclusions indicates their community cluster may be favorable for typical community members. The community cluster of the case study at Purdue is configured for scientific computing. We consider it is better to flexibly accommodate emerging computing frameworks such as Hadoop(Hadoop,) in order to broaden and enhance the advantageous aspects of community clusters.

Cloud computing technologies offer new styles of computing in various activities using computing resources including academic activities. The growing availability of cloud computing technologies enables us to have an option we had not before: using private cloud as well as using public cloud offerings. According to (Armbrust et al., 2009), cloud computing is the sum of SaaS and utility computing, but does not normally include private cloud, which is the term to refer to internal data-centers of a business or other organization that are not made available to the public. From the view point of economies of scale, cloud systems of larger scale are more advantageous than those of smaller scale. While private cloud seems less promising than public one from this view point, there exist various factors in making a decision. Due to the similar grounds of the community cluster, we expect private (or community) cloud can be promising in academic settings.

In this paper, we discuss the situation we are managing a number of bare-metals and we are choosing whether we configure the computing resource as a cluster of bare metal nodes or as a cluster of virtual machines by using cloud computing technologies. One of the driving forces other than cost effectiveness in using cloud technologies is its flexibility. Based on the cloud computing technologies, we can prepare different kinds of computational environment, deploy a specific environment as we choose over virtual machines, and release the resource after the predefined period according to the reservation schedule.

In this paper, we introduce our ongoing work on examining practical effectiveness of private cloud computing in an academic setting. The rest of this paper is organized as follows. Section 2 explains outline of our private cloud. Section 3 shows our preliminary evaluation results.

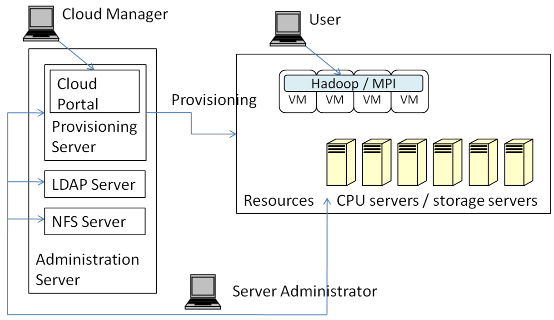

Figure 1: Overview of our private Cloud.

2 OUTLINE OF OUR PRIVATE CLOUD

In our study, we use a small version of IBM Blue-Cloud as our private cloud computing platform. Figure 1 shows the outline of our cloud. Followings are main features of the cloud:

- **Virtualization.** In our cloud platform, we can dynamically add/delete server machines to/from resource pool, if the bare-metal machines are x86 architecture and able to run Xen. In adding a new server to the resource pool in cloud, we connect the bare-metal server to the private network of the cloud. Then, host OS Domain 0 (Dom0) of Xen is automatically installed through the network boot mechanism. We can deploy virtual machines over the host OS machines.
- **Provisioning.** When a user requests a computing platform from the cloud portal web page, he/she can specify the virtual OS image (Domain U (DomU) of Xen in our platform) and applications from the menu, in addition to the virtual machine specification such as the number of virtual CPUs (VCPUs), the amount of memory and storage within the capacity of the cloud resource. In our cloud, the number of VCPUs is limited within the number of physical CPUs in order to guarantee the minimum performance of DomU. When the request is admitted, the requested computing platform is automatically prepared.

In addition to cloning the virtual machines of the same machine image For example, our cloud supports automatic set up of a Hadoop programming environment in fully distributed-mode when provisioning computing resources. We usually need following steps to set up a Hadoop environment on a cluster:

1. Installing a base machine image into nodes
2. Installing Java

3. Mapping IP address and hostname of each node
4. Permitting non-password login from the master machine to all the slave machines
5. Configuring Hadoop on the master machine
6. Copying the configured Hadoop environment to all slave machines from the master machine

We explain corresponding steps to set up a Hadoop environment on our private cloud. First, if we need to increase the machine resource of our cloud, we set new bare-metal machines network-bootable and connect them to the local network of our cloud. The machines are automatically arranged to be a part of our cloud. We have to prepare the desired machine image. Then, we request a Hadoop environment through the portal, and the following process are arranged automatically. We need an extra script as a part of preparation if we want to implement a specific configuration in the postscript phase, such as a master/slave configuration for the Hadoop environment.

Thus, by adopting private cloud computing, we can use labor-reducing mechanisms that are not available in community cluster.

3 PRELIMINARY EVALUATION

In order to evaluate the effectiveness of our private cloud, we prepared two types of platforms: One is a cluster of eight bare-metal servers as a representative of community cluster and the other is a cluster of eight virtual machines in the private cloud mapped onto eight bare-metal servers. We used Dell blade server PowerEdge M600 with Intel Xeon L5410 processor as bare-metal servers. We used two software framework: MPI for numerical computation workloads and Hadoop for emerging non-numerical computation workloads.

3.1 MPI

We evaluated a thermal convection solver with MPICH, an implementation MPI as a numerical parallel computation workload. The data elements were generated by Adventure sFlow, one of modules included in the ADVENTURE project(Kanayama et al., 2005). ADVENTURE sFlow uses the Newton method as the nonlinear iteration, and to compute the problem at each step of the nonlinear iteration a stabilized finite element method is introduced. In this experiment, we measured execution time in changing the number of steps.

