Science & Technology Facilities Council

Use of containers at the RAL Tier-1

Andrew Lahiff

STFC Rutherford Appleton Laboratory

19th June, WLCG Workshop 2017, Manchester

GridPP UK Computing for Particle Physics

Overview

- Introduction
- Batch system at RAL
  - use of containers, past & present
- Cluster managers
  - Mesos
  - Kubernetes

Introduction

- RAL hosts the UK's WLCG Tier-1 centre
  - Provides computing & disk resources, tape for custodial storage of data
- Supports all 4 LHC experiments & others including
  - biomed, ENMR, ILC, MICE, LIGO, LSST, NA62, SNO+, T2K
- 2017 WLCG MoU committment
  - CPU: 212 kHS06 (24,000 cores)
  - Disk: 19 PB
  - Tape: 49 PB

Batch system

- Long history of trying to isolate jobs in our batch system
  - protect the machine from jobs
  - protect one job from another
- Back when we used Torque/Maui, things were very simple
  - jobs ran with different user ids
  - jobs which used too much memory killed
  - jobs which used too much CPU or wall time killed
  - jobs could still do bad things
    - e.g. if a job tried to use all CPUs – it could!

Batch system

- Since migrating to HTCondor in 2013, Linux kernel functionality improved our ability to isolate jobs
  - cgroups (CPU, memory, ...)
    - resource limits & monitoring
    - ensuring processes can't escape the batch system
  - PID namespaces
    - processes in a job can't see any other processes on the host
  - mount namespaces
    - /tmp, /var/tmp inside each job is unique
- This generally worked very well
  - but something was missing...

Batch system

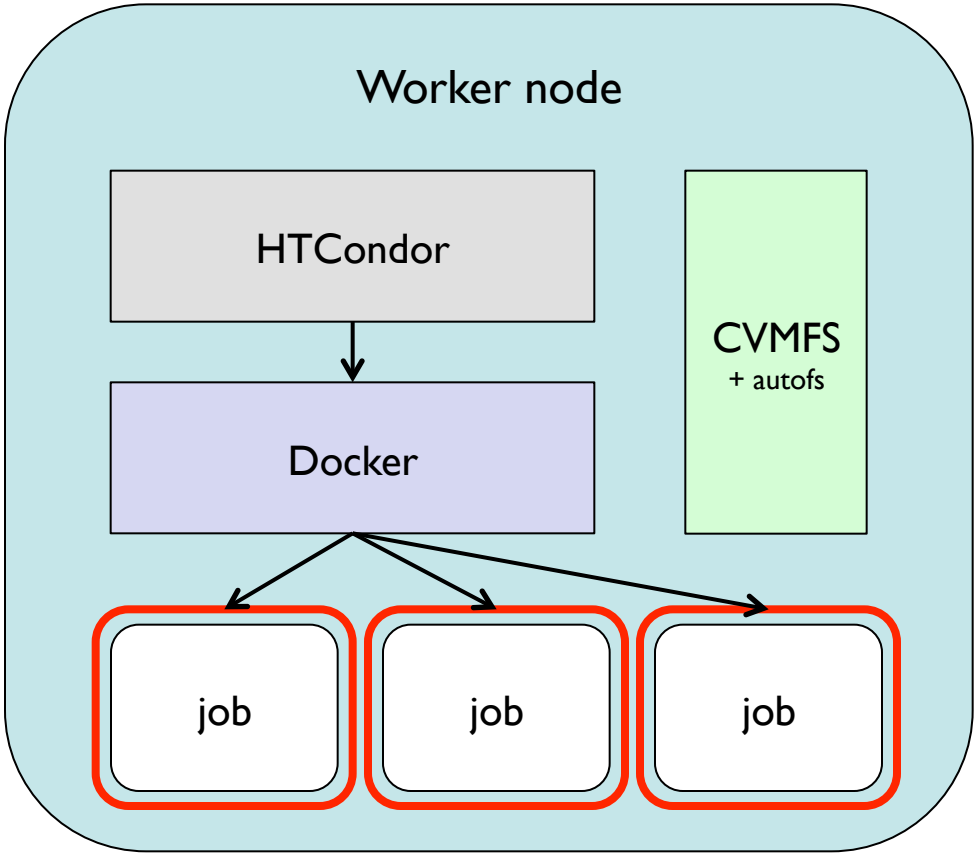
- Limitation: all jobs use the same root filesystem as the host
  - i.e. jobs tied to the host OS
    - an SL6 host can only run SL6 jobs
    - software/OS dictated by majority of LHC experiments
    - if LHC experiments want to migrate to SL7 at different times, need to partition resources
    - what if other communities want something different?
  - it would be very good to get around this limitation
- Possible solution? HTCondor's *named chroot* functionality
  - specify a directory containing an alternative root filesystem
  - a bit awkward to use & never really took off

