Containers in HPC: A Scalability and Portability Study in Production Biological Simulations

Oleksandr Rudyy*, Marta Garcia-Gasulla*, Filippo Mantovani*, Alfonso Santiago*, Raül Sirvent*, Mariano Vazquez*†
*Barcelona Supercomputing Center (BSC), Barcelona, Spain
†ELEM Biotech, Spain
E-mail: {oleksandr.rudyy, marta.garcia, filippo.mantovani, alfonso.santiago, raul.sirvent, mariano.vazquez}@bsc.es

Keywords—Container, HPC, Biological Simulation, Scalability, Portability

I. Extended Abstract

Container technology as lightweight virtualization has revolutionized during the last decade the IT business as well as drawn the attention of the High Performance-Computing (HPC) community. After the appearance of Docker [1] in 2013, there have emerged various container implementations aimed for HPC among which stand out Singularity [1] and Shifter [2]. However, the current literature lacks about deep evaluations of containers in HPC [3]. Even though we can find researches measuring and comparing containers performance with bare-metal, it is not enough representative for large HPC centers nor real scientific applications. Thus, we decided to take advantage of our resources conducting a prominent study about container viability in real HPC environments.

In this work we will not only study container performance using Alya [4], a in-production Computational Fluid Dynamics (CFD) code optimized for HPC, up to 256 computational nodes (around 12k cores), but also test container portability on three different state-of-the-art HPC architectures (Intel Skylake, IBM Power9, and Arm-v8) and compare three important container implementations. From the outcomes of all this, we hope to provide to system administrators, facility managers, HPC experts and field scientists a valuable research which to refer for guidelines and use-case examples.

A. Experimental environment

We will be using **Docker**, **Singularity** and **Shifter** container implementations. All they offer operating system virtualization, though, Docker paradigm differs from Singularity's and Shifter's. In addition that Docker relies on a root owned daemon, it also leverages both `cgroups` [2] and `namespaces` [3] capabilities causing full isolation of the container system from the host. On the contrary, Singularity and Shifter manage the SUID (Set owner User ID upon execution) method for running privileged system calls necessary to deploy the container's environment. Besides, they only handle Mount and PID `namespaces`, so their systems possess a more transparent interaction with the host, for instance, for MPI container-host communications.

[1]More details about Docker in: https://www.docker.com/
[2]cgroups: http://man7.org/linux/man-pages/man7/cgroups.7.html
[3]namespaces: http://man7.org/linux/man-pages/man7/namespaces.7.html

About where to test our containers, in this research we leverage four high-end HPC clusters detailed below

Lenox Cluster is a four-nodes cluster, owned by Lenovo, where we have administrative rights. Each node contains a dual-sockets motherboard, housing 2× Intel Xeon E5-2697v3, with 14 cores each (28 cores per node). It has installed Docker 1.11.1, Singularity 2.4.5 and Shifter 16.08.3. Compute nodes are interconnected via 1GbE network over TCP.

MareNostrum4 is a Tier-0 supercomputer in production at Barcelona Supercomputing Center (BSC) in Barcelona, Spain. Its nodes are based on Intel Xeon Platinum 8160 CPUs with 48 cores per node, with a total number of 3456 nodes available. The interconnection network is 100 Gbit/s Intel Omni-Path. Singularity 2.4.2 is deployed as container technology.

CTE-POWER is also hosted at Barcelona Supercomputing Center. This cluster is based on IBM Power9 8335-GTG processors of 20 cores where each compute node contains two CPUs providing 40 cores per node, with a total number of 52 nodes available. Nodes are interconnected via an Infiniband Mellanox EDR network. It has available Singularity 2.5.1.

ThunderX mini-cluster belongs to Mont-Blanc project [5]. The cluster houses four compute nodes, each containing 96 Armv8-a cores organized in two CN8890 sockets with 48 cores each. Nodes are interconnected using 40 GbE network over TCP. It runs Singularity 2.5.2.

B. Evaluation

1) **Containerization Solutions:** Where we compare the performance of the three container solutions under evaluation (Docker, Singularity and Shifter) regarding deployment overhead, image size and execution time.
2) **Portability:** Where we discuss container portability, executing the same containerized application with Singularity in three different architectures and using two techniques to build the container images.
3) **Scalability:** Where we compare the scalability of the Alya use-case comparing performance obtained running it at scale on MareNostrum4 in a Singularity container versus a bare-metal execution.

