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

Motivation
(1) Charliecloud differs from other runtimes by being lightweight and fully unprivileged.
(2) Integrating Charliecloud with Slurm’s container feature allows users to provide their jobs with a customized software stack.

Building and Testing Charliecloud
Charliecloud is lightweight; the dependencies are minimal. There is one notoriously tricky one: libsquashfuse.

Two problems users might run into when testing:
(1) Charliecloud storage directory was corrupted when canceling the test via <CTRL+C>
   ○ Bug report is live and in the meantime can be fixed via clearing directory.
(2) SELinux needs to be disabled for Charliecloud to work.

Open Containers
The Open Container Initiative (OCI) provides two standards that are relevant to Slurm's --container flag:

I. OCI Bundles - The specification contains two components:
   (1) the root filesystem of the container
   (2) a JSON file containing metadata about the container

II. Container Operations - the OCI defines five container operations, and a sixth (run) is commonly used. The oci.conf file maps these abstract operations to concrete commands in a specific container runtime. The mapping for Charliecloud's runtime (ch-run) is depicted below:

Testing Containerization
I. Manually - using bash to script sbatch jobs in Slurm

```
#SBATCH --nodes=1
#SBATCH --time=0:15:00
#SBATCH --no-requeue
#SBATCH --job-name=container_test
ch-run contdir/ -- ./execute
```

Results:
(1) containerized correctly within the slurm job
(2) container commands are run the same as in CLI

II. With --container flag - using Slurm's container support

```
$srun --container /contdir/ -- echo containerized
containerized

$salloc --container /contdir/ -- /usr/bin/env
USER=root
PATH=/bin:/sbin:/usr/local/
```

Results:
(1) runc commands are tricky; tmp cannot be found
(2) $PATH has mandelbug behavior

Future Work
With more time our team could...
(1) fix mandelbug issue in $PATH
(2) test compatibility of Slurm v22.05 with oci.conf

Next steps for the project include...
(1) update documentation for Charliecloud on SchedMD
(2) parallel programming with Message Passing Interface (MPI)

What Does This All Mean?
Our team successfully prototyped an approach for automatically containerizing Slurm jobs using Charliecloud.

Developers will not need to modify to Charliecloud to implement this approach.

Users will be able to use this approach to more easily run Slurm jobs as Charliecloud containers.

Potential Limitations:
(1) It requires users to upgrade to Slurm 23.02, which does not currently enjoy wide adoption.
(2) It requires hard-coding ch-run options in the configuration file, which reduces flexibility.