Characterizing and Improving Containerized HPC Applications Performance on Fujitsu A64FX Architecture

Principal Investigator : Amit Ruhela
Texas Advanced Computing Center Austin, 10100 Burnet Rd
University of Texas at Austin, Texas, 78758,
Phone No : +1 (614) 209-6159

Project Users : Amit Ruhela, Stephen Lien Harrell, Richard Todd Evans
E-mails : {aruhela, sharrell, rtevans}@tacc.utexas.edu

May 3, 2021

Abstract

Containerization technologies provide a mechanism to encapsulate applications and many of their dependencies, facilitating software portability and reproducibility on HPC systems. However, in order to access many of the architectural features that enable HPC system performance, compatibility between specific components of the container and host is required, resulting in a trade-off between portability and performance. Containerization has become extensively popular in recent years, mainly on cloud infrastructures. To leverage the tremendous computing capabilities along with highly optimized message communication on HPC systems, developers and end-users must learn how to quantify and minimize the performance overheads of containerized applications on HPC systems. They need to understand writing application containers that are not only performant but also portable to run on diverse hardware platforms effortlessly. Further, portability issues mainly due to compilers, libraries, and architectures should be learned and fixed by novel designs and avoided through best practices and guidelines. This proposal seeks $11.6K Node hours as service allocation units for study, analysis, and optimizations for containerization studies on ARM A64FX architecture. The allocation would help advance the use of containerized applications to significantly reduce the applications configuration time on Fujitsu architectures.

1 Research Objectives and Proposal

Containerization is a powerful tool for scientific software development and portability across systems. It considerably reduces the time to build, test, and deploy applications by encapsulating code and dependencies together, allowing them to run on diverse platforms with minimal additional efforts. HPC infrastructures provide tremendous computing capabilities along with optimized message communication actualized through advanced features like eager communication, shared memory, and Remote Direct Memory Access making them ideal for intensive scientific computation but challenging for software portability. Containers provide a promising way to hide system-level complexities, allowing researchers to focus on productive studies that include COVID-19 research, climate modeling, agriculture, healthcare, smart cities, e-commerce, deep learning, etc. With Docker's introduction in 2013, containerization gained tremendous popularity. Since then, several containerization techniques have been developed primarily based on chroot, control groups, and Linux namespace features. Docker is a user-friendly industry-standard containerization approach designed to support stateful microservices. This stateful approach creates security concerns on HPC systems due to its need for root privileges. The security issues combined with a lack of MPI support and resulting scaling limitations make Docker unfit for an HPC environment. Singularity, Charliecloud, Udocker, Podman take different approaches and are designed for HPC users. Once installed with root privileges, Singularity and Charliecloud users can run respective containers without elevated permissions. Several studies in the past have focused on the performance characterization of containerized workloads. These studies, conducted at small problem sizes, indicate near-native performance by container-based techniques. However, none of the prior studies have comprehensively shown the performance, usability, and portability of state-of-the-art container approaches at medium and large scale on diverse CPU as well as GPU architectures. This motivates us to study the following questions.

1. Does the performance of container-based solutions on HPC clusters match bare-metal runs at varying large problem scales?
2. What are the challenges and possible directions to exploit the state-of-the-art container techniques at a massive scale?

With the above broader aims for containerization, we plan to conduct experiments with State-of-the-art containerization techniques and evaluate the overheads and the performance of containerization. The complete set of objectives of this proposal are listed below:

1. Understand the challenges, issues and baseline performance for containerization approaches on Fujitsu A64FX architecture.
2. Evaluate at a high level the performance and overheads in running containers on the various problem and system scales.
3. Develop approaches to quantify containerization overheads at the fine-grain level through profiling and debugging techniques.
4. Improve the performance and portability of Containerized applications by novel designs and optimizations on A64FX architecture.

