CERN

HTCondor-CE: For When the Grid is Dark and Full of Terrors

Iain Steers - CERN IT

Outline

Introduction

History

Configuration

Job Routes

Monitoring

Introduction

Our journey from CREAM and LSF to HTCondor-CE and HTCondor as our Grid offering at CERN.

Glimpse of the Past

As some of you will know, CERN has based its Grid Compute around LSF as the Batch System. However, we've had a rather frought relationship with Compute Elements (CEs).

Several years ago, when the decision was taken to move to a HTCondor batch system, several CEs were evaluated.

The ARC Compute Element

In our initial Grid Pilot of HTCondor we offered the ARC-CE as the entry point.

However, we ran into a few issues and felt like something was missing.

There was a disconnect between how we had to manage ARC, and how we wanted to manage the farm in general.

Enter HTCondor-CE

We'd heard of HTCondor-CE, however, we were under the impression it was tied to the OSG environment.

This didn't turn out to be the case. A couple of days with Brian Bockelman and we had a test CE in the pool.

Migrating VOs

CMS and ATLAS already base their submission infrastructure on condor schedds.

However, the none-OSG VOs had never worked with them in this manner before.

We embarked upon a campaign to offer our help and get the other VOs submitting via schedds.

Review

Over the next couple of months we evaluated where we stood and the pros/cons.

After familiarizing ourselves with configuring/managing the CE, we decided that the HTCondor-CE is where we wanted to take the future of the pool.

CE Configuration

HTCondor-CE is literally just a special configuration and instantiation of HTCondor running on a schedd machine.

A couple of configuration options need to be provided:

- UID_DOMAIN.
- Site-specific security overrides.
- Your job route definitions.

Puppet Sites

```
class{'::htcondor_ce'}
```

Plus some hiera.

https://github.com/cernops/puppet-htcondor_ce

Job Routes

Job Routes are a declarative approach to defining what a job looks like on your local batch system.

You take an incoming resource request from a VO and turn it into what you want a job to look like.

Job Routes ctd.

You'll want a base catch-all route and then maybe some VO/project specific routes.

The routes can be as simple or as complex as you like.

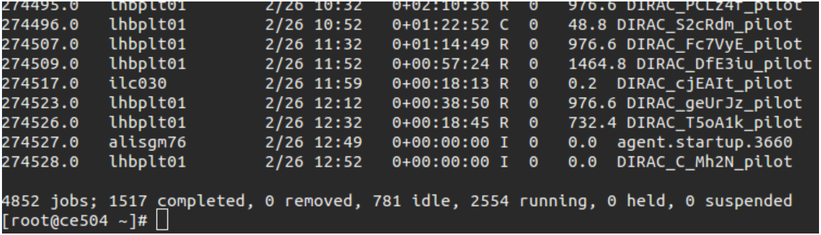
Route Example

Here's an example of our main route, although we have others.

```
JOB_ROUTER_ENTRIES = \
   [ \
    MaxIdleJobs = 4000; \
    TargetUniverse = 5; \
    name = "Local_Condor"; \
    set_AcctSubGroup = ifThenElse(regexp("production",x509userproxyfqan),strc
    set_CERNAcctGroup = toUpper(x509UserProxyVOName)); \
    eval_set_AccountingGroup = strcat("group_u_", CERNAcctGroup, ".", AcctSub
    eval_set_AcctGroup = strcat("group_u_", CERNAcctGroup, ".", AcctSubGroup)
    delete_SUBMIT_Iwd = true; \
    set_WantIOProxy = true; \
    set_default\_maxMemory = 2000; \
    set_DataCentre = "$$(DataCentre:meyrin)"; \
    set_HEPSPEC = "$$(HEPSPEC:80)"; \
  ]
```

CE Management

Simple to manage and easy to see what's going on.
CE versions of all the condor CLIs, e.g. condor_ce_q



```
274495.0   lhbplt01          2/26 10:32   0+02:10:36 R  0   976.6 DIRAC_PCL24f_pilot
274496.0   lhbplt01          2/26 10:52   0+01:22:52 C  0   48.8 DIRAC_S2cRdm_pilot
274507.0   lhbplt01          2/26 11:32   0+01:14:49 R  0   976.6 DIRAC_Fc7VyE_pilot
274509.0   lhbplt01          2/26 11:52   0+00:57:24 R  0   1464.8 DIRAC_DfE3iu_pilot
274517.0   ilc030            2/26 11:59   0+00:18:13 R  0   0.2  DIRAC_cjEAIt_pilot
274523.0   lhbplt01          2/26 12:12   0+00:38:50 R  0   976.6 DIRAC_geUrJz_pilot
274526.0   lhbplt01          2/26 12:32   0+00:18:45 R  0   732.4 DIRAC_T5oA1k_pilot
274527.0   alisgm76          2/26 12:49   0+00:00:00 I  0   0.0  agent.startup.3660
274528.0   lhbplt01          2/26 12:52   0+00:00:00 I  0   0.0  DIRAC_C_Mh2N_pilot

4852 jobs; 1517 completed, 0 removed, 781 idle, 2554 running, 0 held, 0 suspended
[root@ce504 ~]# 
```

Logging

All logs files go to **/var/log/condor-ce**.