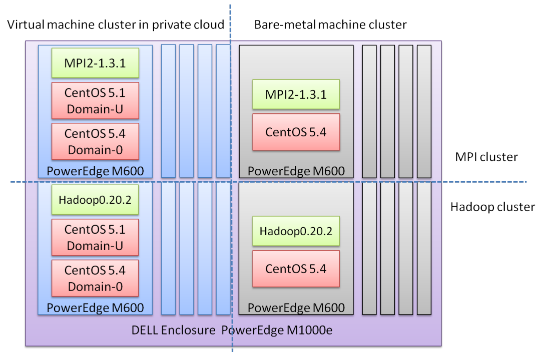

Figure 2: Clusters on virtual machines / bare-metals for MPI/Hadoop.

We show the result in Figure 3 and Table 1. As we see from the results, performance degradation incurred by virtualization in our cloud for this benchmark are around 20% although virtualization is one of the inevitable cloud-enabling technologies.

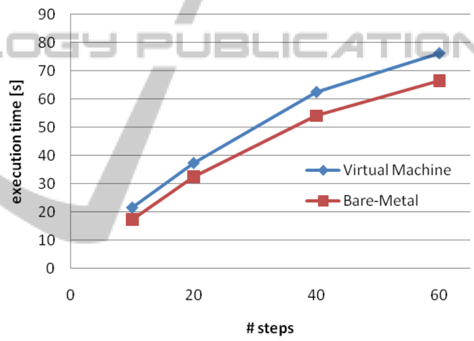

Figure 3: Thermal convection solver execution time.

Table 1: Thermal convection solver execution time (sec).

# steps	10	20	40	60
Bare-Metal	17.33	32.46	54.05	66.52
Virtual Machine	21.62	37.38	62.48	76.20

3.2 Hadoop

We evaluated TestFDSIO benchmark included in the Hadoop distribution as a workload of emerging parallel and distributed applications. Table 2 and Figure 4 show the results. The experiment options were random reading 1MB files, changing the number of files 10 to 50. As seen from the results, throughput of reading files in the virtualized environment in our cloud was constantly degraded to about two-third compared to that of bare-metal environment.

As another experiment, we evaluated π calculation included in the Hadoop distribution. We measured execution time while changing the number of map tasks.

Table 2: Throughput of TestFDSIO benchmark (mb/sec) (random read, file size 1MB, the number of files 10 to 50).

# files	10	20	30	40	50
Bare-Metal	32.57	35.09	33.11	34.04	34.18
Virtual Machine	20.37	21.81	20.83	21.42	20.24

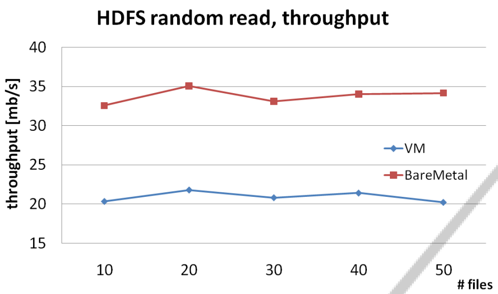

Figure 4: Throughput of TestFDSIO benchmark.

As we can see from the results in Table 3 and Figure 5, performance degradations of the private cloud version were very severe and the situation became worse as the number of map tasks increases. The combination of the behavior of this MapReduce application and the low performance of network interfaces of virtual machines is one of the potential bottleneck. Although we have a plan of performance debugging to alleviate the problem, such kind of extra work may degrade the merit of labor-reducing effect in our private cloud.

Table 3: Execution time for π estimator (sec).

# map tasks	20	40	60	80
Bare-Metal	29.41	37.47	46.57	58.56
Virtual Machine	225.40	465.33	868.00	1119.08

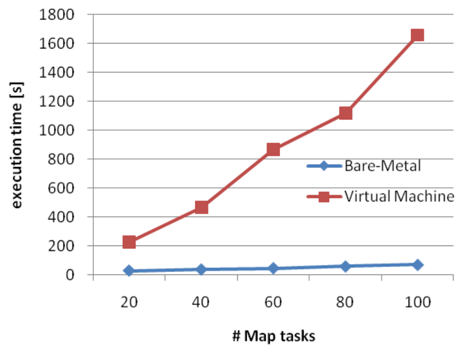

Figure 5: Execution time for π estimator.

4 CONCLUDING REMARKS

Due to cloud computing technologies that are not available in community cluster, we expect private (or community) cloud is more promising than community cluster in some academic settings. While we can easily reinstall and change the software framework on the cluster by using labor-reducing mechanisms in private cloud, the performance degradation may be more severe than expected. While the solution depends on the user pattern, building cluster of bare-metal machines seems more rewardful when users are performance-oriented. Our future work includes automatic performance tuning applicable to our private cloud.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing. Technical report, UCB/EECS-2009-28, Reliable Adaptive Distributed Systems Laboratory.

Carlyle, A. G., Harrell, S. L., Smith, P. M., and Center, R. (2010). Cost-effective hpc: The community ot the cloud? *2nd IEEE International Conference on Cloud Computing Technology and Science*, pages 169–176.

Hadoop (—). As of Feb.1, 11.

Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H. J., and Wright, N. J. (2010). Performance analysis of high performance computing applications on the amazon web services cloud. In *2nd IEEE International Conference on Cloud Computing Technology and Science*.

Kanayama, H., Tagami, D., and Chiba, M. (2005). Stationary incompressible viscous flow analysis by a domain decomposition method. *Domain Decomposition Methods in Science and Engineering XVI*, pages 611–618.