Performance Evaluation of Scientific Workflow on OpenStack and OpenVZ

Amol Jaikar[1,2(✉)], Syed Asif Raza Shah[1,2], Sangwook Bae[2], and Seo-Young Noh[1,2]

[1] Grid and Supercomputing Department, Korea University of Science and Technology, Daejeon 305-350, South Korea
{amol,asif,rsyoung}@kisti.re.kr
[2] National Institute of Supercomputing and Networking, Korea Institute of Science and Technology Information, Daejeon 305-804, South Korea
wookie@kisti.re.kr

Abstract. Cloud computing is capturing attention of the market by providing infrastructure, platform and software as a services. Using virtualization technology, resources are shared among multiple users to improve the resource utilization. By leasing the infrastructure from public cloud, users can save money and time to maintain the expensive computing facility. Therefore, it gives an option for cluster and grid computing technology which is used for industrial application or scientific workflow. Virtual machine enables more flexibility for consolidation of the underutilized servers. However, containers are also competing with virtual machine to improve the resource utilization. Therefore, to adopt cloud computing for scientific workflow, scientist needs to understand the performance of virtual machine and container. We have used cloud computing with different virtualization technologies like KVM and container to test the performance of scientific workflow. In this work, we analyze the performance of scientific workflow on OpenStack's virtual machine and OpenVZ's container. Our result shows that container gives better and stable performance than virtual machine.

Keywords: Performance · Scientific workflow · OpenStack · OpenVZ

1 Introduction

Cloud computing is a new flexible model of loosely coupled distributed system. Cloud computing technology provides infrastructure, platform and software as a service to the users [1]. Software and hardware techniques for virtualization enable cloud computing technology [2] to create virtual infrastructure. Moreover, hardware companies are also looking forward to support the virtualization [3]. Due to the flexible nature of cloud computing, server consolidation technique is used to reduce the power consumption of a data center. Peak clustering based placement technique gives better performance than correction based placement

Y. Zhang et al. (Eds.): CloudComp 2015, LNICST 167, pp. 126–135, 2016.
DOI: 10.1007/978-3-319-38904-2_13

technique for server consolidation [4]. Because of these advantages, research organizations and industries are trying to adopt cloud computing technology.

Research organization requires considerable amount of computing infrastructure to calculate scientific experiment's results. This requirement varies frequently depending on the scientific project. Therefore, flexible infrastructure is more suitable for scientific world. In order to adopt cloud computing, it is compulsory to understand the performance difference between physical and virtual infrastructure. Virtual machines and containers are available options to adopt cloud computing for scientific workflow.

In terms of virtual machines, there are different type of hypervisors like KVM [5], Xen [6] and VMware [7]. We have used KVM for our implementation. To manage the virtual machines, we have used OpenStack [8] as a middleware. In case of containers, we have used OpenVZ [9].

The paper is organized as follows. Section 2 gives the background information including virtualization and batch processing software. Furthermore, Sect. 3 describes the related work. Section 4 covers the implementation part of this paper which include the execution of jobs. Section 5 discusses about suitability for scientific workflow. Lastly, Sect. 6 concludes the paper.

2 Background

This section gives the background information of technologies used in this experiment which includes virtualization and batch processing software.

2.1 Virtualization

Virtualization technology enables to creation of virtual machine on physical machine with CPU, memory, network, operating system for providing different types of services. There are three types of virtualization; full, para and operating system level (OS-level) virtualization [10].

Full virtualization technology provides environment to run virtual machine with unmodified operating system. It includes the simulation of all features of hardware to run the virtual machine. Complete full virtualization is based on hardware support provided by *AMD* and *Intel* by storing guest virtual machine state into VT-x or AMD-V [11]. KVM is an example of full virtualization. We have used KVM as a hypervisor for OpenStack. In case of para-virtualization, the guest operating system needs to modify to run on hypervisor. This technology gives better performance than full virtualization [12]. Xen is an example of para-virtualization technology.

In OS-level virtualization, the kernel of host operating system provides the isolated user space to run different environment like containers, virtualization engines or virtual private servers. Container is a lightweight virtualization technology which enables to run many isolated instances of same host operating system. This technology does not simulate all hardware environment. It also requires fewer CPU and memory to run the virtual environment (container) [13]. We have used OpenVZ as a container technology for testing scientific workflow.