HTCondor Docker universe

- Docker universe
  - introduced in HTCondor 8.3.6 in June 2015
  - HTCondor runs each job in a Docker container
    - Docker makes it easy to create & manage images
  - successfully ran LHC jobs at RAL in 2015
    - jobs in SL6 containers on SL7 worker nodes
- (Some) features
  - can bind-mount directories/files from the host
    - useful for CVMFS, configuration files
  - all Linux capabilities dropped by default
    - needs to be disabled for jobs requiring glexec

HTCondor Docker universe

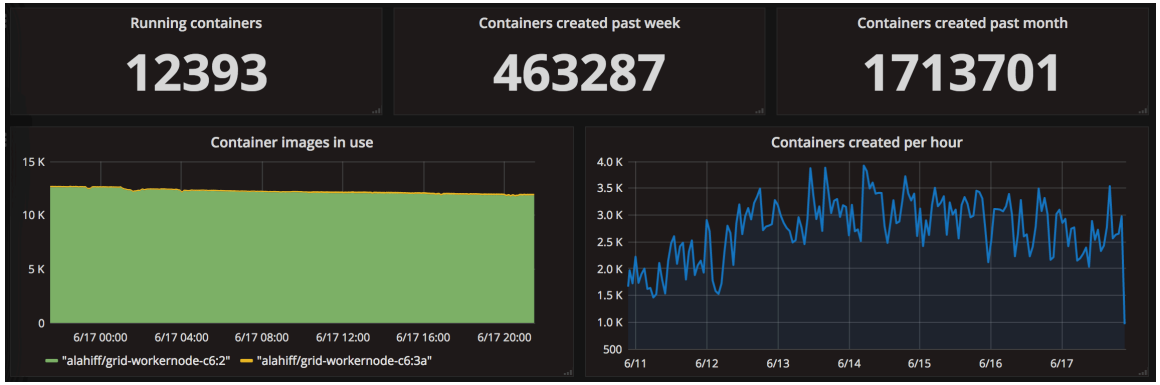
- A RAL SL7 worker node



- containers run as unprivileged pool account users
- users don't have access to the Docker daemon at all
- no way for users to specify arbitrary images via the Grid
- CVMFS available in containers using bind mounts (shared mount propagation)

