LA-UR-24-28252

Author(s): Angelica A. Loshak, London Bielicke

Mentors: Reid R. Priedhorsky, Lucas Caudill

Title: Charliecloud as a Kubernetes Runtime

Abstract:

Kubernetes is a popular system for automating container deployment and management of applications with components that run on various machines with different environments. HPC users require new systems like Kubernetes to support the increasing demand for novel workflows, especially in AI. Many HPC users use Slurm to manage and provision containerized jobs. However, Kubernetes could either replace or supplement Slurm in complex systems. For example, users can schedule containerized jobs, scale containers, and maintain metrics on container health using Kubernetes' declarative approach. Furthermore, Kubernetes decouples the container runtime from container management by supporting various container runtimes, meaning that users can customize their workloads using container systems that meet their needs.

HPC workloads could also benefit from Charliecloud, a lightweight, fully unprivileged container service. However, Kubernetes only supports container services that implement the Container Runtime Interface (CRI), which Charliecloud does not yet support. We developed a prototype server program with essential CRI methods that map to Charliecloud operations, so that Kubernetes can run containers using Charliecloud. Kubernetes can now communicate with Charliecloud over a server to: 1. Create and start containers, 2. Track container health metrics, 3. Run pods, a kubernetes component that creates another layer of isolation around containers, and 4.

Respawn containers within a pod. Kubernetes expects certain features that Charliecloud does not use, such as network namespaces. However, Charliecloud can communicate with Kubernetes without certain expected operations. For example, the method PortForward() is a no-op, since Charliecloud containers share the host IP address. Furthermore, Kubernetes runs containers using separate operations: create and start, while Charliecloud uses a single operation. Mapping Charliecloud operations to CRI methods ensures that Charliecloud and Kubernetes make compatible assumptions.

By implementing the CRI as a server in Charliecloud with 700 lines of code, modifying less than 50 lines of other Charliecloud source code, and making no changes to Kubernetes, we demonstrate that Kubernetes and Charliecloud are compatible tools. Integrating these technologies facilitates scientific advancements which require large compute power and workflows novel to HPC.