Hiding All the Details: Running Grid Jobs Inside Docker Containers on the OSG

Derek Weitzel, John Thiltges, Brian Bockelman
University of Nebraska - Lincoln

Hiding the Details

- Going back to the 1980's, HTCondor strives to have the job runtime environment be run and defined by the submit host.
    - This is surprisingly difficult to do - look at the limitations (and hence popularity) of the standard universe.
- Why is this a good idea? Two examples:
    1. Enable OS updates independent of job environment
        - Sysadmins may want to run RHEL6
    2. Allow users to define their own execution environment
        - Special environments for applications

Large Hammer, Small Problem

- Great! Let's use the VM universe!
  - Virtual machines are *hard* to author - existing tools are poor and user-unfriendly.
  - Virtual machine environments are *large* (in MB).
  - Potentially significant overheads - especially in IO.
- Ouch!

Recent History

- Greg Thain gave a talk on isolating users.
- Used PID Namespaces & `cgroups` to isolate.
- `chroot`s are used to provide a user environment distinct from the host.

Usage at Nebraska

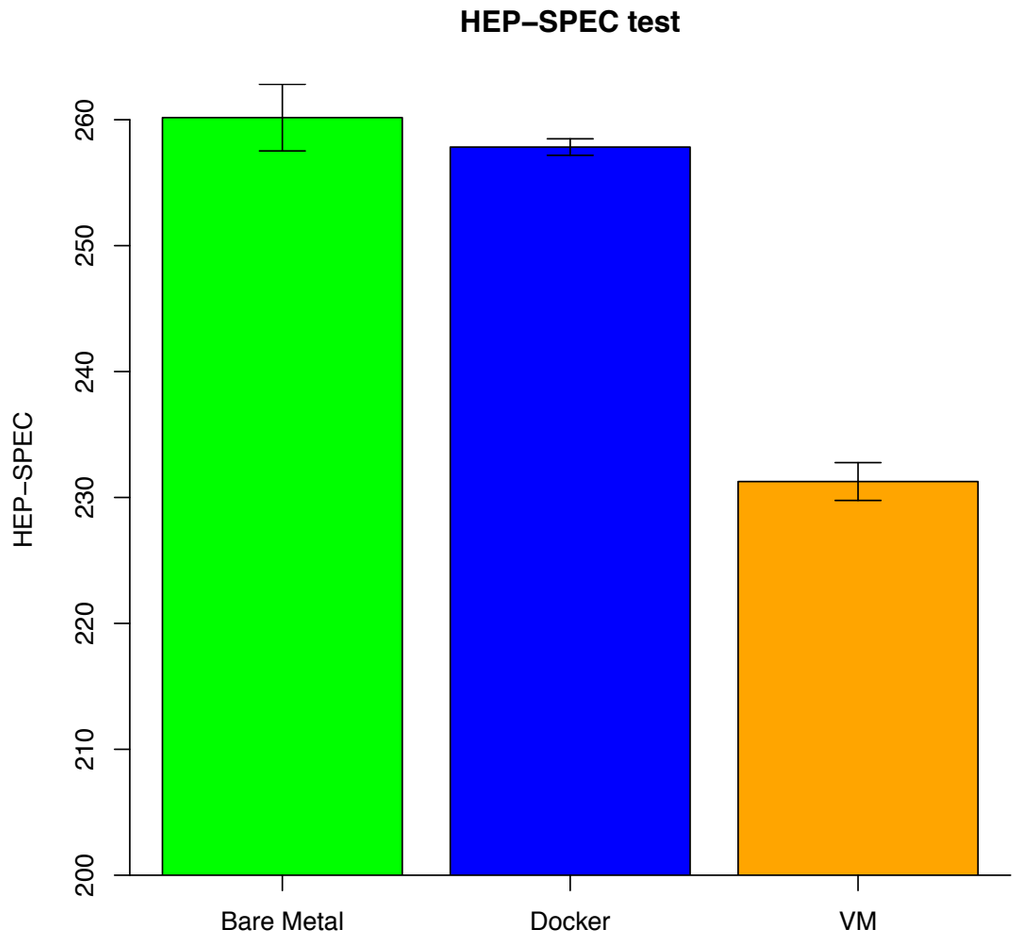
- With sufficient effort, we used the built-in techniques to manage the transition from RHEL5 to RHEL6.
  - Allowed us to transition to RHEL6 at our own rate, before all users were ready.
- Our `chroot` capability has slowly degraded over time.
  - Why? These are hard to author — like VMs, no great tooling exists to manage 'raw' `chroot`s.

A new approach: Docker

- Chroot, namespaces, and cgroups are all a part of Docker's containerization solution.
    - **IMPORTANT**: Docker provides very approachable way to compose and publish images.
        - We don't need to maintain a RHEL6 image, but only our local customizations on top.
- Decided to use HTCondor's **new** Docker universe.
- **Big picture**: transform incoming grid jobs into Docker universe jobs.