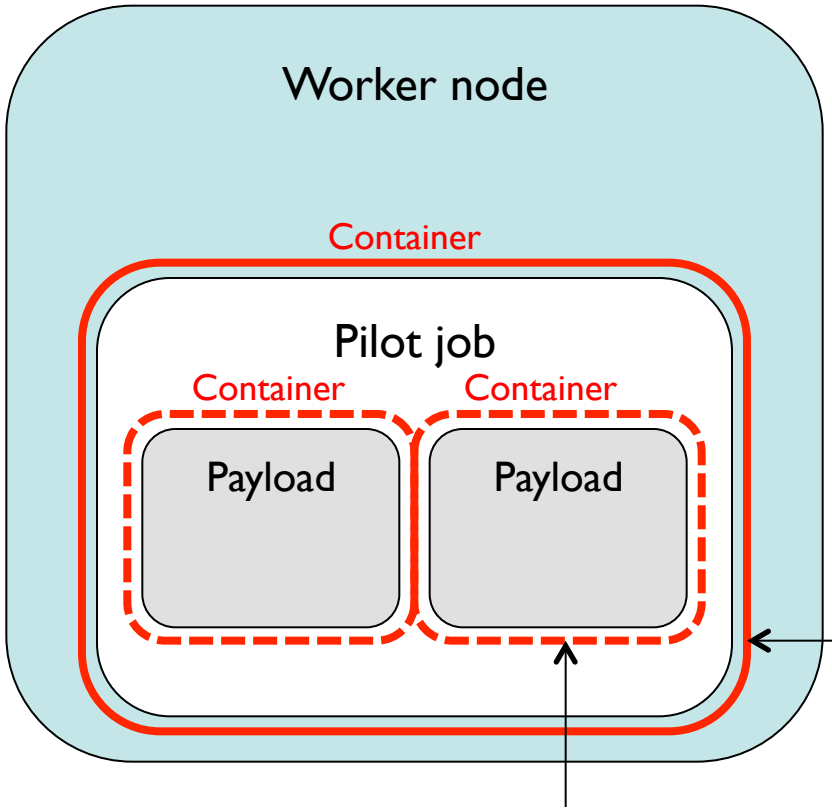
HTCondor Docker universe

- This year we migrated fully to the Docker universe
  - all jobs run in containers on bare metal
  - migrated slowly over a period of a few months
  - all existing functionality preserved, e.g. glexec, machine job features, CPU accounting, ...
- Some statistics
  - ~400K containers per week
  - 1.7M past month



Singularity

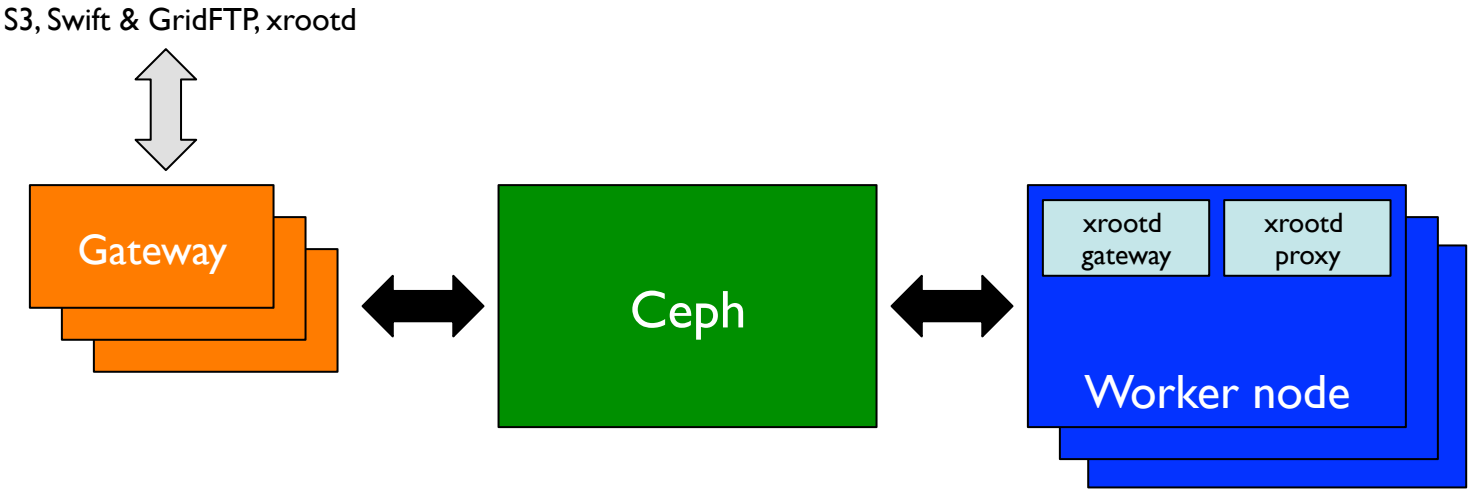
- Will provide Singularity in Centos 7 containers
  - allow VOs such as CMS to run payload jobs in containers



- payload jobs cannot see other processes on the host or even processes from the pilot
- payload jobs cannot see any files from the pilot