We have conducted large scale experiments on Petascale clusters like Purdue Bell, TACC Frontera, and TACC Longhorn to understand and analyze the performance of Intel, AMD, and NVIDIA architectures and want to include the ARM platform. The A64FX processor developed by Fujitsu based on ARM architecture is designed with the latest innovations e.g. Scalable Vector Extension (SVE), great performance vs. power ratio, and multiple precision options, makes it an excellent fit for HPC/AI applications. We work towards the above-said goals to conduct a holistic study and analysis of Containerization performance and overheads across the state-of-the-art processor architectures. In total, we request 11,655 node hours for this research project, whose details are given in the following section.

2 Computational Research Plan

USAGE: Testbed
Disk space (home, project, scratch): 100-200 GigaByte
Personnel Resources: Not Required
Required Software : C/C++/Fortan Compilers, MPI, Containerization Softwares (e.g. Singularity, CharlieCloud, Udocker) : All of the required containerization software can be installed in the userspace with minimal support required from System Administrators.

We plan to run microbenchmarks and applications to profile and evaluate the performance of containerization approaches. We will run 30 iterations of MPI and IOR benchmarks (and throughput benchmark) at 1, 2, 4, 8, 16, 32, and 64 nodes at benchmarks level. MILC - a lattice quantum chromodynamics code and VPIC - a 3d electromagnetic relativistic Vector Particle-In-Cell code for modeling kinetic plasmas application are chosen to evaluate the performance of the scientific application at the scale of 1,4,8,16,32 and 64 nodes scale. The containerization techniques have worked flawlessly on Intel and Power9 architectures and therefore should work quite well on ARM architectures as promised by corresponding documentations and the earlier experiments. More details are available in the Table 1 given below.

3 Expected Impact

The proposal will benefit both the scientific, HPC and Big Data communities. We plan to take a comprehensive look at the performance of containerized runtimes in the context of their fine-grained

Table 1: Computational Plan

Task	Benchmark/Application	Node Hours
Microbenchmarks	IOR and Throughput Benchmarks	2,230
Applications	MILC	2,040
Applications	VPIC	2,040
Profiling and Debugging	Benchmarks and Applications	2,000
Total		11,655

analysis of startup and teardown overheads and improvements in the performance of scientific applications. For the NSF community, the benefits are that these codes will take less system time for their allocations. In addition, as a result of working with several different codes, we will develop best practices and example implementations of containerized techniques. These best practices and examples will be disseminated to the greater HPC community via research papers and presentations at HPC-focused conferences and meetings.

4 Research Team:

Amit Ruhela works is a Research Associate in the HPC group at TACC, Austin. He has a strong expertise in MPI communication libraries, HCA interconnects, and containerization approaches. Stephen Lien Harrell is an Engineering Scientist in the HPC Performance and Architectures Group at the Texas Advanced Computing Center. He is primarily focused on benchmarking tools and infrastructure. Richard Todd Evans is a manager in High Performance Computing group and has an extensive experience in high-energy physics, HPC systems, and application performance monitoring and analysis.

5 Research Grant:

All three project users are currently working on BigHPC project with goals to explore, develop and integrate new monitoring, visualization and storage management components capable of dealing with the scale and heterogeneity of HPC infrastructures and applications and enable efficient, scalable, and reliable collection of performance and resource consumption metrics from large-scale HPC infrastructures. This work is supported by *University of Texas at UT Austin - Portugal* Program, a collaboration between the Portuguese Foundation of Science and Technology and the University of Texas at Austin, award UTA18-001217.

6 Talks and Publications:

1. Amit Ruhela, Matt Vaughn, Stephen Lien Harrell, Gregory J. Zynda, John Fonner, Richard Todd Evans, Tommy Minyard, ***Containerization on Petascale HPC Clusters***. In State of the Practice Talks of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'20), 2020

2. Amit Ruhela, Stephen Lien Harrell, Richard Todd Evans, Gregory J. Zynda, John Fonner, Matt Vaughn, Tommy Minyard, John Cazes, ***Characterizing Containerized HPC Applications Performance at Petascale on CPU and GPU Architectures*** In ISC High Performance 2021. (ISC 2021)