Docker Performance

- Docker, in practice, is often faster than Virtual Machines.



Robin Long (2015). Use of containerisation as an alternative to full virtualisation in grid environments.. Journal of Physics: Conference Series, 664, 022027.

Base Environment

- CentOS 7.2: This is our admin's preferred OS.
- Docker v1.9.1: Default version of Docker for RHEL7.
- HTCondor 8.5.4: Contains a few useful bug fixes and new features over the current stable series.
- We're focusing on enabling jobs from CMS and OSG: hence we'll need to layer on a few quirky customizations.
  - Not necessarily needed by others.

Default Container Setup

- Based off of CentOS 6
  - + OSG WN packages
  - + gcc, glibc-headers… for various system dependencies from CMS.
- https://hub.docker.com/r/unlhcc/osg-wn-el6/

Full Dockerfile

```
FROM centos:centos6

RUN yum -y install http://repo.grid.iu.edu/osg/3.3/osg-3.3-el6-
release-latest.rpm && \
    yum -y install epel-release && \
    yum -y install osg-wn-client osg-wn-client-glexec cvmfs && \
    yum -y install glibc-headers && \
    yum -y install gcc && \
    yum -y install redhat-lsb-core sssd-client && \
    yum clean all && \
    yum -y update

# Create condor user and group
RUN groupadd -r condor && \
    useradd -r -g condor -d /var/lib/condor -s /sbin/nologin condor

# Add lcmaps.db
COPY lcmaps.db /etc/lcmaps.db
```

That's it!

Docker Volumes

- There are a few important directories from the host that need to be available to the container - for example, the HDFS-based storage system.
    - Docker refers to these as **volume mounts**. Currently, we bring in a total of 6 different directories.
- Most volumes are marked read only - no need for the jobs to write to these.
- Exception is SSSD: need to write to a Unix socket to lookup usernames.
    - Access control to HDFS is based on Unix username: hence, we need to leak this information to the container. May not be necessary for others!

```
DOCKER_VOLUME_DIR_CVMFS          = /cvmfs:/cvmfs:ro
DOCKER_VOLUME_DIR_ETC_CVMFS      = /etc/cvmfs:/etc/cvmfs:ro
DOCKER_VOLUME_DIR_HDFS           = /mnt/hadoop:/mnt/hadoop:ro
DOCKER_VOLUME_DIR_GRID_SECURITY = /etc/grid-security:/etc/grid-security:ro
DOCKER_VOLUME_DIR_SSSD           = /var/lib/sss/pipes/nss
DOCKER_VOLUME_DIR_NSSWITCH       = /etc/nsswitch.conf:/etc/nsswitch.conf:ro
DOCKER_MOUNT_VOLUMES = CVMFS, ETC_CVMFS, HDFS, GRID_SECURITY, SSSD, NSSWITCH
```

