OpenStack and Docker Comparison for Scientific Workflow w.r.t. Execution and Energy

Amol Jaikar*†, Sangwook Bae†, Heejune Han†, Byungyun Kong†, Syed Asif Raza Shah*† and Seo-Young Noh*†
*Grid and Supercomputing Department,
Korea University of Science and Technology, Daejeon, South Korea 305-350.
†National Institute of Supercomputing and Networking,
Korea Institute of Science and Technology Information, Daejeon, South Korea 305-804.

***Abstract*—Cloud computing is an emerging technology in the market to manage the data center's infrastructure efficiently. It provides not only more flexibility, but also increases resource utilization. However, virtualization overhead reduces the performance of the job execution. Moreover, virtual machine's specification also affects on job execution. In the scientific world, the experiment's execution time is considerably high. Therefore, it is very important to understand the performance penalties involved in adoption of cloud computing technology for the scientific workflow. In this research, we have tested the scientific workflow on multiple virtual machines and containers with different configuration. We also captured the performance of each virtual machine and container by considering execution time and energy consumption. Our results show that high configured virtual machine gives faster results. Moreover, it also shows that virtualization technology increases energy consumption considerably. About comparison between OpenStack virtual machine and Docker container, Docker container gives faster results for CPU and memory intensive jobs. With these results, scientific community will understand the impact of adoption of cloud computing technology on scientific workflow.**

***Index Terms*—OpenStack, Docker, Scientific Workflow, Execution, Energy**

I. Introduction

Cloud computing technology provides infrastructure, platform and software as a service [1]. This technology enables to get resources dynamically with the pay-as-you-go model. Cloud computing offers more flexibility and increases resource utilization. Moreover, high energy consumption problem of data center attracts the researcher to develop new techniques to reduce the energy consumption [2]. Cloud computing promises high utilization of resources by running multiple virtual machines or containers on a single physical machine. This feature enables to reduce the energy consumption of a data center.

Virtualization is a backbone of cloud computing technology. Virtualization technology simulates hardware to support new virtual machine. There are many software and hardware techniques for x86 virtualization [3]. However, virtualization comes with the performance overhead. It is also difficult to decide which configuration of virtual machine or container is suitable for the specific job. To adopt cloud computing for scientific workflow, it is required to understand the performance penalties involved in it.

Scientific experiment consists of different types of jobs. These jobs can be categorized into CPU, memory and I/O intensive jobs. Currently, grid computing technology is used to execute these jobs directly on the physical machine. In order to adopt the cloud computing technology, we have to understand the execution time and energy consumption of the scientific workflow. Our motivation behind this research is to analyze the performance penalties involved in the adoption of cloud computing for scientific workflow and to decide best virtual machine's or container's specification for the scientific workflow.

In this research, we have not only created the OpenStack and Docker cloud infrastructure, but also tested scientific jobs with the help of HTCondor framework. We have tested these jobs on different configuration of virtual machines. While executing scientific workflow, we also monitored the energy consumption of the physical host. Result shows that high configured virtual machine gives better performance than low configured virtual machine. However, it also shows that virtualization technology increases the energy consumption of the physical host.

The remaining paper is organized as follows. Section II gives motivation for this research. Section III discusses related work. Background knowledge is described in Section IV. Implementation and result analysis are covered in Section V. Finally, Section VI concludes the paper.

II. Motivation

Global Science Data hub Center (GSDC) [4] of Korea Institute of Science and Technology Information (KISTI) is supporting high energy physics (HEP) projects like Compact Muon Solenoid (CMS) [5] and Laser Interferometer Gravitational-Wave Observatory (LIGO) [6] by proving computational resources. GSDC also faces the problem of low utilization of resources. We have statically allocated resources to these projects. One of the projects utilizes the resources around 85% throughout the year. However, utilization of other project is lower than 20%. Due to static allocation of resource, we can not use these resources for another project even though they are idle most of time. Therefore, dynamic allocation of resources will help to increase the resource utilization. Cloud computing technology has capability to provision the resources dynamically. Therefore, we are trying to adopt cloud computing technology for our data center. The motivation behind this paper is to understand the performance penalties involved in transforming from grid to cloud computing.

III. Related Work

This section gives overview of existing work. Omesh Tickoo et al. showed the performance penalties involved in virtualization with the help of standard benchmarks like *Sysbench* and *SPECjbb* [7]. The author also stated the mathematical model for the virtual machine performance.