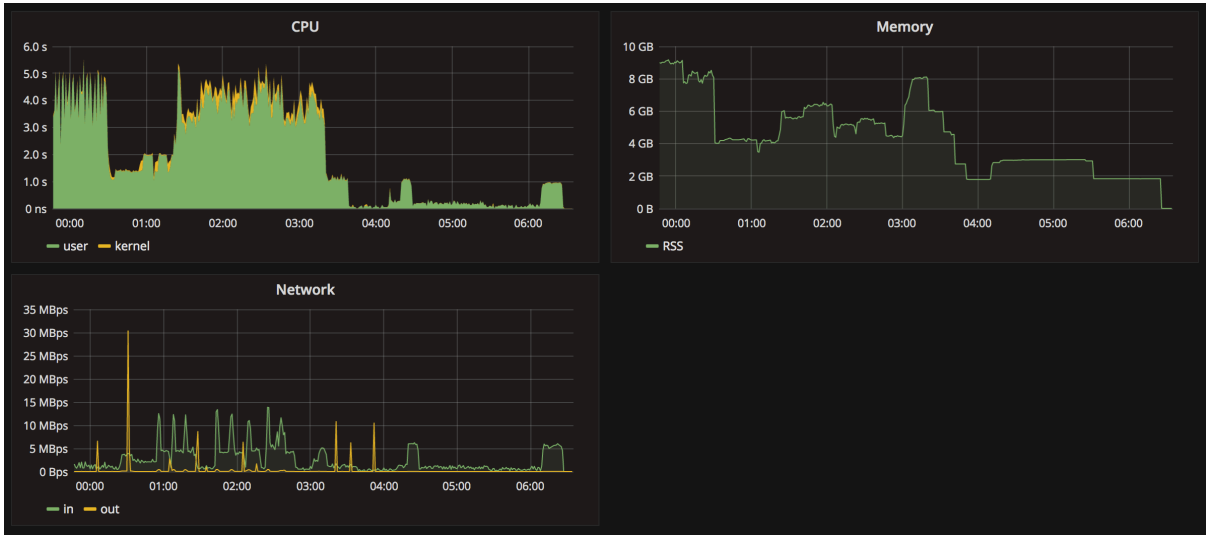
Worker nodes & storage

- Started rolling out xrootd Ceph gateways & proxies onto worker nodes
  - migrating from CASTOR to Ceph for disk-only storage
  - an important driver for migrating to SL7 worker nodes
    - jobs access data via the local gateways
    - highly scalable xrootd access to Ceph
  - xrootd daemons running in containers on each WN



Monitoring & traceability

- Containers give greater visibility into what each job is doing



Time series resource usage metrics per job, including network

```
170614 11:01:49 144 XrootdXeq: tatls011.5668:88@htcjob5609679_0_slot1_11_pid5408.ralworker pvt IP
v4 login as atlasprod
170614 11:01:49 144 acc_Audit: tatls011.5668:88@htcjob5609679_0_slot1_11_pid5408.ralworker grant
gsi atlasprod@htcjob5609679_0_slot1_11_pid5408.ralworker read atlas:scratchdisk/rucio/panda/ce/f1
/panda.0614095933.558317.lib._11488760.9783001038.lib.tgz
170614 11:01:59 144 XrootdXeq: tatls011.5668:88@htcjob5609679_0_slot1_11_pid5408.ralworker disc 0
:00:10
```

Local xrootd gateway access to storage by user & by job

Network connections per job

Towards the future

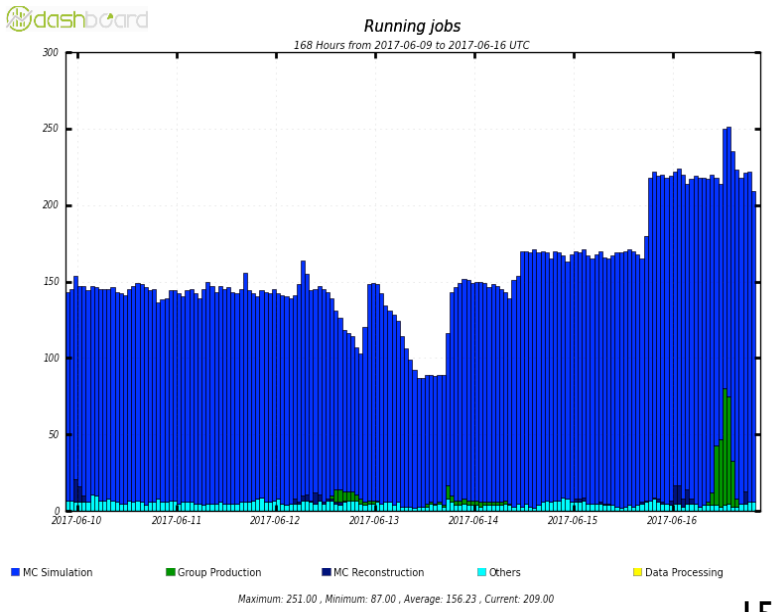
- Since on worker nodes we're
  - running jobs in containers
  - running xrootd servers in containers
- Why not just run everything in containers?
  - just doing this on its own wouldn't give many benefits
- However, if the containers are managed by schedulers
  - instead of having just a dedicated HTCondor batch farm, the same nodes could be used for
    - Big Data, HPC, cloud hypervisors, ...
  - gain lots of more flexibility, help support a wider range of activities
    - 'new' communities becoming more and more important