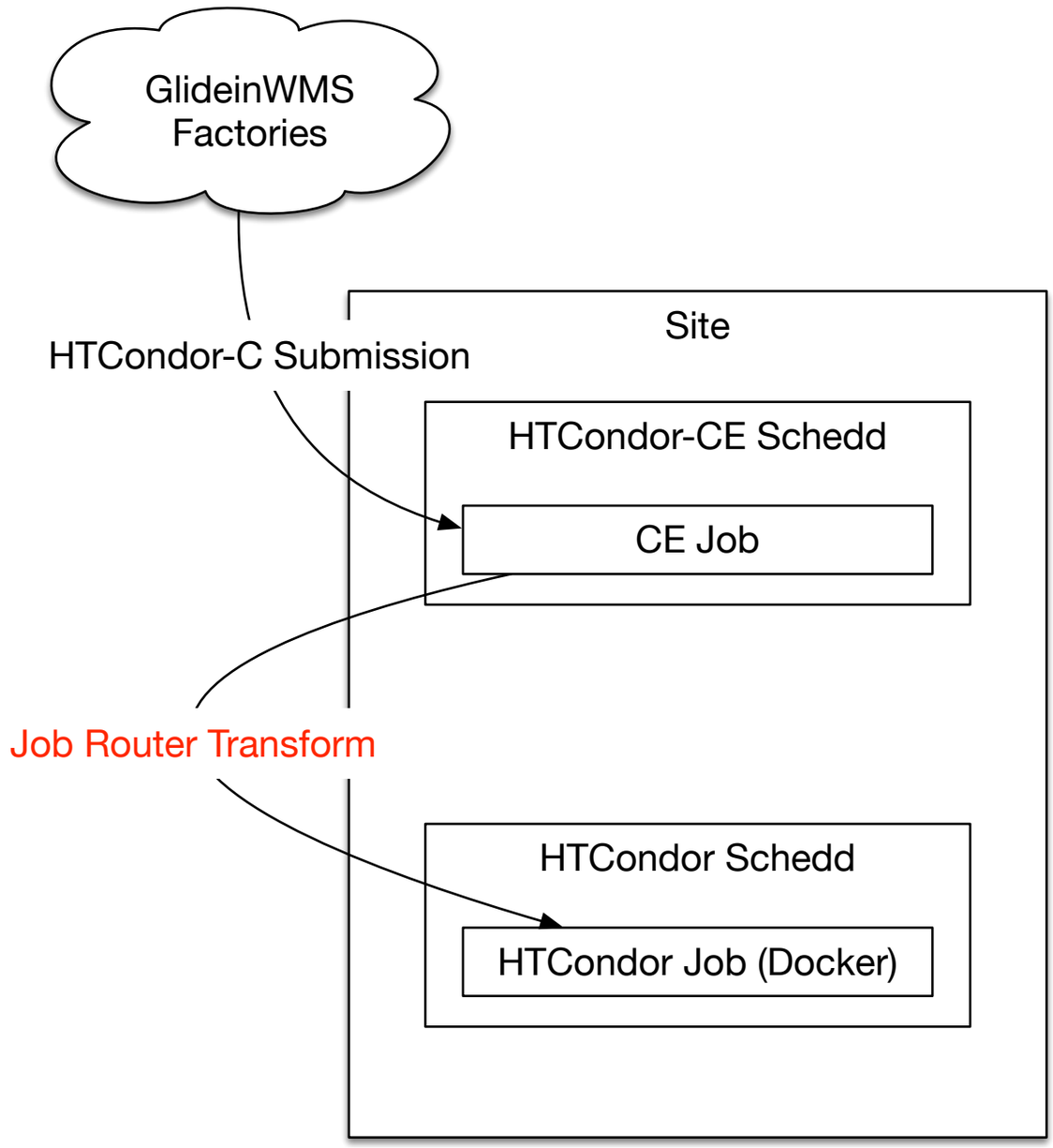
OSG Flow

The HTCondor-CE uses the `condor_job_router` to provide sites with the ability to customize jobs.

1. GlideinWMS factories submit to the HTCondor-CE.

2. Job Router component transforms the CE job to use Docker universe.

   - Surprisingly, no new `JobUniverse`.
   - Sets `DockerImage`.
   - Changes the `Cmd` string.



Job Route Configuration

- Snippets from `condor_job_router` transform language

  - Cmd needs to be prepended with ./

```
copy_Cmd = "orig_Cmd"
eval_set_Cmd = ifThenElse(regexp("^/", orig_Cmd), orig_Cmd, strcat("./",orig_Cmd))
```

- Docker image needs to be set

```
copy_DockerImage = "orig_DockerImage"
eval_set_DockerImage = ifThenElse(isUndefined(orig_DockerImage),
                                  “unlhcc/osg-wn-el6",
                                  orig_DockerImage)
```

The Full Route

This is one of multiple possible routes jobs can match

```
[ \
  GridResource = "condor localhost localhost"; \
  eval_set_GridResource = strcat("condor ", $(FULL_HOSTNAME), $(FULL_HOSTNAME)); \
  TargetUniverse = 5; \
  MaxIdleJobs = 5; \
  name = "Local_Docker"; \
  set_Requirements = ( TARGET.Memory >= RequestMemory ) && ... (remainder truncated)
  delete_PeriodicRemove = true; \
  /* Set Docker parameters */ \
  set_WantDocker = true; \
  /* If Cmd does not start with '/', prepend './' to include cwd */ \
  copy_Cmd = "orig_Cmd"; \
  eval_set_Cmd = ifThenElse(regexp("^/", orig_Cmd), orig_Cmd, strcat("./",orig_Cmd)); \
  /* Trying to directly test DockerImage failed, so we copy first */ \
  copy_DockerImage = "orig_DockerImage"; \
  eval_set_DockerImage = ifThenElse(isUndefined(orig_DockerImage), "unlhcc/osg-wn-el6", orig_DockerImage)
  /* Do not match Andrea Sciaba's various DNs against this route (all DNs use the same email address) */
  requirements = target.x509UserProxyEmail =!= "User@example.com"; \
]
```

Note **MaxIdleJobs** prevents too many OSG jobs from using this route. Limit will be lifted as we become more comfortable with Docker.