Morabito Roberto et al. [8] have tested some of the benchmarks to compare between virtual machine and container. The author has mentioned that container performs better than virtual machine. Adufu Theodora et al. [9] also tested the performance difference between OpenStack virtual machine and Docker container with *autodock3* benchmark. The author concluded the paper mentioning that container is more efficient than virtual machine.

Berzano et al. [10] have created the cloud infrastructure to support A Large Ion Collider Experiment (ALICE) HEP experiment. The research article not only explains about virtual machine and container technology, but also shares the experience to manage. The author specified that the average launching time of docker container is much lesser than launching time of virtual machine. The HEP workload is tested on virtual machine and container by Roy Gareth et al. [11]. The author mentioned that container's execution overhead is around 2% while virtual machine's overhead is around 10 to 20%.

Our research is focused to understand the performance penalties and energy consumption for scientific workflow to adopt cloud computing technology. This research will also help not only to determine the virtual machine's configuration, but also to understand the energy consumption and the virtualization overhead for the scientific workflow.

IV. Background

This section gives background knowledge to understand the testing environment details. In this experiment, we have used HTCondor, OpenStack and Docker.

HTCondor is an open source high throughput computing framework, which is developed by *University of Wisconsin-Madison*[12]. It is batch processing software for scientific workflow. It has master-slave architecture where master is responsible to manage the incoming non-interactive jobs and slaves are used to execute these jobs. This framework allows user to submit the job on each core of slave. Therefore, if slave has 16 cores then user can submit 16 jobs simultaneously. This feature increases the throughput of the slave. HTCondor also has waiting job pool, where job resides until they get executed. In our experiment, each virtual machine and container works as a slave of a master.

OpenStack is an open source cloud computing framework to create and manage the cloud infrastructure[13]. OpenStack consists of controller and compute node. OpenStack controller node has responsibility to run identity, image, database, dashboard, message broker, management part of compute and network services. In case of compute node, it runs virtual machine (VM) on hypervisor like KVM. It also runs networking agent for layer 2. In this experiment, we have used two node architecture of OpenStack framework to create and manage the virtual machines.

Docker is also an open source project to create virtual infrastructure i.e. container, based on operating system (OS)-level virtualization [14]. This technology is based on *namespaces*, which provides isolation of resources. These containers share the physical host's Linux Kernel and libraries. This is lightweight layered approach of virtualization.

V. Implementation

This section not only describes the environmental setup to test the scientific workflow, but also elaborates more about test cases and their results.

A. Setup

The environmental setup consists of physical host with 8 core processor and 4 GB of RAM. OpenStack, Docker and HTCondor setup is installed on physical host. Scientific jobs are used to test the performance of OpenStack's virtual machine and Docker's container. To measure the energy consumption while executing scientific workflow, *wattmeter* is used. This physical machine is connected with 1 Gbps network.

We have used *CentOS 7* as an operating system for image file with the installation of HTCondor framework for new virtual machine and container. This newly created virtual machine slave will seamlessly join the HTCondor master to execute jobs. Table I shows the setup configuration where each physical host has one or more virtual machines. *Config. 1* of shows that one physical host has only one virtual machine with the specification of 8 vCPU and 4096 MB RAM. In *Config. 4*, one physical machine has 8 virtual machines, and each virtual machine's configuration consists of 1 vCPU and 512 MB of RAM. These virtual machines will execute the scientific workflow through HTCondor. In case of Docker, container will also serve as a HTCondor slave node.

TABLE I
OpenStack Setup Configuration

Config.	Physical Host	No. of Virtual Machines	Each VM's Specification	
			vCPU	RAM (MB)
1	1	1	8	4096
2	1	2	4	2048
3	1	4	2	1024
4	1	8	1	512

B. Results

1) Scientific Workflow w.r.t. Execution and Energy: In order to calculate the results, we have categorized the scientific workflow into CPU, memory and I/O intensive jobs. CPU intensive job consists of matrix multiplication of continuously changing values of two matrices. This job is executed on each core of the virtual machine to increase the throughput. In case of *Config. 1*, we launched 8 CPU intensive jobs on a single virtual machine and monitored execution time of each job. We tested execution time on different configured virtual machines. Fig. 1 shows the average execution time of CPU intensive jobs on virtual machines. By analyzing Fig. 1, we can conclude

