Comparison of Virtualization and Containerization Techniques for High-Performance Computing

Yuyu Zhou[*†], Balaji Subramaniam[†], Kate Keahey[†], John Lange[*]
[*]Department of Computer Science
University of Pittsburgh
[†]Mathematics and Computer Science Division
Argonne National Laboratory

1. INTRODUCTION

High Performance Computing (HPC) users have traditionally used dedicated clusters hosted in national laboratories and universities to run their scientific applications. Recently, the use of cloud computing for such scientific applications has become popular, as exemplified by Amazon providing HPC instances. Nonetheless, HPC users have approached cloud computing cautiously due to various reasons. First, the instances in the cloud are virtualized. Such virtualization comes with an associated performance overhead. Second, virtual instances in the cloud are co-located to improve the overall utilization of the cluster and co-location leads to prohibitive performance variability. In spite of these concerns, cloud computing is gaining popularity in HPC due to high availability, lower queue wait times and flexibility to support different types of applications (including legacy application support).

Recent improvements and developments in virtualization and containerization have alleviated some of these concerns regarding performance. There are new container-related technologies such as Linux containers (LXC) and Docker which aim to deliver near bare-metal performance while continuing to provide some of the benefits of a virtualized instance. The applicability of such technologies to HPC applications has not yet been thoroughly explored. Moreover, kernel-based virtual machine (KVM) has several tunable parameters which can be explored to improve its applicability to HPC environments. Furthermore, scalability of scientific applications in the context of virtualized or containerized environments is not well studied.

In this poster, we seek to understand the applicability of virtualization (exemplified by KVM) and containerization (exemplified by Docker) technologies to HPC applications.

2. RELATED WORK

Several studies have been conducted to quantify the performance of virtualization and containerization. However, the performance of HPC applications in such environments has not been thoroughly studied.

As early as 2007, Soltesz et al., compared System V and Xen in [1]. Later, Regola et al., in [2] compared OpenVZ, KVM and EC2 and Felter et al., compared Docker and KVM [3]. They all led to the same conclusion, namely, containerization outperforms virtualization in almost all test scenarios. However, HPC workloads were not the focus in [1] and only single-node experiments were conducted in [2] and [3].

3. QUALITATIVE COMPARISON

From users' perspective, virtual machines (VM) and containers provide a similar illusion of environment with its own file system, network stack, memory, processes, users and group management. But the underlying mechanism is different. For KVM, a separate guest operating system built on emulated virtual hardware is required for each VM. In contrast, the Docker engine sits atop a single host operating system and only provides each container with a specialized Application Programming Interface (API) and Application Binary Interface (ABI) environments.

4. QUANTITATIVE COMPARISON

Our experimental testbed is Chameleon [4], a reconfigurable testbed based on OpenStack bare-metal deployment. We use two compute nodes from the Chameleon platform. Each compute node has two Intel Xeon E5-2670 v3 (Haswell) processors with 128GB of memory.

4.1 Micro-benchmark: lmbench

We use micro-benchmarks to observe the effect on KVM and Docker on low-level system activities such as system calls and memory operations. The lmbench benchmark suite provides a set of tools which stress such low-level system activities. For most of the benchmarks, KVM shows worse performance than Docker.

4.2 Mini-applications from Mantevo

We picked three mini-applications in different application area from the Mantevo benchmark suite:

- CloverLeaf is a hydrodynamics application using a two-dimensional Eulerian formulation.
- CoMD is an extensible molecular dynamics proxy application.
- MiniFE is an unstructured implicit finite difference method.

4.2.1 Scale-Up Experiment

We tested OpenMP implementation of the mini-applications on a single node. KVM has 1.57%, 1.60% and 2.00% and Docker has -0.17%, 0.50% and 0.07% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE.

4.2.2 Scale-Out Experiment

We tested MPI implementation of the mini-applications up to 64 nodes. KVM has 73.77%, 84.85% and 1789.74% and Docker has -1.49%, -0.41% and 17.90% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE.

5. CONCLUSION AND FUTURE WORK

In this poster, we have investigated the applicability of virtualization and containerization to HPC applications. In the future, we plan to investigate the root cause behind the different performance behaviors by measuring low-level performance counters. We also plan to investigate how co-location affects performance.

6. REFERENCES

[1] Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007, March). Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In A*CM SIGOPS Operating Systems Review* (Vol. 41, No. 3, pp. 275-287). ACM.

[2] Regola, N., & Ducom, J. C. (2010, November). Recommendations for virtualization technologies in high performance computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on* (pp. 409-416). IEEE.

[3] Felter, W., Ferreira, A., Rajamony, R., & Rubio, J. (2014). An updated performance comparison of virtual machines and linux containers. *technology*, 28, 32.

[4] The Chameleon Cloud Project. Available at: https://www.chameleoncloud.org/.