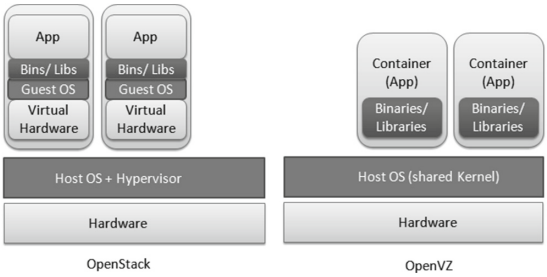

Fig. 1. OpenStack and OpenVZ Architecture

Virtual Machine. In case of virtual machine, hypervisor runs along with host operating system. This hypervisor simulates the virtual hardware including CPU, memory and network. The virtual machine runs on this hypervisor with its own operating system called as guest operating system. Therefore, instruction execution must pass through guest operating system as well as hypervisor which lead to an overhead. Nested page table technique has huge overhead for memory access [14]. We have used KVM as a hypervisor for OpenStack middleware. OpenStack is responsible for management of the virtual infrastructure including creation, deletion, migration and many more operations of virtual machine. It provides simple interface for user to interact with the virtual infrastructure. OpenStack is scalable, flexible and compatible for most of the hypervisors [15].

Container. Virtual machine comes with an overhead. In order to avoid this overhead, container-based virtualization technology provides lightweight alternative [16]. Hypervisor works at hardware level while container-based virtualization isolates the user space running at operating system level. Containers run at operating system level by sharing the host operating system kernel. Therefore, container-based virtualization has comparatively less isolation than hypervisor. Restriction of resource usage feature is available in containers which enables the fair usage of resources among all containers. Figure 1 refers the OpenVZ container architecture. OpenVZ system uses process ID (PID) and inter process communication (IPC) namespace to isolated the process contexts. OpenVZ also provides the network namespace for better network performance [17].

2.2 HTCondor

HTCondor is a non-interactive high throughput computing batch processing software which is developed by University of Wisconsin [18]. It has master-slave architecture where master is responsible for management of jobs and slave is responsible for execution of jobs. Figure 2 shows the logical view of HTcondor environment. HTCondor is used in this experiment for scientific job submission. The jobs, which are submitted by different user, are executed by slaves (OpenStack's virtual machine and OpenVZ's container).

3 Related Work

Jianhua Che et al. described the performance evaluation on different virtualization technologies by using standard benchmarks with the consideration of high performance computing [19]. Nathan Regola et al. also showed the performance comparison for high performance system [20]. Igli Tafa et al. have evaluated the transfer time, CPU consumption and memory utilization using FTP and HTTP protocol [21]. Igli Tafa et al. have also showed the virtual machine migration performance [22] for different virtualization technology. Our contribution is to evaluate the performance of high throughput computing scientific workflow on virtual infrastructure.

4 Implementation

This section gives the description of not only experimental setup but also scientific jobs.

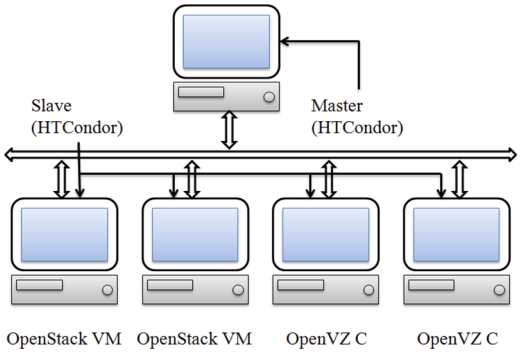

Fig. 2. Cloud System Architecture

4.1 Setup

In this testing environment, we used four servers of the same configuration of 6 cores with 2660 MHz of frequency and 24 GB of RAM with Scientific Linux distribution. Two among four servers are OpenStack controller and compute. We have used two-node architecture for OpenStack. Third server is used for OpenVZ and last server is for HTCondor master. To test the performance, we have used OpenStack virtual machine with HTCondor installed on Scientific Linux image. In case of OpenVZ, we have used containers with HTCondor installed on Scientific Linux template. Figure 2 shows the logical view of the HTCondor virtual cluster where OpenStack virtual machines and OpenVZ containers are working as a slave.

In order to test the performance of the scientific jobs, we characterized them into CPU, I/O, memory and network intensive jobs. Figure 3 shows the performance of scientific jobs using *top* command. Process id (PID) 3204 is a CPU intensive job while PID 3277 is a I/O intensive job. In case of PID 3080, it consumes more memory while PID 3254 is a network intensive job. Virtual machine and container are of the size 1 vCPU, 2 GB RAM and 20 GB HDD. In this experiment, we have executed 10 jobs on 2, 6 and 10 virtual machines and containers to analyze the performance of scientific workflow. These virtual machines and containers are on two different physical machines.

