Charliecloud's Successful Prototype Integration with Slurm

A Promising Approach with Some Strings Attached

Layton McCafferty
Montana State University - Bozeman
layton.mccafferty@gmail.com

Nicholas Volpe
New Jersey Institute of Technology
ncv8@njit.edu

Hank Wikle
University of New Mexico
hwikle@unm.edu

Mentors:
Reid Priedhorsky
Lucas Caudill

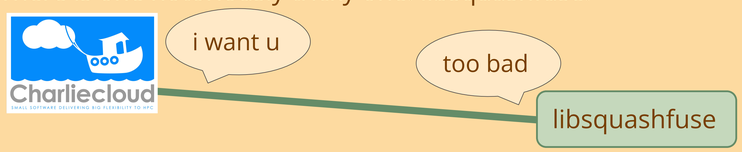
Los Alamos NATIONAL LABORATORY

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

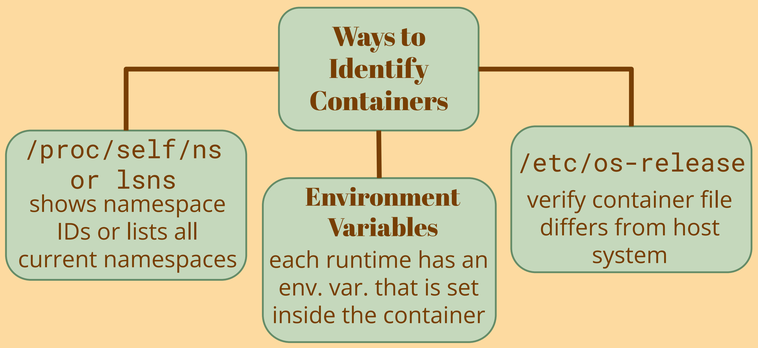
Charliecloud is lightweight; the dependencies are minimal. There is one notoriously tricky one: libsquashfuse.



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

Containerization Test Program



.Charliecloud is written in Python and runs on the Linux operating system. We had the choice of either Python or Shell code to test our containers. We wrote a shell script that tests for containerization across runtimes via environment variables. That script is below:

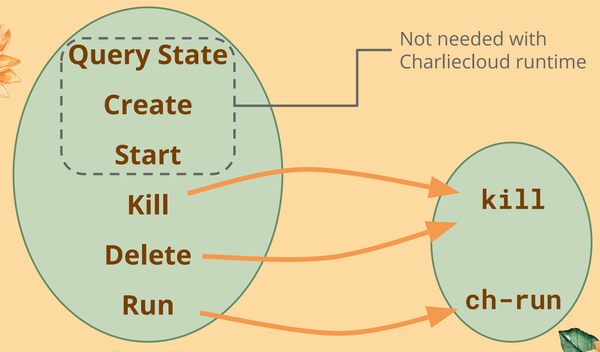
```
if [ "$runc_container" = "true" ]
then
  echo "is a runc container"
elif [ -n "$CH_RUNNING" ]
then
  echo "is a Charliecloud container"
else
  echo “container not found”
  exit 1
fi
```

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=containertest
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**

Collaboration with SchedMD

Many features involving oci.conf require Slurm 23.02, which convinced us to upgrade our Slurm version. After upgrading, we encountered an error pertaining to `slurmstepd`:

```
error: _forkexec_slurmstepd: slurmstepd failed to
send return code got 0: No error
```

dmesg revealed `slurmstepd` was segfaulting upon invocation:

```
slurmstepd[8395]: segfault at 4347b1 ip
000000000040d72e sp 00007ffd67f28660 error 7 in
slurmstepd[400000+3f000]
```

We analyzed the coredump to track the problem to a specific line in the Slurm codebase. We shared this bug with Nate Rini, the developer of the container feature, who patched the code:

> **Nate Rini** **2023-07-25 08:51:33 MDT** **Comment 27** [tag] [reply] [–]
>
> (In reply to Nicholas Volpe from comment #24)
> > Here seems to be the full thing of gdb. For a little context, it looks like
> > a library or two is missing from gdb as seen below:
>
> Thank you for the gdb outputs. I reproduced the bug and have a patch for it pending review in ~~bug#17272~~. I will update this ticket once it has been reviewed. I also noticed a second bug where the environment file was not getting cleaned up at job end and a corrective patch for it is getting reviewed in bug#17273.

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**.

Sunflower images taken from FreePik and Flaticon

MONTANA STATE UNIVERSITY
NJIT New Jersey Institute of Technology
THE UNIVERSITY OF NEW MEXICO

LA-UR-23-28983