Science & Technology
Facilities Council

Virtualisation & Containers
(A RAL Perspective)

Andrew Lahiff

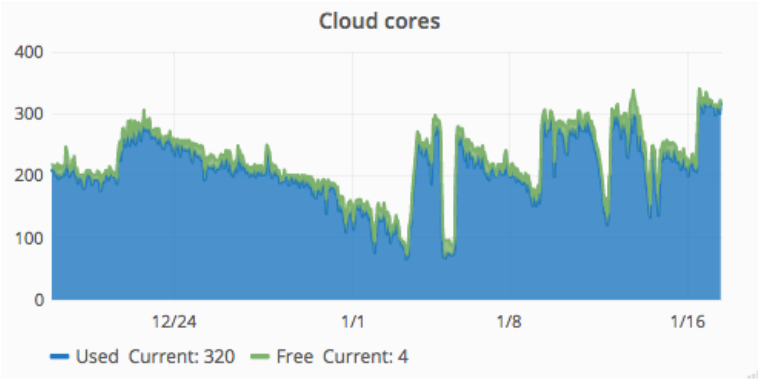
ATLAS Sites Jamboree, 18th January 2017

Overview

- Virtualisation
- Containers
  - SL7 HTCondor worker nodes
    - Docker
    - Singularity
  - Mesos
  - Kubernetes

Virtualisation at RAL

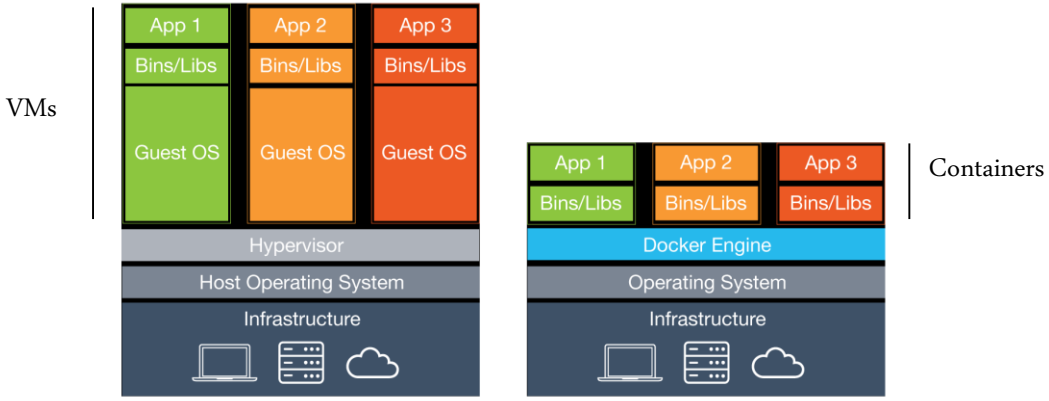
- HyperV
  - grid services etc
- OpenNebula
  - ~892 cores, Ceph storage backend
  - for almost 2 years the RAL HTCondor pool has made opportunistic use of unused cloud resources
  - HTCondor Rooster daemon provisions VMs as needed
- OpenStack
  - service under development
  - will replace OpenNebula this year



Cloud usage by HTCondor over past month, generally up to ~300 cores are used

Containers

- What is a container?
  - a container consists of an application and all its dependencies which can be run in an isolated way
  - make uses of kernel features (cgroups, namespaces, ...)
- Benefits include
  - independence from host OS & libraries
  - can be run anywhere, regardless of kernel version or host Linux distribution



Batch systems

- Use of containers on worker nodes can be very useful
  - Jobs are decoupled from the OS & libraries on the hosts
  - Jobs can have exactly the same environment no matter what site they're running at
  - More flexibility for sites
    - site can upgrade worker node OS without affecting VOs
    - different VOs can use different OSs if necessary
- For sites using HTCondor as a batch system, there are now 2 ways of running containers
  - Docker universe
  - Singularity (new)