OpenStack(2VM) means OpenStack's 2 virtual machines are executing these 10 jobs to calculate the result. In case of *OpenVZ(2C)*, OpenVZ's 2 containers are executing these 10 jobs simultaneously. Number of jobs completed per unit time is the main attribute of high throughput computing.

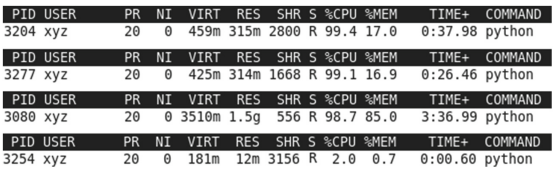

Fig. 3. Scientific job's performance

4.2 CPU Intensive Job

This job does the matrix multiplication and consumes more CPU cycles. As per Fig. 3, it(PID 3204) consumes around 100 % of CPU. We executed 10 CPU intensive jobs on multiple virtual machines and containers. Figure 4 shows execution time of these jobs. X-axis shows the number of the job and Y-axis shows the execution time. When we tested on 2 virtual machines of OpenStack and 2 containers of OpenVZ , the performance is almost identical to the physical host. But OpenStack virtual machine gives non-stable results. In case of 6 virtual machines and 6 containers, OpenVZ containers perform better than OpenStack virtual machines. Same results occur when testing on 10 virtual machines and 10 containers. But in this case, the execution time gets double than physical host which is considerable overhead. From the Fig. 4, we can analyze that OpenVZ gives better and stable performance than OpenStack.

4.3 I/O Intensive Job

This job creates the file on the HDD and writes data into the file. This job creates around 1 GB file. Figure 3 shows that it(PID 3277) consumes around 100 % of CPU and 16 % of memory. This job is to test the I/O performance of virtual machine as well as container. Figure 5 shows the execution time of these jobs. X-axis shows the number of the job and Y-axis shows the execution time in

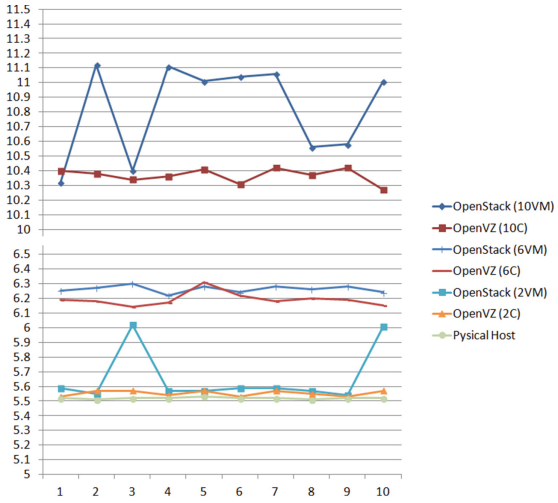

Fig. 4. CPU performance

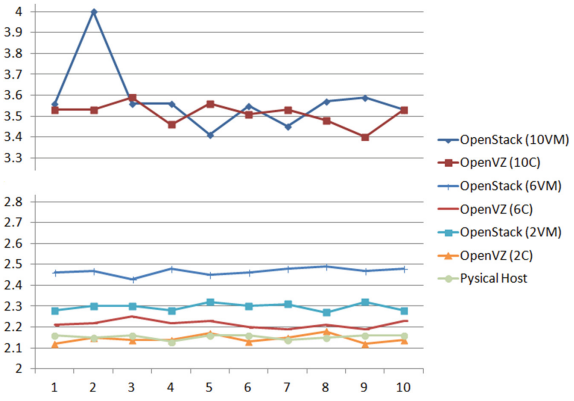

Fig. 5. I/O performance

minutes of that job. OpenVZ container always takes less time than OpenStack virtual machine, to execute the I/O intensive job. In case of 10 virtual machines and containers, container gives stable performance than virtual machine.

