



HTCondor within the European Grid & in the Cloud

Andrew Lahiff

STFC Rutherford Appleton Laboratory

HEPiX 2015 Spring Workshop, Oxford

The Grid



Introduction

- Computing element requirements
 - Job submission from LHC VOs
 - AliEn: ALICE
 - HTCondor-G: ATLAS, CMS
 - DIRAC: LHCb
 - EMI WMS job submission
 - Still used by some non-LHC VOs
 - Usage (probably) likely to decrease
 - Information system
 - Information about jobs & worker nodes needs to go into the BDII



CREAM CE

- Does not currently work out-of-the-box with HTCondor
 - Support for Condor used to exist but was dropped, some functionality still exists (e.g. BLAH)
- New set of scripts to be added to the official CREAM release is under development
 - Including info-providers, APEL parser, ...
- Support is available for any sites wishing to install CREAM & HTCondor starting from the existing scripts



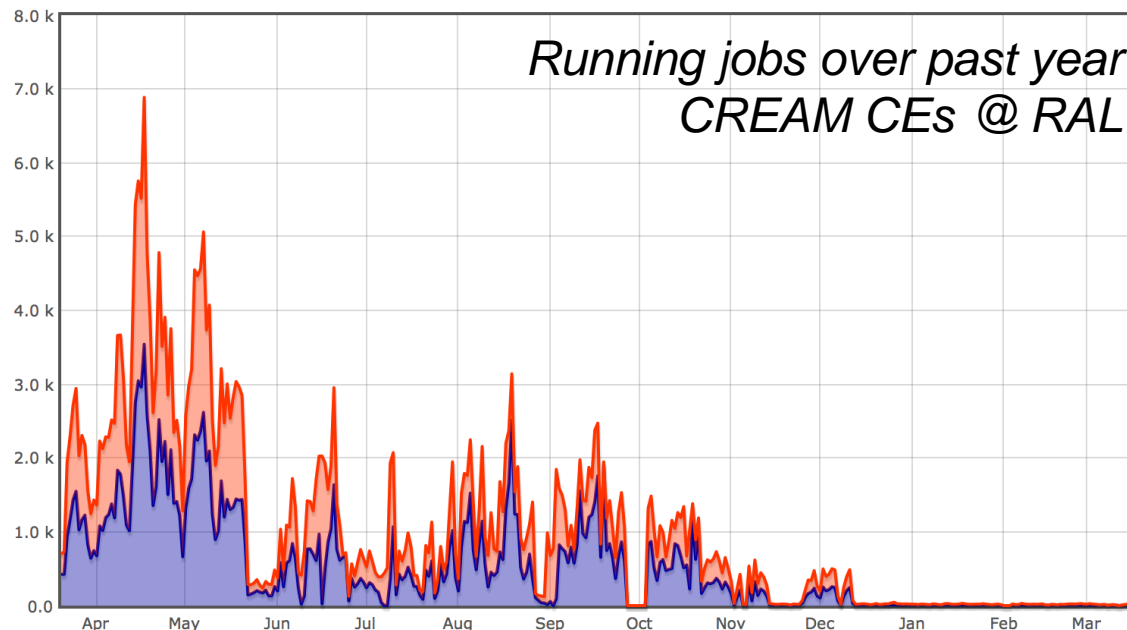
CREAM CE

- Problems with existing scripts
 - No YAIM function for configuring CE to use HTCondor
 - Scripts to publish dynamic information about state of jobs missing
 - YAIM function for configuring BLAH doesn't support HTCondor
- RAL solution
 - Made use of scripts from very old versions of CREAM which did support HTCondor
 - Updated these to work with the current EMI-3 CREAM CE
 - Needed modernizing, e.g. to support partitionable slots
 - Wrote our own APEL parser



CREAM CE

- RAL has been running 2 CREAM CEs with HTCondor in production for ~1.5 years
 - Was used by ALICE, LHCb, non-LHC VOs
 - Now only used for ALICE SUM tests, will be decommissioned soon








ARC CE

- NorduGrid product
- Features
 - Simpler than CREAM CE
 - Can send APEL accounting data directly to central broker
 - File staging: can download & cache input files; upload output to SE
- Configuration
 - Single config file `/etc/arc.conf`
 - No YAIM required



