Shifter experiences with high performance containers

Kean Mariotti (kean.mariotti@cscs.ch), Lucas Benedicic, Alberto Madonna, Felipe A. Cruz. CSCS, Swiss National Supercomputing Centre.

Abstract
High Performance Computing (HPC) applications have to reliably run across many platforms and environments. Containers are a type of lightweight virtualization technology that attempts to solve this problem by packaging applications and their runtime environments into standard units of software that are: portable, easy to build and deploy, have a small footprint and a low runtime overhead. In this poster, we present a container workflow for HPC applications using two tools: Docker, for building and packaging applications into containers; and Shifter, a container runtime that has been built to fulfill the specific needs of HPC. We will present use cases of building and testing containers on workstations and deploying them on HPC systems, where they take advantage of the available specialized-hardware: fast interconnects and GPU accelerators. We will provide performance results for a variety of scientific applications and discuss how such workflow can aid HPC users.

1. Building and maintaining HPC applications is hard
- Applications are built directly on HPC system
- Dependencies are often built from source
- Slow modify-build-test cycle
- The application is not portable
- Build process often not reproducible

2. Virtualization containers to the rescue
- Docker: mature, widely used container ecosystem
- Docker Hub provides many officially supported images out of the box (TensorFlow, Trilinos, CUDA Toolkit, etc.)

3. Workflow
2. Push to Docker Hub
3. Pull into storage at HPC center
4. Run at scale on HPC system
1. Develop Docker image

4. Shifter [1][2]
- Container engine for HPC
- Designed for performance
- Integration with workload manager
- Compatible with Docker

5. Shifter + Docker
- Applications are built on the workstation
- Dependencies are installed from package manager
- Rapid modify-build-test cycle
- The container is portable
- Enforced reproducibility of build process
- Same performance as native application

6. Examples of applications

- High Performance Linpack
- N-Body (GPU)
- TensorFlow (GPU)

7. Specialized Hardware support
- Container accesses GPU
- Container accesses native MPI and high-speed interconnect
- Same performance as native application [3]

8. Conclusions
- Docker: faster build, portable and reproducible
- Shifter: run containers at scale with native performance
- Shifter + Docker: simplified workflow without performance penalties

References