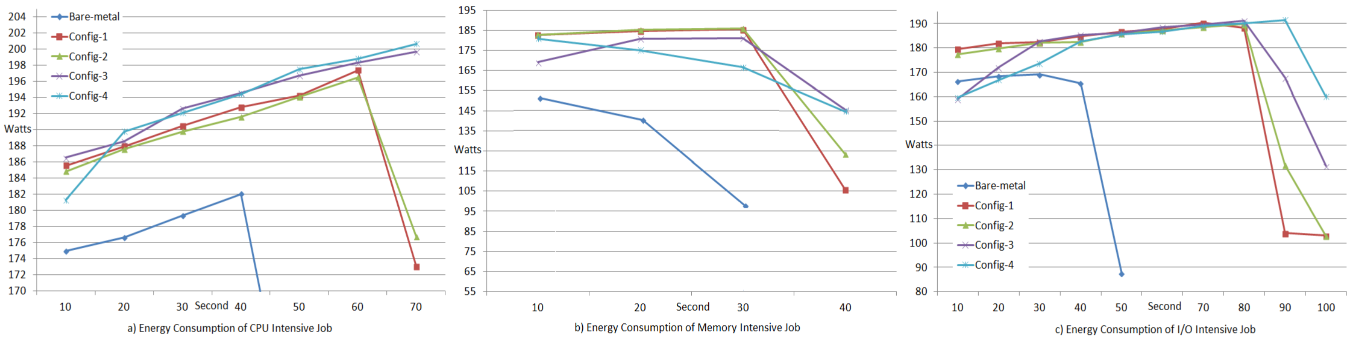

Fig. 2. Energy Consumption

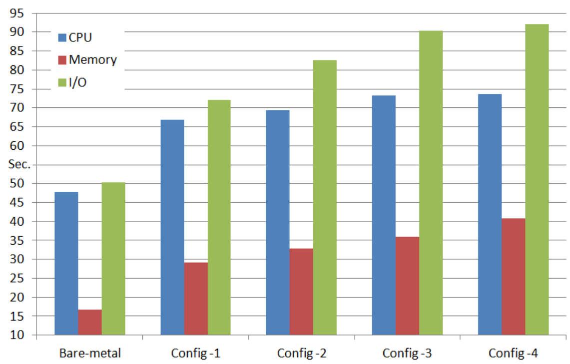

Fig. 1. Scientific Workflow Execution

that the CPU intensive job's execution time is less on high configured virtual machines.

I/O intensive job continuously writes the data into the disk. After writing that data, again it reads data to verify. Therefore, this job consists of many I/O operations. We have tested and these jobs on different configured virtual machines. Fig. 1 shows the result. The result shows that, the performance is better with high configured virtual machines. Fig. 3 shows that, test jobs have used around 100% of computational power. In case of memory intensive job, 91.3% memory is used to execute this job. These jobs are based on local datasets.

The memory intensive job continuously adds the data into a string. This job also runs on each core of the virtual machine. Therefore, a virtual machine with 4 vCPU can run four different jobs concurrently. Fig. 1 shows the average execution time of the memory intensive jobs. A virtual machine with low vCPU has higher execution time than high vCPU configuration virtual machines. Therefore, we can conclude that high vCPU virtual machine gives better performance for scientific workflow.

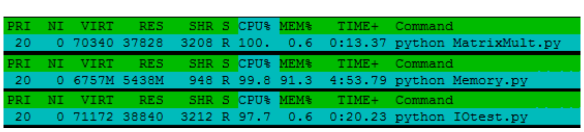

Fig. 3. htop command observation

While testing these jobs, we have monitored the energy consumption of the physical host with the help of *wattmeter*. Fig. 2 shows the results. Bare-metal i.e. physical host consumes less energy while execution of the scientific workflow. However, energy consumption increases while executing same set of jobs on virtual machines. Fig. 2 also shows the energy consumption of physical host with different configuration of virtual machines. Low vCPU configured virtual machine leads not only to high consumption of energy, but also to increase execution time of the scientific workflow.