For our evaluations we employ two biological use cases of Alya:

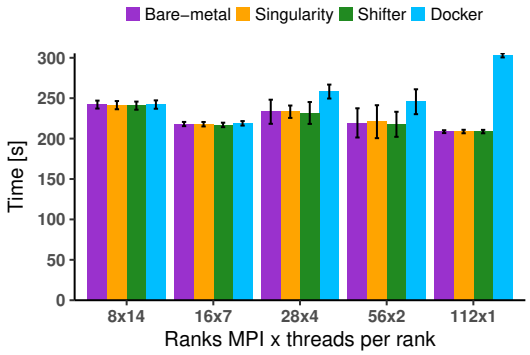

Fig. 1: Average elapsed time of the artery CFD case in Lenox

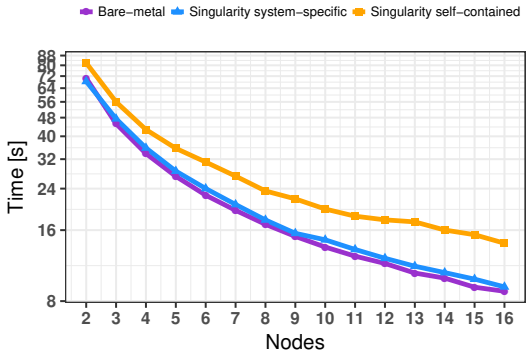

Fig. 2: Average elapsed time of artery CFD case in CTE-POWER

- **CFD:** The simulation of the fluid (blood) through the artery, which is a single code solving the Navier-Stokes equations for fluid dynamics.
- **FSI:** a fluid-structure artery simulation that requires two instances of different codes: the first code studying the fluid sub-domain and the second one simulating the solid sub-domain.

C. Results

In Fig. 1 the average time step duration for each version (Bare-metal, Docker, Singularity and Shifter) is shown. In the x-axis is displayed different configurations of MPI processes and OpenMP threads per rank. We can observe that HPC designed containers (i.e., Shifter and Singularity) can reach close to bare-metal performances whereas Docker degrades soon as we scale in MPI. During the portability test, we have realized that containers can be integrated with the host to leverage its specific features, for example the fast MPI interconnection. So, Fig. 2 shows the average time of three CFD artery versions. The integrated container (Singularity system-specific) can equal bare-metal performance, the opposite of the self-contained container, which is unable to use the Mellanox EDR network. Finally, Fig. 3 presents our scalability test of Alya's FSI use case in MareNostrum4 using up to 256 nodes (12.288 cores). As before, the integrated container can leverage the Intel Omni-Path network, unlike the self-contained which at 32 nodes stops scaling.

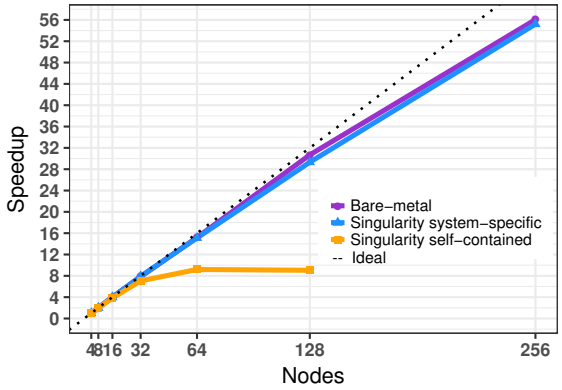

Fig. 3: Scalability plot of Alya artery FSI case in MareNostrum4

D. Conclusions and Future Work

In this paper we extensively investigated the deployment and performance of three container technologies under a production HPC workload. In summary, we found that containers are able to obtain bare-metal performances and can be tuned to leverage host-specific features in exchange of portability. Our study lacks a deeper evaluation of I/O and distributed storage performance using containers, which could be interesting future work.

II. Acknowledgment

This work has been accepted and is pending of presentation in the International Parallel and Distributed Processing Symposium (IPDPS), 2019.

References

[1] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PloS one*, vol. 12, no. 5, p. e0177459, 2017.

[2] D. M. Jacobsen, "Contain this , unleashing docker for hpc," 2015.

[3] N. G. Bachiega, P. S. L. Souza, S. M. Bruschi, and S. d. R. S. de Souza, "Container-based performance evaluation: A survey and challenges," in *2018 IEEE International Conference on Cloud Engineering (IC2E)*, April 2018, pp. 398–403.

[4] E. Casoni, A. Jérusalem, C. Samaniego, B. Eguzkitza, P. Lafortune, D. D. Tjahjanto, X. Sáez, G. Houzeaux, and M. Vázquez, "Alya: computational solid mechanics for supercomputers," *Archives of Computational Methods in Engineering*, vol. 22, no. 4, pp. 557–576, 2015.

[5] N. Rajovic, A. Rico, F. Mantovani *et al.*, "The Mont-Blanc prototype: An alternative approach for HPC systems," in *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov 2016, pp. 444–455.

Oleksandr Rudyy was born in Ivano-Frankivsk, Ukraine, in 1997. Currently he is finishing his Bachelor's Degree in Computer Engineering at UPC-FIB. Since 2018 he has been working in Barcelona Supercomputing Center as a junior research engineer, where he evaluates application container technologies within the Joint Research Activities of HPC-Europa3 project.