Performance Analysis of Applications using Singularity Container on SDSC Comet

Extended Abstract

Emily Le
Department of Mathematics,
University of California, San Diego

e9le@ucsd.edu

David Paz
Department of Electrical and Computer Engineering, University of California, San Diego
dpazruiz@ucsd.edu

Comet's Compute nodes used for the performance analysis of containerized vs non-containerized runs.

ABSTRACT

Multiple nodes of the Comet supercomputer at the San Diego Supercomputer Center were used to analyse and compare (between Singularity and non-Singularity) performance of resource intensive applications of benchmark codes such as NEURON (a computational neuronal simulation tool), OSU Benchmarks, and Intel MPI Benchmarks (IMB). NEURON software was used to simulate a complex neuronal network model (Jones Model from ModelDB hosted at Yale University) and the IMB benchmarks were used to calculate the latency and performance of each method. communication efficiency of MPI. The containerized runtimes were directly compared with the corresponding non-containerized runtime of jobs to analyse the performance of each method. For future work, we plan to explore other technologies such as Shifter and study their performance on HPC.

CCS CONCEPTS

• **High Performance Computing → Containerization** [1]

KEYWORDS

Singularity, IMB: Intel's MPI Benchmark, OSU: Ohio State University Benchmark, NEURON: Neuronal Simulation Tool.

ACM 978-1-4503-5272-7/17/07. 10.1145/3093338.3106737

ACM Reference format:
E. Le, D. Paz, 2017. SIG Proceedings Paper in word Format. In *Proceedings of PEARC17 conference, New Orleans, LA USA, July 2017*, 4 pages.

1 INTRODUCTION

Container technologies have been widely used in recent years due to their applications and benefits in version control, reproducibility, portability and often lightweight properties. Container technologies, such as Docker, provide most--if not all--of the above mentioned benefits. However, some limitations of containerization arise when high performance computing (HPC) is involved. Docker requires the user to be root which is not allowable in most HPC resources due to security concerns. However, Singularity containers inherit the users' permissions and prevent unauthorized deletion or modification of valuable data within the containers--while keeping the container lightweight and HPC supportive. Singularity is available on HPC resources such as the XSEDE Comet HPC cluster.

This study aims to analyze the properties and HPC performance implications of Singularity to determine if container benefits amortize the overhead cost.

2 EXPERIMENTAL AND COMPUTATIONAL DETAILS

2.1 Resources

The Comet supercomputer at the San Diego Supercomputer Center, was utilized to benchmark the different aspects that characterize high performance computing in containerized and non-containerized form. The 1,944 compute nodes of Comet consist of Intel Xeon E5-2680v3 processors, 128 GB DDR4 DRAM (64 GB per socket), and 320 GB of SSD local scratch memory.

MEG of Somatosensory Neocortex (Jones et al. 2007) [1] is a biophysically realistic neuron network model of SI (somatosensory primary neocortex magnetoencephalography) available from ModelDB. The IMB benchmark [2] measures point-to-point and global MPI communication. The OSU benchmark [3] is a set of independent MPI message passing performance microbenchmarks. NEURON is a simulation tool for empirically-based simulations of neurons and networks of neurons [4].

2.2 Data Collection

The data collection phase consisted of three different types of performance tests: Intel MPI's Benchmarks (IMB), Ohio State University's Benchmarks, and the Jones ModelDB Model.

OSU and IMB benchmarks ran with different number of nodes. Each data point represents the average of 10 runs: represented by **Figures 1, 2, and 3**. The data was collectively averaged and analyzed with a Python algorithm.

The Jones Model performance test was performed by a sequence of smaller runs. Each job ran through the compute nodes with varying core count: 1 - 24 cores, 48 cores, and 96 cores. The results are shown in **Figure 4.**