Mesos

- Mesos is a cluster manager which
  - enables a large group of machines to appear as a single pool of resources
  - allows you to have multiple schedulers sharing the same resources
- Have had a Mesos cluster running for around 2 years
  - varied in size from 256 to over 7000 cores (currently 352)
- What has it being used for?
  - originally concentrated on investigating the benefits of container orchestration for long-runnng services
  - more recently looking at providing flexible computing infrastructure

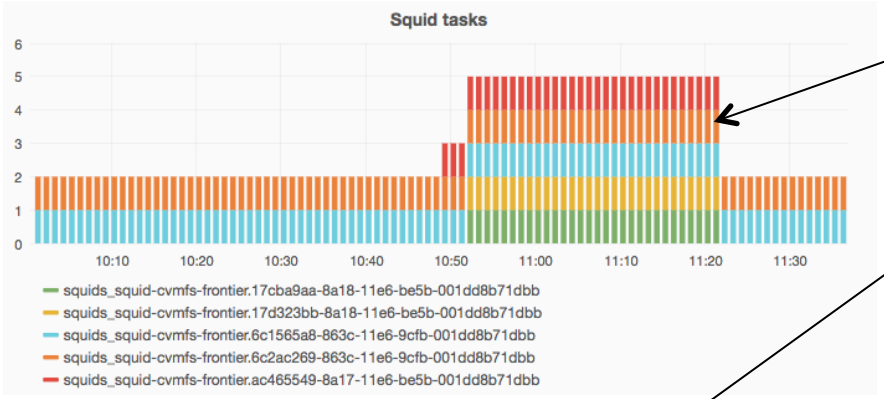
Mesos

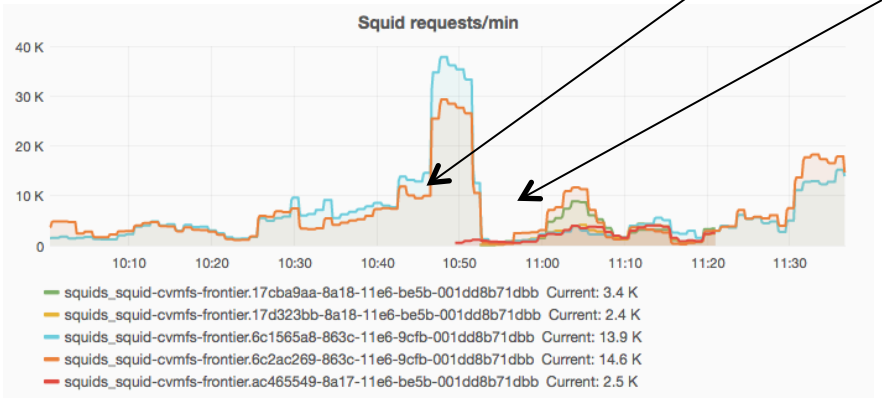
- Last year did tests running > 5000 cores of jobs from all LHC experiments
  - startds + CVMFS running in containers on Mesos joining our production HTCondor pool
- Currently an improved version is running real ATLAS jobs
  - CVMFS provided by (privileged) containers
  - startds in unprivileged containers join a CERN HTCondor pool



Mesos

- Example: number of squid instances changing based on load (request rate)
  - scale up quickly, scale down slowly