View from the worker node

```
ndor     23733  0.0  0.0 110324  8220 ?        Ss   May09   0:16 /usr/sbin/condor_master -f
ot       23760  0.2  0.0  24540  4776 ?        S    May09  31:59  \_ condor_procd -A /var/run/condor/procd_pipe -L /var/log/condor/ProcdLog -R
ndor     23761  0.0  0.0 109576  6392 ?        Ss   May09   0:13  \_ condor_shared_port -f
ndor     23762  0.1  0.0 111064  8396 ?        Ss   May09  19:22  \_ condor_startd -f
ndor   2668869  0.0  0.0 123408  7552 ?        Ss   May16   0:41  |   \_ condor_starter -f -a slot1_1 red-gw2.unl.edu
ndor   2668874  0.0  0.0 221452 14968 ?        Ssl  May16   0:14  |       \_ /usr/bin/docker run --cpu-shares=80 --memory=20000m --name HTCJob1
ndor     31579  0.0  0.0 110988  8704 ?        Ss   May09   0:44  \_ condor_startd -f -local-name sleeper
.
ot       14516  0.0  0.0 115240  1396 ?        Ss   May06   0:00 /bin/sh -c /usr/bin/docker daemon $OPTIONS           $DOCKER_STORAGE_OPTIONS
ot       14517  0.1  0.1 1007760 28272 ?       Sl   May06  16:17  \_ /usr/bin/docker daemon --selinux-enabled --storage-driver devicemapper --s
sprod  2668918  0.0  0.0  15992  1980 ?        Ss   May16   0:01  |   \_ /bin/bash /var/lib/condor/execute/dir_2668869/condor_exec.exe -v std -
sprod  2673531  0.0  0.0  15728  1692 ?        S    May16   0:00  |       \_ /bin/bash /var/lib/condor/execute/dir_2668869/glide_4SA9tV/main/co
sprod  2674340  0.0  0.0  99312 10604 ?        S    May16   0:04  |           \_ /var/lib/condor/execute/dir_2668869/glide_4SA9tV/main/condor/s
sprod  2674342  0.0  0.0  20900  3176 ?        S    May16   1:25  |               \_ condor_procd -A /var/lib/condor/execute/dir_2668869/glide_
sprod  2674343  0.7  0.0 101052 12732 ?        S    May16  10:46  |               \_ condor_startd -f
sprod  2377872  0.0  0.0  99812 10480 ?        S    03:34   0:16  |                   \_ condor_starter -f -a slot1_1 cmsgwms-submit1.fnal.gov
sprod+ 2377974  0.0  0.0  15604  1548 ?        S    03:34   0:00  |                   |   \_ /bin/bash /var/lib/condor/execute/dir_2668869/glid
sprod+ 2378009  0.1  0.0 195064 24132 ?        Sl   03:34   0:38  |                   |       \_ python2 Startup.py
sprod+ 2378111  0.0  0.0  15984  1888 ?        S    03:35   0:00  |                   |           \_ /bin/bash /var/lib/condor/execute/dir_2668
sprod+ 2378157 46.8  6.0 1968504 1489988 ?     Rl   03:35 225:17  |                   |               \_ cmsRun -j FrameworkJobReport.xml PSet.
sprod  2397825  0.0  0.0  99812 10456 ?        S    05:45   0:02  |                   \_ condor_starter -f -a slot1_3 vocms074.cern.ch
sprod+ 2397859  0.0  0.0  15604  1548 ?        S    05:45   0:00  |                   |   \_ /bin/bash /var/lib/condor/execute/dir_2668869/glid
sprod+ 2397893  0.1  0.0 193968 22028 ?        Sl   05:45   0:28  |                   |       \_ python2 Startup.py
sprod+ 2397995  0.0  0.0  15868  1740 ?        S    05:45   0:00  |                   |           \_ /bin/bash /var/lib/condor/execute/dir_2668
sprod+ 2398041 99.1  3.3 1080192 825668 ?      R    05:45 347:01  |                   |               \_ cmsRun -j FrameworkJobReport.xml PSet.
.
```

(Un)Trusted Images

- HTCondor treats all Docker images the same.
  - We want to differentiate the images that come from the “good guys” (us) versus the “bad guys” (users).
  - Still uncomfortable with the idea of allowing users to request arbitrary images.
  - RHEL7.2 includes various sandboxing mechanisms: there’s no (publicly) known ways to break out, but the track record is relatively poor.

Status

- Running Production CMS and OSG jobs
- Currently ~10% of the Nebraska Tier 2 is Docker-enabled.
- Will be expanding to the entire cluster in the coming weeks: goal is to be done by the end-of-summer.
- Next step is to further explore how to (safely) expose this capability to OSG VOs and users.

Wishlist

- Things that would simplify our setup:
    - Pass resource accounting (CPU, memory usage) from Docker to HTCondor. Scheduled for 8.5.5.
    - Avoid prepending `./` to the `Cmd`.
    - Make volume mounts conditional: we only want to expose HDFS and SSSD to CMS jobs.
- Ability to whitelist particular images - evaluated on worker node!
- Ability to mark jobs in “untrusted images” with the Linux “NO_NEW_PRIVS” flag (prevents setuid).

Questions?