4.4 Memory Intensive Job

Memory intensive job consumes more memory to calculate the results. This job creates the string and append it with number of other strings continuously. Figure 3 shows that, it(PID 3080) consumes 85 % of the memory to get the result. Figure 6 shows the execution time of the memory intensive jobs. In this

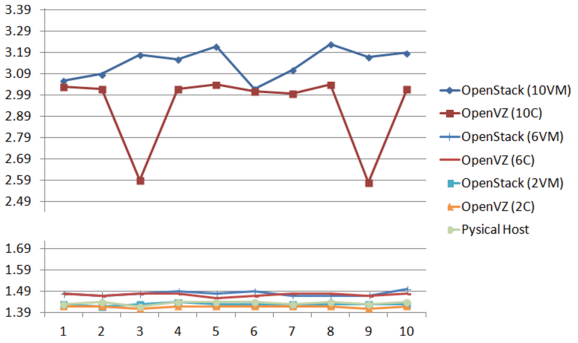

Fig. 6. Memory performance

case also, container performs better than virtual machine. But in case of 10 virtual machines and 10 container, execution time gets double than physical host which is also considerable.

4.5 Network Intensive Job

In this network intensive jobs, it continuously download the file from server and writes on the hard disk. Figure 3 shows that it(PID 3254) consumes less memory and CPU. Figure 7 shows the execution time of the network intensive jobs. The results are scattered around the numbers from 13 to 23 min, it may be because of network traffic. In case of virtual machines and containers, execution time is surprisingly less than physical host's execution time. It may be because of cache.

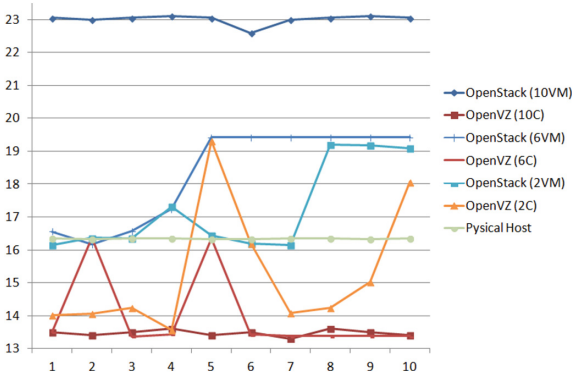

Fig. 7. Network performance

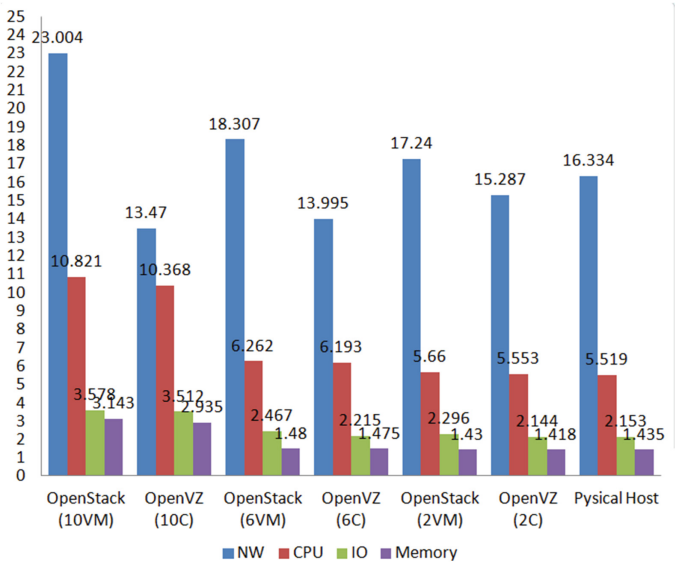

Fig. 8. Average execution time

5 Discussion

In order to understand the exact difference between OpenStack's virtual machine and OpenVZ's container, we took an average of 10 job's execution time. Figure 8 shows the average execution of network, CPU, I/O and memory intensive jobs. This graph clearly shows that containers gives better performance than virtual machine. Therefore, containers give better throughput than virtual machine.

In the scientific world, every user is authenticated by set of processes. Therefore, only authorized person can use this computing infrastructure of scientific world. Most of the computing infrastructure uses Scientific Linux as an operating system. Moreover, the scientific data is also accessible to all these authorized scientists. Therefore, containers are more suitable for scientific world than virtual machine. Moreover, containers are faster and stable in performance than virtual machines. The advantage of virtual machine is, different operating system virtual machines can reside on single physical machine. But in case of scientific world, most of them are using Scientific Linux distribution.

6 Conclusion

Cloud computing enables more opportunities for the scientific workflow. Resources can be leased from the public cloud when needed. This feature empowers more flexibility for the users to save money and time to maintain the computing infrastructure. There are two options for scientific community to adopt

cloud computing technology for scientific workflow; virtual machine and containers. Thus, in this paper we tried to understand the performance of scientific workflow on virtual machine and container. To investigate, we first characterized the scientific workflow into CPU, I/O, memory and network intensive jobs. Then, we executed and captured the performance of these jobs on virtual machines and containers. We found that container gives better and stable performance than virtual machine. As per our discussion, containers are more suitable for scientific workflow.