Each colour corresponds to a unique squid instance in a container

Spike in request rate triggers creation of additional squid instances

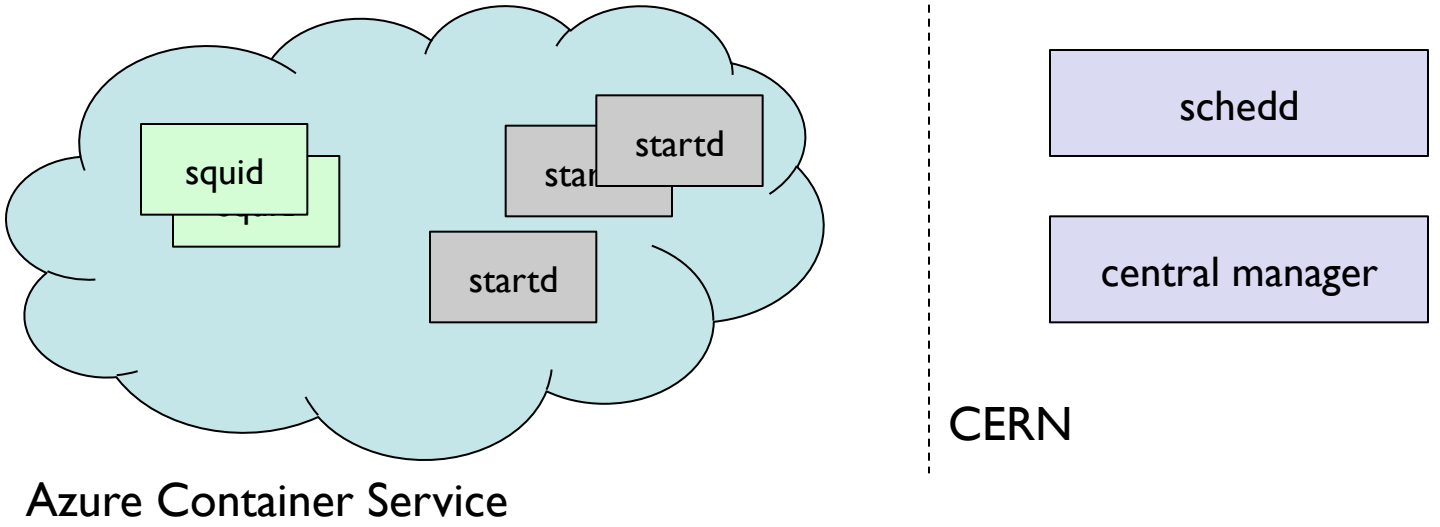
Drop in request rate therefore number of squid instances will be reduced

Kubernetes as an abstraction layer

- Kubernetes is an open-source container cluster manager which can be run anywhere
  - on-premises
  - "as a service" on public clouds (natively or via 3rd parties)
- Using it as an abstraction to enable portability between
  - on-premises resources
  - multiple public clouds
- Benefits compared to traditional ways of using public clouds
  - Don't need to worry about handling different cloud APIs
  - Run a *single command* to create an elastic, self-healing Grid site
    - on a single Kubernetes cluster
    - on multiple clusters around the world (via Kubernetes federations)

Kubernetes as an abstraction layer

- Did initial testing with CMS CRAB3 analysis jobs
  - RAL, Google (GKE), Azure (ACS), AWS (via StackPointCloud)
- Now running ATLAS production jobs on Azure
  - using "vacuum" model for independently creating startds which join a HTCondor pool at CERN



Thanks to Microsoft for an Azure Research Award

Summary

- Containers are being used a lot at RAL in production
  - migrated our HTCondor batch system to run all jobs in Docker containers
  - have started rolling out xrootd gateways to Ceph in containers on worker nodes
- Other efforts at RAL involving containers
  - providing more flexible computing infrastructure
  - making it easier to use public clouds

For more information

- HTCondor
  - https://indico.cern.ch/event/611296/contributions/2608192/attachments/1469768/2280587/EuroHTCondor_ALahiff.pdf
- Mesos
  - https://indico.cern.ch/event/384358/contributions/909266/attachments/1170757/1690077/HEPiX2015_MesosAtRAL.pdf
  - https://indico.cern.ch/event/505613/contributions/2227447/attachments/1347485/2041461/Oral-531.pdf
- Kubernetes
  - http://indico4.twgrid.org/indico/event/2/session/45/contribution/143/material/slides/0.ppt