ARC CE

- Can the LHC VOs submit to ARC?
 - ✔  ATLAS
 - Use HTCondor-G for job submission
 - Able to submit to ARC
 - ARC Control Tower for job submission
 - ✔  CMS
 - Use HTCondor-G for job submission
 - ✔  LHCb
 - Last year added to DIRAC the ability to submit to ARC
 - ✔  ALICE
 - Recently regained the capability to submit to ARC
- Submission via EMI WMS
 - ✔  Uses HTCondor-G for job submission



ARC CE

- 4.1.0
 - Contains many patches provided by RAL for HTCondor backend scripts
- 4.2.0
 - Bug fix for memory limit of multi-core jobs (HTCondor)
- 5.0.0 (upcoming)
 - Able to make use of per-job HTCondor history files
 - Used in production at RAL for a long time now
 - Bug fix for CPU time limit for multi-core jobs
- Repository
 - Available in EMI, UMD, EPEL, NorduGrid repositories
 - At RAL we use NorduGrid



ARC CE

- At RAL, we're doing things "the HTCondor way"
- ARC CE configured to have a single queue
 - VOs need to request resources
 - Number of cores, memory, wall time, CPU time, ...
 - No problem for ATLAS and CMS
- It's of course also possible to setup queues



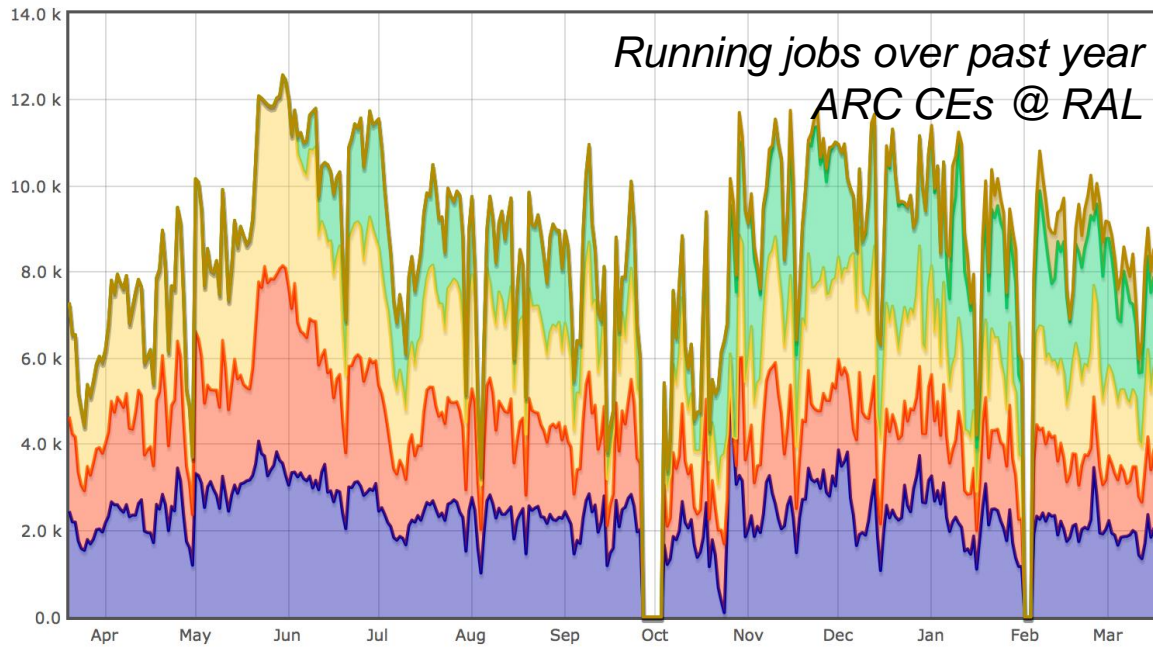
ARC CE

- Information system limitations
 - Doesn't publish per-VO running/idle jobs by default
 - With HTCondor as backend LRMS, doesn't publish max time limits
 - HTCondor has no "MaxWalltime" parameter like other batch systems
 - Instead, policy expression configured on CE (or on worker nodes)
 - Not easy (or possible?) to extract a "MaxWalltime" in general
 - We've made some small changes to `/usr/share/arc/glue-generator.pl` to overcome this



ARC CE

- The only way all 4 LHC VOs (& other VOs) now run work at RAL is by ARC CE
- RAL ARC CE usage