Acknowledgment. This work was supported by the program of the Construction and Operation for Large-scale Science Data Center (K-15-L01-C05) and by National Research Foundation (NRF) of Korea (N-15-NM-IR01).

References

1. Armbrust, M., Armando, F., Rean, G., Joseph, A.D., Randy, K., Andy, K., Gunho, L., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010)
2. Adams, K., Agesen, O.: A comparison of software and hardware techniques for x86 virtualization. ACM Sigplan Not. **41**(11), 2–13 (2006)
3. Uhlig, R., Gil, N., Dion, R., Santoni, A.L., Fernando, M., Anderson, A.V., Bennett, S.M., Alain, K., Leung, F.H., Larry, S.: Intel virtualization technology. Computer **38**(5), 48–56 (2005)
4. Verma, A., Dasgupta, G., Nayak, T.K., De, P., Kothari, R.: Server workload analysis for power minimization using consolidation. In: Proceedings of the Conference on USENIX Annual Technical Conference, p. 28. USENIX Association (2009)
5. http://www.linux-kvm.org/page/Main_Page. Accessed June 2015
6. http://www.vmware.com/. Accessed June 2015
7. http://www.xenproject.org/. Accessed June 2015
8. https://www.openstack.org/. Accessed June 2015
9. https://openvz.org/Main_Page. Accessed June 2015
10. Matthews, J.N., Wenjin, H., Hapuarachchi, M., Deshane, T., Dimatos, D., Hamilton, G., McCabe, M., Owens, J.: Quantifying the performance isolation properties of virtualization systems. In: Proceedings of the 2007 Workshop on Experimental Computer Science, p. 6. ACM (2007)
11. Walters, J.P., Chaudhary, V., Cha, M., Guercio Jr., S., Gallo, S.: A comparison of virtualization technologies for HPC. In: 22nd International Conference on Advanced Information Networking and Applications, AINA 2008. pp. 861–868. IEEE (2008)
12. Deshane, T., Zachary Shepherd, J., Matthews, M.B.-Y., Shah, A., Rao, B.: Quantitative comparison of Xen and KVM. In: Xen Summit, Boston, MA, USA, pp. 1–2 (2008)
13. Vaughan-Nichols, S.J.: New approach to virtualization is a lightweight. Computer **39**(11), 12–14 (2006)
14. Kivity, A., Kamay, Y., Laor, D., Lublin, U., Liguori, A.: kvm: the Linux virtual machine monitor. In: Proceedings of the Linux Symposium, vol. 1, pp. 225–230 (2007)
15. Sefraoui, O., Aissaoui, M., Eleuldj, M.: Openstack: toward an open-source solution for cloud computing. Int. J. Comput. Appl. **55**(3), 38–42 (2012)

16. Soltesz, S., Ptzl, H., Fiuczynski, M.E., Bavier, A., Peterson, L.: Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In: ACM SIGOPS Operating Systems Review, vol. 41, no. 3, pp. 275–287. ACM (2007)
17. Xavier, G., Miguel, M.V., Neves, F., de Rose, C., Augusto.: A performance comparison of container-based virtualization systems for mapreduce clusters. In: 2014 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 299–306. IEEE (2014)
18. http://research.cs.wisc.edu/htcondor/. Accessed June 2015
19. Che, J., Yong, Y., Shi, C., Lin, W.: A synthetical performance evaluation of openvz, xen, kvm. In: Services Computing Conference (APSCC), 2010 IEEE Asia-Pacific, pp. 587–594. IEEE (2010)
20. Regola, N., Ducom, J.-C.: Recommendations for virtualization technologies in high performance computing. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 409–416. IEEE (2010)
21. Tafa, I., Beqiri, E., Paci, H., Kajo, E., Xhuvani, A.: The evaluation of transfer time, cpu consumption and memory utilization in XEN-PV, XEN-HVM, OPENVZ, KVM-FV and KVM-PV hypervisors using ftp and http approaches. In: 2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 502–507. IEEE (2011)
22. Tafa, I., Zanaj, E., Kajo, E., Bejleri, A., Xhuvani, A.: The comparison of virtual machine migration performance between XEN-HVM, XEN-PV, Open-VZ, KVM-FV, KVM-PV. IJCSMS Int. J. Comput. Sci.: Manag. Stud. **11**(2), 65–75 (2011)