Containerization and Virtualization Summary Report

Richard Carlson, Vipin Chaudhary, and Dhruva Chakravorty
MAGIC Team

March 2020[1]

Overview

The Federal Networking and Information Technology Research and Development Program's Middleware and Grid Interagency Coordination Team held a speaker series on containerization and virtualization technologies in Spring 2018. Federal, industry, and academic stakeholders came together to discuss the role of these technologies in supporting scientific workflows involving high-performance computing, high-throughput computing, and cloud computing. The series provided insights into why different fields of science adopted these technologies, how they accomplished the transition, and the challenges they faced. Participants identified the state of the art and discussed the path forward. Scalability, both in terms of code performance and adoption of large-scale computing, was a core focus as the speakers analyzed these innovative frameworks to support multidomain science.

Introduction[2]

Over the last decade, the tremendous growth in data-enabled science and engineering, coupled with the push for exascale computing, are leading researchers to adopt new large-scale computing practices. These include traditional high-performance computing (HPC) and high-throughput computing (HTC) as well as emergent commercial and public sector cloud computing services. Even though computational research communities are migrating toward these technologies at various times and differing rates, the result is the same, widespread adoption of innovative and individualized approaches and platforms. This move away from analytics and tools being developed on standardized platforms with standard libraries is a challenge for HPC and HTC facilities where improved networks have made the live analysis of streaming data possible, but the need for compatibility between platforms has increased.

Coinciding with these research trends is a growing demand for accuracy, reproducibility, fairness, ethics, transparency, and accountability. Led by research journal publishers, researchers are asked to meet rigorous standards, including submission to the publisher of all data required for reproducibility of their results. Taken together, these factors are driving the development of containerization and virtualization

[1] This document is a work of the United States Government and is in the public domain (see 17 U.S.C. §105). It may be freely distributed and copied with acknowledgment to the NITRD Program. This and other NITRD documents are available online at https://www.nitrd.gov/publications/index.aspx. Published in the United States of America, 2020.

[2] Any mention of commercial products within this document is for information purposes only; it does not imply recommendation or endorsement by the NITRD Program.

technologies that assist researchers in managing and maintaining both their computing and data resources.

The Middleware and Grid Interagency Coordination (MAGIC) Team of the Federal Networking and Information Technology Research and Development (NITRD) Program held a speaker series on containerization and virtualization technologies in the Spring of 2018. Federal, industry, and academic stakeholders came together to discuss and analyze the role of these innovative technologies in HPC, HTC, and cloud computing workflows that support multidomain science. The series and the follow-up MAGIC meeting in May provided insight into why different fields of science are adopting these technologies, how they are accomplishing the transition, and the challenges they face. As participants identified the state of the art and discussed the path forward, scalability emerged as a focus for both code performance and the adoption of large-scale computing.

Discussion Summary

Context

Participants analyzed the importance of innovative containerization and virtualization computing frameworks in the following contexts:

1. *Definition of Container.* A container is an operating system within an operating system. Unlike a Virtual Machine (VM), it does not virtualize the processor. As such, there is no overhead for translating machine instructions. A container shares the kernel of the host operating system while spoofing file system and network access via system calls. In contrast to VMs, containers are not viewed as long-running processes.
2. *HPC and HTC Sites.* These sites support modeling and simulation, for both new and experienced users, in a variety of disciplines and experiments. A common concern is that researchers need different resources at different times and increasingly require support for data-intensive use cases. Containers can easily run data-intensive analysis or visualization wherever users have access; however, containers are not necessarily amenable to the traditional modeling and simulation model.
3. *Characteristics and Functions of Containers.* Containers give users greater control over the environments in which they execute their applications. In the context of large-scale, data-intensive workflows, the goal at most sites is to assist users in deploying the container regardless of the site's underlying technology. Containers shield users from the underlying complexity of systems. They allow users to ensure reproducibility on other machines, and to install arbitrary software as non-root users (root users have access to all commands and files on an operating system). Communities can run their own code on HPC systems, connect to the systems via gateways, and often receive performance benefits. However, security concerns increase as these infrastructures become available to more users.

The speaker series was organized into three sessions that provided expert overviews of container services in general, in research and education (R&E) workflows, and in large-scale computing platforms.

Container Services

Containers and virtualization technologies have their home in the cloud computing communities where there is an operational need to provide users with root-like privileges while reducing the overhead on the physical hardware. Since commercial clouds provide micro-services to the user and are not designed to support HPC and HTC types of research workloads, the first containers were not developed with the needs of researchers in mind. Today, containers provide users with the benefits of VMs, without the overheads and cloud, and with tight integration to HPC and HTC facilities. By deploying containers, users can focus on developing an application instead of managing their infrastructure. Free scalability, fault tolerance (redeployment capability), and flexibility are core attributes which drive the rapid adoption of containers in the R&E community.

Container Services in Research and Education Workflows

HPC and HTC facilities are successfully deploying containerization technologies for users in a variety of scientific research fields. For example, in a fundamentally heterogeneous environment, the Open Science Grid (OSG)[3] creates the appearance of a homogenous environment for users by supporting a variety of container technologies. Many Federally supported research facilities have successfully deployed these techniques including:

- Extreme Science and Engineering Discovery Environment's Comet[4]
- Laser Interferometer Gravitational-Wave Observatory (LIGO)[5]
- National Energy Research Scientific Computing Center (NERSC)[6]
- Fermilab's High Energy Physics Portal (HEPCloud)[7]
- Advanced Light Source (ALS)[8]
- Earth System Grid Federation (ESGF)[9]

Container Services in Large-Scale Computing Platforms

While Docker remains the standard bearer in cloud computing, technologies such as Shifter, Spin and Singularity have emerged to support scientific research workloads. Kubernetes continues to remain the leader in orchestration. Each technology is summarized below.