Accounting & scaling factors

- Most sites have heterogeneous farms
 - Need to scale CPU & wall time for correct accounting
- Torque is able to scale CPU & walltime
 - `cpumult` & `wallmult` in the `pbs_mom` config file
- HTCondor doesn't have this ability. At RAL:
 - Machine ClassAds contain scaling factors
 - Schedds configured to insert the appropriate scaling factors into the job ClassAds
 - For ARC CEs, we use an auth plugin to scale CPU & wall time before accounting records are generated
 - For CREAM CEs, our APEL parser scales CPU & wall time appropriately



HTCondor-CE

- OSG product
 - <https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallHTCondorCE>
- Special configuration of HTCondor
- Can work at sites with PBS, GE, etc, as batch system, not just HTCondor
 - Most interesting for sites with HTCondor as batch system



The Cloud



HTCondor & the cloud

- There are 2 things that can be done
 - HTCondor can manage VMs in cloud resources
 - Using the grid universe (EC2, GCE)
 - HTCondor can use cloud resources
 - Dynamic worker nodes



Managing VMs in the cloud

- Example submit description file

```
universe = grid
grid_resource = ec2 http://cloud.mydomain:4567
executable = $(ec2_ami_id)
log = $(executable).$(cluster).$(Process).log

ec2_access_key_id = /home/alahiff/AccessKeyID
ec2_secret_access_key = /home/alahiff/SecretAccessKey
ec2_ami_id = ami-000000067
ec2_instance_type = m1.large
ec2_keypair_file = $(executable).$(cluster).$(Process).pem
ec2_user_data_file = /home/alahiff/my_user_data
queue 5
```



Managing VMs in the cloud

- Submit jobs in the usual way

```
-bash-3.2$ condor_submit vms.sub  
Submitting job(s).....  
5 job(s) submitted to cluster 86.
```



Managing VMs in the cloud

- Check status of jobs (VMs)

```
-bash-3.2$ condor_q
```

```
-- Submitter: lcgwms01.gridpp.rl.ac.uk : <130.246.180.119:45554> :  
lcgwms01.gridpp.rl.ac.uk
```

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
86.0	alahiff	3/24 09:08	0+00:01:17	R	0	0.0	ami-00000012
86.1	alahiff	3/24 09:08	0+00:01:17	R	0	0.0	ami-00000012
86.2	alahiff	3/24 09:08	0+00:01:22	R	0	0.0	ami-00000012
86.3	alahiff	3/24 09:08	0+00:01:17	R	0	0.0	ami-00000012
86.4	alahiff	3/24 09:08	0+00:01:17	R	0	0.0	ami-00000012

```
-bash-3.2$ condor_q -af EC2InstanceName EC2RemoteVirtualMachineName
```

```
i-00003557 130.246.223.217  
i-00003559 130.246.223.243  
i-00003555 130.246.223.208  
i-00003558 130.246.223.219  
i-00003556 130.246.223.211
```



Dynamic resources

- Most batch systems require a hardwired list of worker nodes
 - E.g. Torque `/var/torque/server_priv/nodes`
 - Difficult to handle resources appearing & disappearing
- HTCondor
 - `startds` send updates to the collector
 - Collector maintains details about resources
 - Can configure how quickly `startds` disappear from the collector after it stops sending updates
- This makes HTCondor an ideal choice for situations when there are dynamic resources



Provisioning resources

There are number of ways that resources can be provisioned as they are needed

- Vcycle
- Cloud scheduler
- Cloud auto-scaling
 - OpenStack Heat, OpenNebula OneFlow, ...
- μ CernVM
 - Has HTCondor & ElastiQ built-in
 - ElastiQ monitors the queue
 - Requests VMs if there are idle jobs (scales up)
 - Shutdown idle VMs (scales down)



Provisioning resources

- Can also use existing HTCondor functionality
- condor_rooster
 - Designed to wake-up hibernating bare-metal worker nodes
 - Can specify an arbitrary command to run to “wake-up” worker nodes
 - Configure it to create VMs instead!
 - Can provision different types of VMs as needed
 - Single-core or multi-core
 - VO-specific VMs



Summary

- There are no blocking issues preventing European sites from migrating to HTCondor
 - Different choices of grid middleware available
 - An official release of CREAM supporting HTCondor will be available eventually
- HTCondor & clouds
 - Can manage resources on clouds
 - Can use dynamic resources easily
 - Many ways of automatically provisioning resources on clouds