2) Comparison between OpenStack and Docker: Virtual machine refers to full and para virtualization technology and containers to operating system level virtualization. When comparing virtual machine with container, we have taken OpenStack and Docker framework to test the scientific workflow. In this test case, we have changed the memory intensive job to consume more memory for long period of time. Therefore, it increases the execution time. In this case, we have tested physical host with OpenStack virtual machine and Docker's container with 8 vCPU and 4 GB of RAM. Fig. 4 shows the results. Due to loosely coupled nature of CPU and memory in Docker, CPU and memory intensive jobs are executed faster than virtual machine. However, OpenStack virtual machine gives faster results for I/O intensive jobs than Docker container.

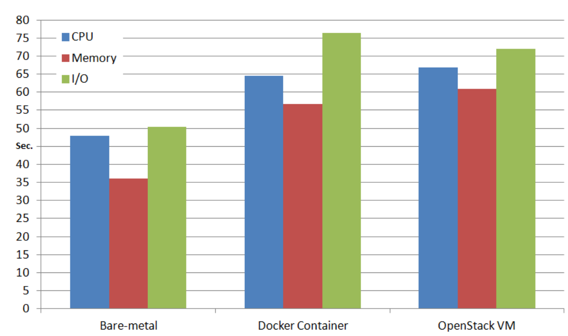

Fig. 4. Execution Comparison with Docker

In terms of percentage, Table II shows the virtualization overhead with respect to physical host. These results are considerable in terms of scientific workflow. If we consider some scientific memory intensive job, which takes 10 hours to produce the results, then we can predict that around 17 hours will require to produce the same set of results by virtual infrastructure. Therefore, it is very important to understand

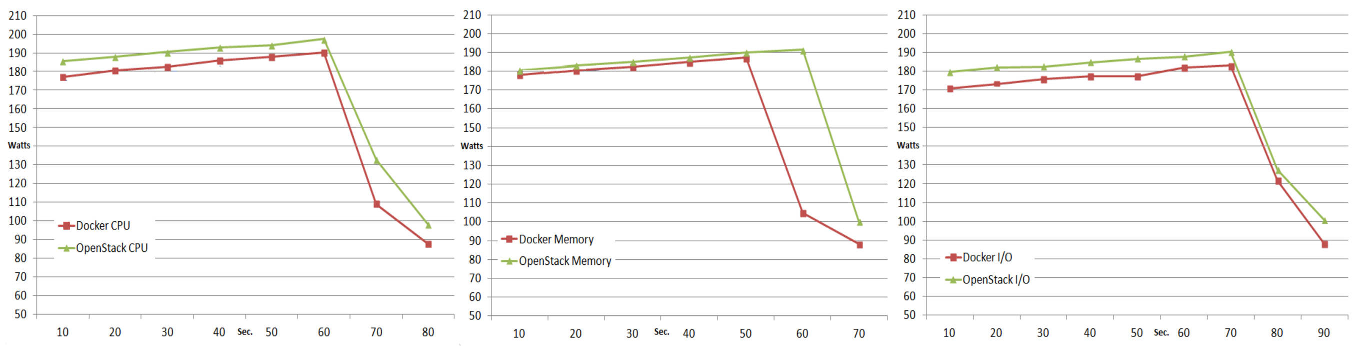

Fig. 5. Energy Consumption of OpenStack and Docker

TABLE II
PERCENTAGE OVERHEAD W.R.T. BARE-METAL

Intensive Jobs	Virtual Infrastructure	
	Virtual Machine	Container
CPU	39.71%	34.78%
Memory	69.05%	57.25%
I/O	43.05%	51.6%

pros and cons of the technology before adoption. Due to these results, we can understand the virtualization overhead with respect to scientific workflow.

TABLE III
IDLE MACHINE POWER CONSUMPTION

Parameter	Each VM's Specification	
	Min. Watts	Max. Watts
Idle Physical Machine	85.3	89.2
Idle Container	86.4	92.6
Idle Virtual Machine	94.7	102.9

While execution these scientific workflow on OpenStack's virtual machine and Docker's container, we have monitored the energy consumption through *wattmeter* after every 10 seconds. Fig. 5 shows the results. These graphs show that Docker container consumes less energy than OpenStack virtual machine while executing CPU, memory and I/O intensives jobs.

In the data center, there is high possibility of running idle virtual resources. While considering this point, we have measured the power consumption of idle virtual machine and container. Table III shows the results. The results show that keeping idle virtual machine is more expensive in terms of power consumption than keeping idle container. Idle container consumes around same power with respect to physical machine.

