INTEGRATING LINUX CONTAINERS INTO BATCH SYSTEMS

GOALS

- Use containers to create homogeneous job environments.
- Give users reproducible and predictable environments independent of host configuration.
- Integrate well with existing infrastructure.

IMPLEMENTATION

- Implemented in Go
- Supports Docker images.
- Has to be installed on each worker node.
- Supports Torque as the resource manager.
  - Uses the $jobstarter option in MOM config.

HOW DOES IT WORK

- Prepare container.
- Start and attach to container.
- Clean up.

PREPARE CONTAINER

- Retrieve image if needed.
- Create a container based on the image.
- Mount files on the host into container:
  - /etc/passwd and /etc/group
  - $HOME (for the user running the job)
  - The batch job.

START THE CONTAINER

- Start the container
- Stdin, Stdout and Stderr is passed on from jobstarter to container.

CLEAN UP

- Detach from the container.
- Remove the container.

FURTHER WORK

- Create an image for running jobs.
- Test
  - Complex jobs
  - Lengthy jobs
- Evaluate security risks
- Support other container implementations
  - LXC
  - rkt
  - runC

HELP WANTED

- How can the end user pass config to the jobstarter?
  - Image to use.
  - Where to get the image.
- Complex example jobs.
- To what extent can containers namespaces and cgroups interfere with existing infrastructure?
- Make it possible for containers to communicate (jobs with several processes).

QUESTIONS