HTCondor Docker universe

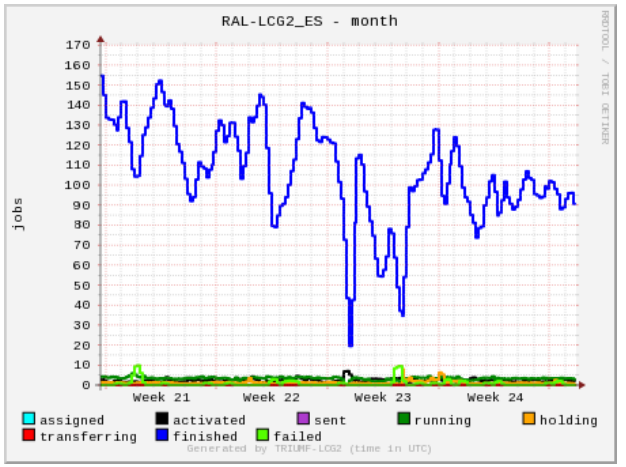
- Allows jobs to be executed inside Docker containers
  - requires Docker engine to be installed & running on each worker node
- Available in HTCondor since 8.3.6, but use a more recent version!
  - 8.5.5: CPU & wall time now reported correctly
    - important for accounting
  - 8.5.8: allows admins to specify what volumes to mount in containers using an expression
    - e.g. make ATLAS CVMFS repository only visible to ATLAS jobs

HTCondor Docker universe

- First successfully tested at RAL with LHC jobs in 2015
- SL7 worker nodes with the following installed
  - Docker
  - HTCondor
  - CVMFS (**without autofs**)
  - CA certificates
  - fetch-crl
  - pool accounts
- Some directories & files bind-mounted from host into containers
  - /cvmfs, /etc/grid-security
  - /etc/passwd, /etc/group

HTCondor Docker universe

- Networking
  - each container has its own network interface
  - stats available for network I/O per job
  - no ports exposed
    - if someome tries to start a server in a job, it won't be able to accept any incoming connections
    - prevents rfio from working (e.g. rfcp from worker node to SE)
- Docker universe in production
  - we've successfully run real jobs from all LHC VOs
  - Nebraska T2 migrated fully last year



Example of ATLAS jobs running in SL6 containers at RAL

Singularity

- Project from Berkeley Labs (http://singularity.lbl.gov/)
- Designed to allow non-privileged users on HPC systems to provide their own OS
  - isolates filesystem & processes
  - no daemon
  - no UID switching
- Being pushed by Brian Bockleman & the WLCG Traceability & Isolation Working Group
  - seen as a (future) alternative to glexec
  - provides isolation but not traceability
    - payload cannot attack pilot or other payloads on same host

Singularity

- An SL6 CernVM environment can be obtained from CVMFS
  - worker nodes have CVMFS anyway
  - no additional container image needed
  - Singularity can be configured to bind mount CVMFS into the containers

```
[alahiff@vm49 ~]$ cat /etc/redhat-release
Scientific Linux release 7.2 (Nitrogen)

[alahiff@vm49 ~]$ singularity shell /cvmfs/cernvm-prod.cern.ch/cvm3
Singularity: Invoking an interactive shell within container...

sh-4.1$ cat /etc/redhat-release
Scientific Linux release 6.8 (Carbon)

sh-4.1$ ls /cvmfs/
cernvm-prod.cern.ch  cms.cern.ch  grid.cern.ch
```

Singularity

- Initial Singularity support added to HTCondor in 8.5.8
  - condor_startd automatically detects if Singularity is installed & functional
  - jobs can be run in Singularity containers
  - admin can specify
    - an expression to decide what image to use
    - an expression to decide what volumes to mount
- Have done some basic tests at RAL (this week)
  - have not yet tried running jobs from an LHC VO in this way
- Has the potential to be the simplest way for sites to run jobs in SL6 containers on SL7 worker nodes