While working with OpenStack and Docker, we understand the pros and cons of each technology. OpenStack gives more flexibility to manage the resources. For example, Windows virtual machine can also run on Linux physical machine. Migration of virtual machine from one physical machine to another can work without any limitation of operating system or operating system kernel. In case of Docker, it is light weight virtualization technology. Therefore, it takes less time not only to create the container, but also to execute CPU and memory intensive job. Energy consumption of container is also low with respect to virtual machine. However, it has limitation of operating system and operating system kernel. This limitation works for migration too. It means that the user can launch Linux container on Linux physical machine only, due to sharing of physical machine's operating system kernel. Therefore, OpenStack gives more flexibility to manage the hybrid infrastructure which contains many operating system with different setup environment.

VI. CONCLUSION

Cloud computing enables more challenges in the management of data center with the provision of flexibility. Virtualization, which is a backbone of cloud computing, has performance overhead. In order to adopt cloud computing for scientific experiments, researchers need to understand the performance differences between physical machine and virtual infrastructure. Our motivation behind this research is to identify which of the configuration virtual machine is suitable for the CPU, memory and I/O intensive jobs. In this case study, we have analyzed the performance of CPU, memory and I/O intensive jobs on different configuration of virtual machines.

Our results reveal that high configured virtual machine is more suitable for scientific workflow. In terms of energy consumption, low vCPU configured virtual machine consumes more energy than high vCPU configured virtual machine. About comparison between OpenStack virtual machine and Docker container, container outperforms for CPU and memory intensive jobs. Moreover, Docker container consumes less power while executing scientific workflow. But for I/O intensive jobs, virtual machine gives faster result than container. Results also showed that running idle virtual machine is more expensive in terms of power consumption than running idle container. Therefore, these results will help scientific community to understand the impact of cloud computing on scientific workflow.

ACKNOWLEDGMENT

This work was supported by the program of the Construction and Operation for Large-scale Science Data Center (K-16-L01-C06) and by National Research Foundation (NRF) of Korea (N-16-NM-CR01).

References

[1] Armbrust, Michael, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee et al. ”A view of cloud computing.” Communications of the ACM 53, no. 4 (2010): 50-58.

[2] Berl, Andreas, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Quan Dang, and Kostas Pentikousis. ”Energy-efficient cloud computing.” The computer journal 53, no. 7 (2010): 1045-1051.

[3] Adams, Keith, and Ole Agesen. ”A comparison of software and hardware techniques for x86 virtualization.” ACM Sigplan Notices 41, no. 11 (2006): 2-13.

[4] GSDC, http://en.kisti.re.kr/supercomputing/, Nov. 2015.

[5] CMS, http://cms.web.cern.ch/, Nov. 2015.

[6] LIGO, https://www.ligo.caltech.edu/, Nov. 2015.

[7] Tickoo, Omesh, Ravi Iyer, Ramesh Illikkal, and Don Newell. ”Modeling virtual machine performance: challenges and approaches.” ACM SIGMETRICS Performance Evaluation Review 37, no. 3 (2010): 55-60.

[8] Morabito, Roberto, Jimmy Kjallman, and Miika Komu. ”Hypervisors vs. Lightweight Virtualization: a Performance Comparison.” In Cloud Engineering (IC2E), 2015 IEEE International Conference on, pp. 386-393. IEEE, 2015.

[9] Adufu, Theodora, Jieun Choi, and Yoonhee Kim. ”Is container-based technology a winner for high performance scientific applications?.” In Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific, pp. 507-510. IEEE, 2015.

[10] Berzano, D., J. Blomer, P. Buncic, I. Charalampidis, G. Ganis, and R. Meusel. ”Lightweight scheduling of elastic analysis containers in a competitive cloud environment: a Docked Analysis Facility for ALICE.” In Journal of Physics: Conference Series, vol. 664, no. 2, p. 022005. IOP Publishing, 2015.

[11] Roy, Gareth, Andrew Washbrook, David Crooks, Gang Qin, Samuel Cadellin Skipsey, Gordon Stewart, and David Britton. ”Evaluation of containers as a virtualisation alternative for HEP workloads.” In Journal of Physics: Conference Series, vol. 664, no. 2, p. 022034. IOP Publishing, 2015.

[12] HTCondor, https://research.cs.wisc.edu/htcondor/, Nov. 2015.

[13] OpenStack, https://www.openstack.org/, Nov. 2015.

[14] Docker, https://www.docker.com/, Nov. 2015.