Docker

Docker makes possible the widespread adoption of containers in research computing scenarios. It popularized the idea of containers as a means for distributing software, performing resource management, and providing applications for private, customized environments. Dockerfile produces an image, creating a container. That container can be moved around, provisioned at new sites, and executed on different platforms. This makes the building, shipping, and running of applications portable. However, for HPC and HTC environments, there are security and storage-based challenges. For example, with its "all or nothing" security model, once a user runs a Docker container, they are given full system privileges; in shared environments where a batch system manages nodes and a file system manages storage, Docker can be problematic.

Singularity

Singularity is a container technology that is designed specifically for scientific computing. Developed at DOE's Lawrence Berkeley National Laboratory, it allows users to have full control of their environment,

[3] https://opensciencegrid.org/

[4] https://portal.xsede.org/sdsc-comet

[5] https://www.ligo.caltech.edu/

[6] https://www.nersc.gov/

[7] https://computing.fnal.gov/hep-cloud/

[8] https://als.lbl.gov/

[9] https://esgf.llnl.gov/

and to use containers to package scientific workflows, software, libraries, and even data. It is compatible with both Docker images and complicated HPC architectures. Its images can be archived and managed as data, and its security model is designed to support untrusted users from untrusted containers. Singularity's image format allows for global header and object descriptors to reference raw data objects within the file, enables cryptographic signatures for data objects, and adds a signature block as a data object. This allows for changes to be monitored. A user can also ensure software reproducibility by copying a container, a feature which is ideal for journal publications.

Shifter

Shifter is a container technology developed at NERSC. It leverages the Docker image ecosystem while allowing users to securely run containers in an HPC environment with access to a shared file system and highspeed network. Shifter can be integrated with a workload manager as well.

Spin

Spin was also developed at NERSC. It is a container technology that provides a flexible system to build and deploy science gateways, workflow managers, and "edge services" quickly using Docker containers. Everything is built from containers that are software-defined; resources can be spun up quickly and are easy to manage. Spin also offers opportunities to deploy entire stacks in batch job prologue via application programing interface (API) calls.

Kubernetes

Kubernetes is an open-source platform designed to automate and orchestrate the deployment, scaling, and operation of application containers. It currently operates as a managed service on Amazon, Google, and Azure cloud platforms. It is suitable for projects that need to control their own applications and data and performs well for certain workloads.

OpenStack

OpenStack provides infrastructure as an abstraction. OpenStack Ironic provisions bare metal machines and is used in the Chameleon Cloud. Container-hosted OpenStack services, such as OpenStack Ansible, are infrastructure containers that have privileged system access and provide infrastructure support and management.

AWS

AWS is a commercial cloud provider and uses containers at scale via Elastic Container 2 (EC2) instances. AWS envisions a future in which EC2 returns to tasks and essentially deconstructs containers.

Conclusion

There is a growing need to support diverse compute requirements in scientific research computing environments. Containers and virtualization technologies assist researchers in meeting computational needs, and there has been significant progress in developing the underlying technology and understanding the critical operational requirements. These technologies will continue to be developed with open-source software, security, and easy integration into current large-scale computing sites. Usability is driving demand. The growth of these tools in the academic, government, and private sectors shows tremendous promise. Improved cross-sector collaboration and cooperation, as well as engagement with standards bodies, will help bridge the existing differences between science and commercial use. Ideas for further discussion include:

- Management and operations support that includes provisioning, integration, balancing, open APIs, authentication, monitoring, self-healing, resilience, reliability, availability, and data analytics.
- Policies for interoperability, trust, verification, and authentication that do not impact usability.
- Forums for collaboration, progress assessment, tool sharing, test suite development, workforce development, and training.
- Innovative tools for multi-tenancy and real-time access controls to enforce security policies.

About the Authors

The NITRD Program is the Nation's primary source of federally funded work on pioneering information technologies (IT) in computing, networking, and software. The NITRD Subcommittee of the National Science and Technology Council's Committee on Science and Technology Enterprise guides the multiagency NITRD Program in its work to provide the R&D foundations for ensuring continued U.S. technological leadership and meeting the needs of the Nation for advanced IT. The National Coordination Office (NCO) supports the NITRD Subcommittee and the Interagency Working Groups (IWGs) and Teams that report to it. The NITRD Subcommittee's Co-Chairs are Kamie Roberts, NCO Director, and Margaret Martonosi, Assistant Director of the NSF Directorate for Computer and Information Science and Engineering. More information about NITRD is available online at http://www.nitrd.gov.

The MAGIC Team was established in 2002 and provides for information sharing among Federal agencies and non-Federal participants with interests and responsibility for middleware, grid, and cloud projects; individuals involved in middleware, grid, and cloud research and infrastructure; individuals involved in implementing or operating grids and clouds; and users of grids, clouds, and middleware. The MAGIC Team reports to NITRD's Large Scale Networking (LSN) IWG. More information is available online at https://www.nitrd.gov/groups/magic/.

Acknowledgments

The National Coordination Office for the NITRD Program gratefully acknowledges MAGIC Co-chairs Richard Carlson (DOE) and Vipin Chaudhary (NSF); Dhruva Chakravorty (Texas A&M University); guest presenters; and the members of the MAGIC Team who helped plan and implement the Containerization and Virtualization Technologies Speaker Series and write and review this report