Mesos

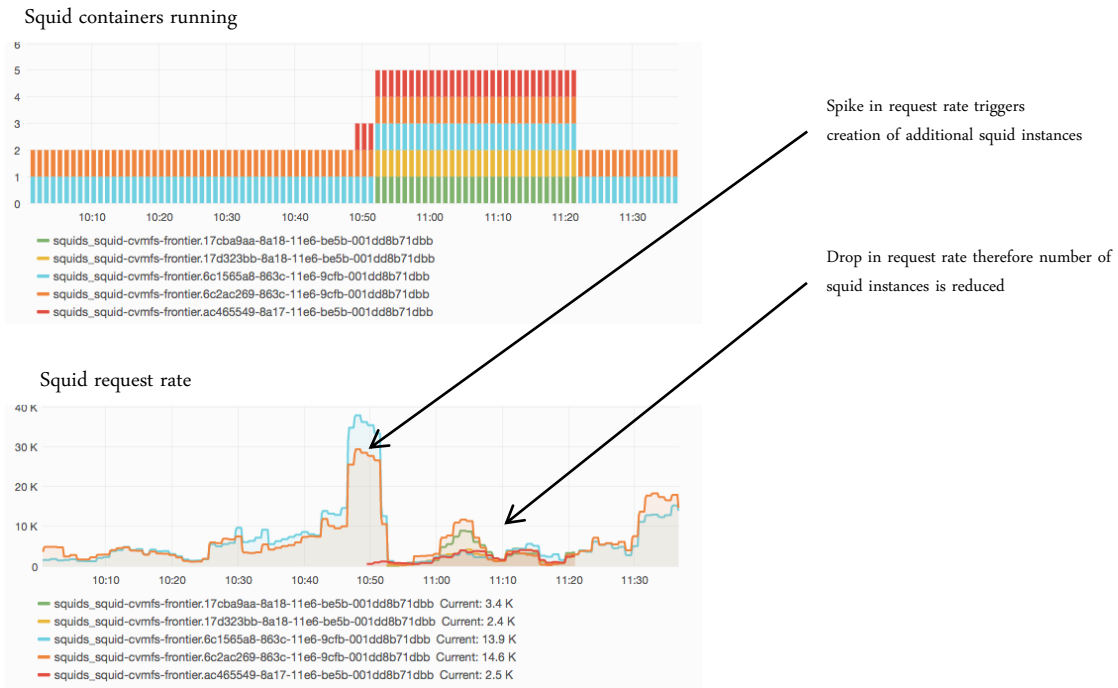
- Apache Mesos is a container cluster manager which
  - enables a large of machines to appear as a single pool of resources
  - allows you to have multiple schedulers sharing the same resources
- Have been investigating Mesos as a platform for compute & services at RAL
- Motivation
  - We need to become more flexible
    - broadening user-base of the UK Tier-1: “non-LHC” & local facilities becoming more important
  - Staff effort more likely to decrease than increase
    - need to be more efficient

Mesos

- Benefits of using Mesos
  - more efficient utilisation of resources
    - multiple activities sharing the same resources
    - e.g. LHC jobs, “Big Data”, long-running services, ...
  - significantly higher levels of automation than a traditional platform, e.g.
    - self-healing
    - auto-scaling
    - automated rolling upgrades
    - application-specific scheduling
  - potentially higher availability with less effort

Mesos

- An example
  - last year had over 5000 concurrent real jobs from all LHC VOs running in containers on Mesos
  - also used squids running in containers on Mesos



Kubernetes

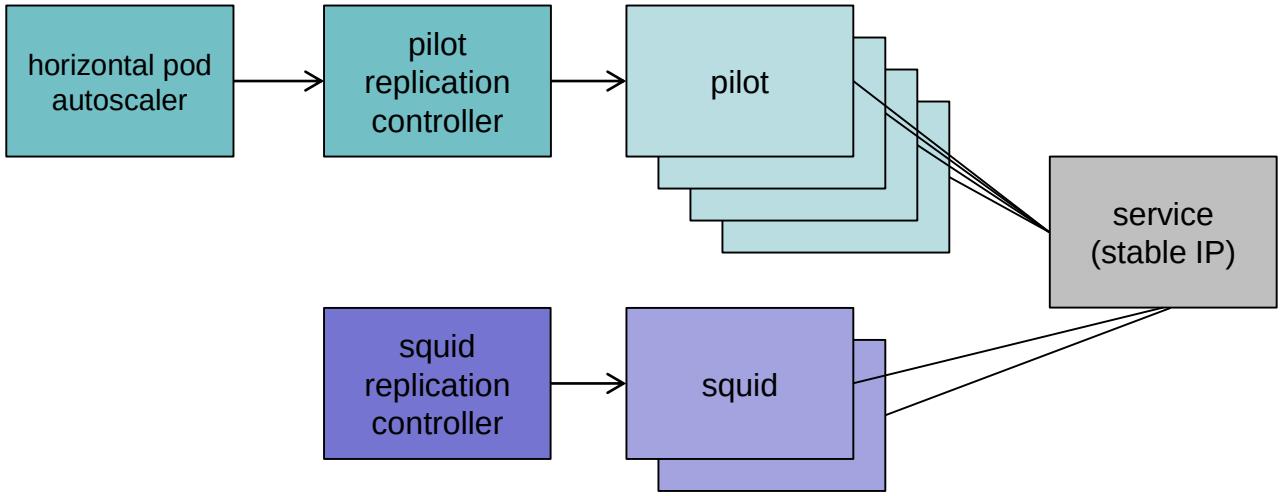
- RCUK Cloud Working Group Pilot Project investigating
  - portability between on-prem resources & public clouds
  - portability between multiple public clouds
- Using particle physics workflows as an example
  - in particular CMS (so far)
- Kubernetes
  - Open source container cluster manager, originally developed at Google
  - Can be installed on-prem (bare metal or on a cloud), also available on public clouds
    - “click a button” on Google & Azure
    - straightforward to install on AWS, ...

