

Charliecloud's Successful Prototype Integration with Slurm: A Promising Approach with Some Strings Attached

Abstract

Layton McCafferty, Nicholas Volpe, Hank Wikle
Mentors: Lucas Caudill and Reid Priedhorsky

LA-UR-23-28984

Containerization is becoming increasingly important in HPC. Container technologies leverage Linux kernel isolation mechanisms to promote software package flexibility, application portability, and customization of user software stacks. Charliecloud brings these benefits to HPC, providing a lightweight, fully unprivileged container runtime.

Slurm, another key tool in HPC, is a workload management tool responsible for scheduling the allocation of resources and jobs across multiple interconnected nodes. In 2021, Slurm 21.08 added support for container workflows via the `--container` flag, which provides an interface for interacting with standardized container bundles via any container runtime compliant with the Open Container Initiative (OCI) standard.

Since ease of use can make or break adoption of software tools, it is important that the Charliecloud runtime integrates smoothly with Slurm. Issues related to ease of use can pose barriers to adoption of Charliecloud, which in turn prevents users from reaping the benefits that it provides.

In collaboration with developers at SchedMD, our team successfully prototyped an approach to integrating Charliecloud with Slurm's `--container` flag and underlying features. While this new approach is a viable solution to the original problem, the somewhat convoluted and intricate configuration process places a potential burden on both system administrators and users. Specifically, this new approach imposes limitations in two key ways: (1) the approach may require an upgrade to Slurm 23.02, a process which at present is error-prone and may not always be possible; and (2) the approach requires options and arguments passed to Charliecloud's runtime to be hard-coded in a configuration file. We believe that this prototypical approach shows promise and that future work may be able to eliminate these limitations.