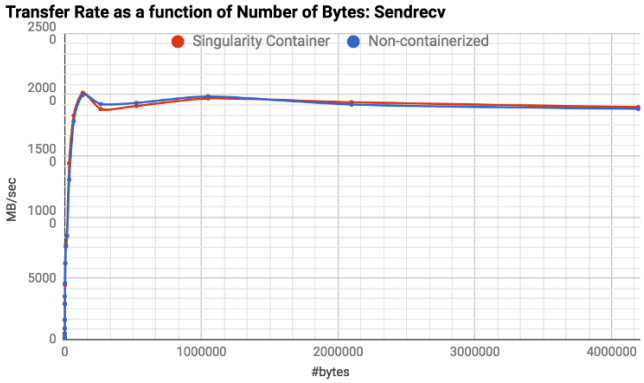

Figure 1: IMB Sendrecv Run using Singularity and Non-Singularity

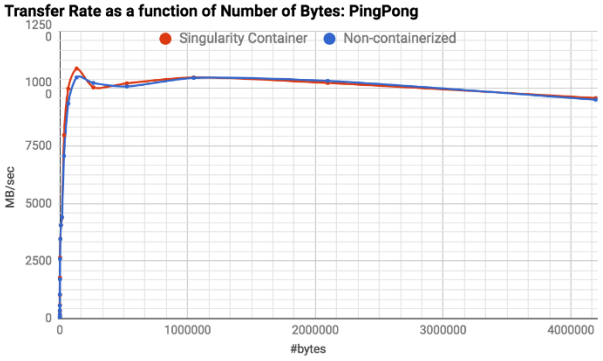

Figure 2: IMB PingPong Run Using Singularity and Non-Singularity

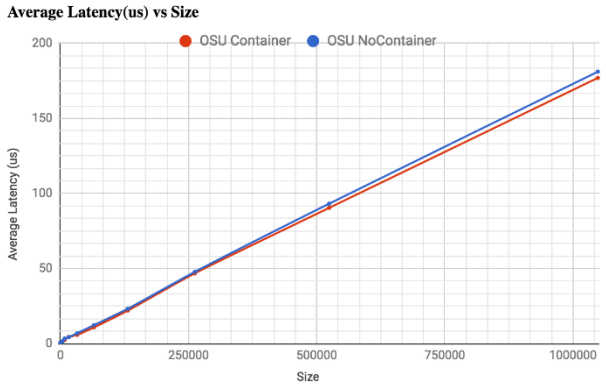

Figure 3: OSU Run using Singularity and Non-Singularity

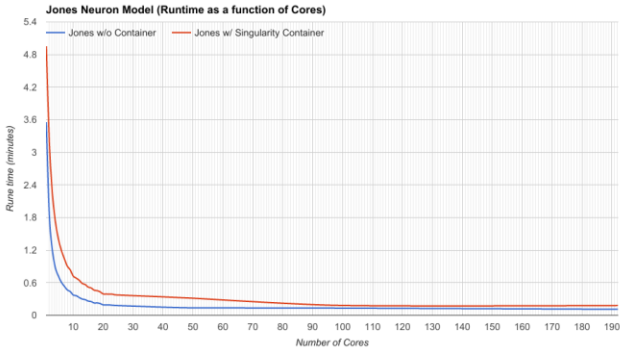

Figure 4: Jones Model NEURON Run Using Singularity and Non-Singularity

2.3 Details

IMB, OSU and NEURON were compiled and installed using Intel's compiler (ICC). Once the source code and executables were compiled and installed, jobs were submitted through Comet's batch system. For NEURON, the code was binded to the container and then jobs were submitted to Comet.

To submit the containerized jobs through the batching system, two separate Singularity containers were bootstrapped. The first container was bootstrapped on an Ubuntu image with NEURON. The second container was bootstrapped with CentOS; IMB and OSU benchmarks were installed in the second image.

The Comet Supercomputer's computing resource, from the San Diego Supercomputer Center at the University of California, San Diego, was utilized for performing these performance tests.

3 RESULTS AND DISCUSSION

3.1 Singularity's Performance

Within the IMB Sendrecv benchmark (**Figure 1**), the data shows the Singularity container running only slightly slower than the non-Singularity process. By analyzing IMB's PingPong benchmark's higher end in **Figure 2** we can see an interleaved pattern between the containerized and non-containerized runs.

Although IMB's Sendrecv and PingPong suggest nearly identical data transfer rate performance, a more noticeable difference can be observed in OSU's Average Latency Benchmark (**Figure 3**). In OSU's Average Latency Benchmark, the initial latency between the containerized and non-containerized job runs is nearly negligible; however, once the input size exceeds 400,000 bytes, the non-containerized begins to deviate at a slightly higher rate. Although these outputs still differ by a small margin, this is more notable difference that can be explained by the container's overhead.

A similar result is produced by the Jones NEURON Model (**Figure 4**). Although this performance test was performed with multiple nodes, we observe more overhead from runs with lower number of cores. As the number of cores and nodes increase, the overhead decreases and remains constant throughout the domain. Between OSU's benchmark and the Jones NEURON Model performance tests, we can see that although we may experience some overhead from the container, as we increase the number of cores in a node, Singularity's performance challenges direct non-containerized jobs. In terms of the transfer rate benchmark testing, the difference is negligible and users could use Singularity containers without an issue.

4 CONCLUSIONS

Singularity features encapsulation, portability, reproducibility and prevention of root escalation privileges. This makes it ideal for use on supercomputers because once a container image is transferred over it does not permit privileged operations with root access. This prevents users from creating and executing malicious software on a supercomputer's shared resources. Singularity's performance results from high performance computing

benchmarks suggest that its negligible performance overhead should not be a significant decision factor when considering its intended applications as the performance is fairly close to non-containerized jobs.

4.1 Future Work

Future work includes performance benchmarking by using Singularity and other HPC containers on different types of virtual clusters. Jetstream's and Comet's HPC virtual clusters could show the performance gains or costs of using virtual clusters over non-containerized and containerized jobs.

REFERENCES

[1] Jones SR, Pritchett DL, Stufflebeam SM, Hamalainen M, Moore CI (2007) Neural correlates of tactile detection: a combined magnetoencephalography and biophysically based computational modeling study. *J Neurosci* ***27***:10751-64

[2] G. Slavova, *Intel MPI Benchmarks*, Intel, 2017.

[3] D. Panda. *OSU*, NERSC, 2017.

[4] M. Hines, *NEURON*, 2017.

ACKNOWLEDGMENTS

This work was partially supported by the NSF grant # 1339856. We would like to thank XSEDE for providing access to HPC resources.

We acknowledge the guidance from the mentors Kenneth Yoshimoto, Subhashini Sivagnanam, Mahidhar Tatineni and Amit Majumdar from the San Diego Supercomputer Center, UCSD.