Kubernetes

- Why Kubernetes?
  - it's standard open-source software (not HEP-specific)
  - using it as a means of abstracting differences between on-prem resources & different public clouds
  - eliminate vendor lock-in by avoiding any cloud or vendor specific APIs
    - no need to write software to talk to different cloud APIs, just use the Kubernetes API only
  - also has federation functionality making it easy to deploy workloads across multiple clusters (new)

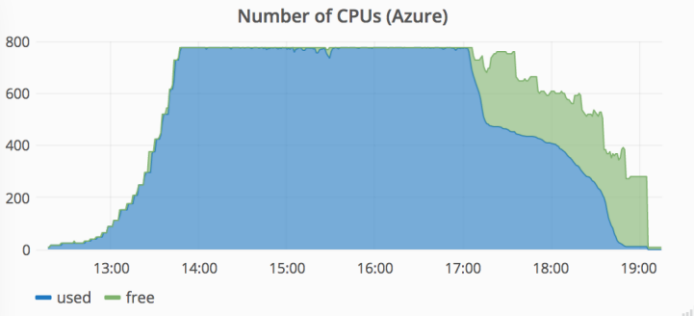
Kubernetes

- How Kubernetes is being used to run CMS jobs
  - create a pool of pilots which scales automatically depending on how much work is available (“vacuum” model)
  - squids (currently static, but could autoscale)
- Uses only existing functionality within Kubernetes



Kubernetes

- Initial testing running CMS jobs on 3 clusters
  - RAL (on bare metal)
  - Google Container Engine*
  - Azure Container Service**
- Identical containers & identical Kubernetes configuration
- Testing has included
  - MC production workflows (CPU intensive)
  - more recently MC reprocessing workflows (I/O intensive)



Number of used & idle cores in glideinWMS HTCondor pool with 8-core pilots (example of autoscaling pilots)

* Thanks to Google for credits ** Thanks to Microsoft for an Azure Research Award

Summary

- Use of containers beneficial for both VOs and sites
  - jobs no longer depend on OS version or software installed on worker nodes
  - easier to provide a consistent environment at multiple sites
- Singularity seems to be a simple way for sites to run jobs in containers
- Container cluster managers
  - Mesos can be used to provide an efficient platform for long-running services & multiple compute activities
  - Kubernetes can be used to provide portability between local resources & multiple public clouds

References

- HTCondor provisioning virtual worker nodes
  - https://indico.cern.ch/event/304944/contributions/1672235/attachments/578487/796619/IntegratingGridAndCloudResourcesRAL.pdf
- Mesos at RAL
  - https://indico.cern.ch/event/384358/contributions/909266/attachments/1170757/1690077/HEPiX2015_MesosAtRAL.pdf
  - https://indico.cern.ch/event/466991/contributions/1143587/attachments/1259900/1861490/Mesos_SpringHEPIX2016_v4.pdf
  - https://indico.cern.ch/event/531810/contributions/2328936/attachments/1359312/2056355/HEPiX2016Oct_Containers_RAL-ADL.pdf
- Kubernetes on public clouds
  - https://indico.cern.ch/event/593830/contributions/2399945/attachments/1384966/2107035/CMS_Clouds_20161207_v1.pdf
- SL7 HTCondor worker nodes
  - http://indico.cern.ch/event/518392/contributions/2182742/attachments/1296501/1933410/SL7_HEPSYSMAN2016_v1.pdf (old)