Important log files: *JobRouterLog*, *SchedLog*, *AuditLog*

AuditLog: Anything that happened on the queue, proxies, authentication etc. Automatically configured to be kept for 90 days.

JobRouter Log

Excerpt from the Job Router Log:

```
03/02/16 09:43:24 JobRouter: Checking for candidate jobs. routing table is:
Route Name             Submitted/Max        Idle/Max     Throttle Recent: S
Local_Condor           6313/  12000         40/   4000   none             5
External_Cloud         203/  10000          160/  2000   none             3
03/02/16 09:43:24 JobRouter (src=3043640.0,dest=3951953.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043641.0,dest=3951954.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043614.0,dest=3951929.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043634.0,dest=3951957.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043636.0,dest=3951959.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043591.0,dest=3951897.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043638.0,dest=3951961.0,route=Local_Condo
03/02/16 09:43:25 JobRouter (src=3043639.0,dest=3951962.0,route=Local_Condo
03/02/16 09:43:26 JobRouter (src=3043558.0,dest=3951880.0,route=Local_Condo
03/02/16 09:43:26 JobRouter (src=3027782.0,dest=3940672.0,route=Local_Condo
03/02/16 09:43:26 JobRouter (src=3043599.0,dest=3951921.0,route=Local_Condo
03/02/16 09:43:27 JobRouter (src=3043576.0,dest=3951942.0,route=External_Cl
```

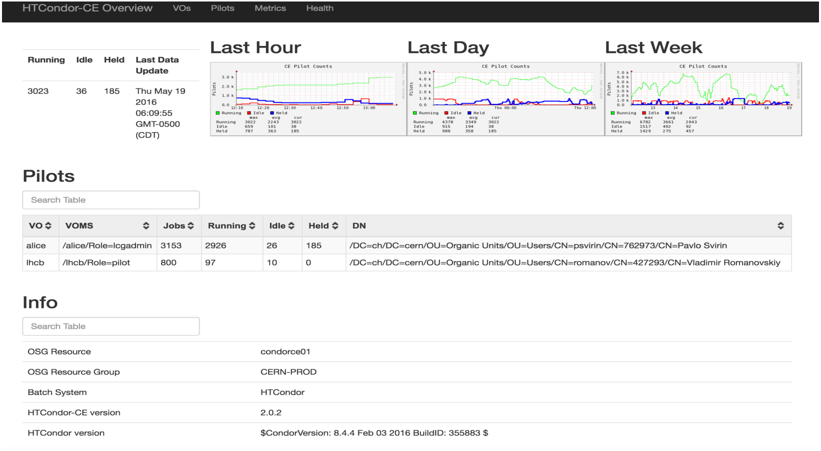
Monitoring

We monitor primarily with the python-bindings.

Make sure the CE Schedd and Batch Schedd aren't out-of-sync with job numbers.

Monitor and alarm on the Job Router run-time.

Monitoring Example



Running	Idle	Held	Last Data Update
3023	36	185	Thu May 19 2016 06:09:55 GMT-0500 (CDT)

Last Hour

	max	avg	cur
Running	3022	2243	3022
Idle	659	101	38
Held	787	363	185

Last Day

	max	avg	cur
Running	4378	3349	3022
Idle	915	194	38
Held	980	358	185

Last Week

	max	avg	cur
Running	6782	3661	2043
Idle	1517	402	92
Held	1429	275	457

Pilots

Search Table

VO ⇕	VOMS ⇕	Jobs ⇕	Running ⇕	Idle ⇕	Held ⇕	DN ⇕
alice	/alice/Role=lcgadmin	3153	2926	26	185	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=psvirin/CN=762973/CN=Pavlo Svirin
lhcb	/lhcb/Role=pilot	800	97	10	0	/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=romanov/CN=427293/CN=Vladimir Romanovskiy

Info

Search Table

OSG Resource	condorce01
OSG Resource Group	CERN-PROD
Batch System	HTCondor
HTCondor-CE version	2.0.2
HTCondor version	$CondorVersion: 8.4.4 Feb 03 2016 BuildID: 355883 $

How the VOs Interact

There are two submission methods.

- Submit via a condor_schedd running on a vobox.
- Direct submission to the CE with condor_submit.

The Schedd set-up has some great advantages and power.

More involvement from the VO framework is required for direct submission.

HTCondor Grid Universe

The schedd based submission mentioned on the previous slide relies on a special job universe called the Grid Universe.

The Grid universe allows submission to a number of classical grid systems and public clouds.

See Todd's Talk of Lies for the full list.

Grid Universe Power

Full job management semantics of usual condor.

Periodic actions, i.e. remove, hold.
Job Requirements also can be expressed.

VO Management

VO	Job Manager	Info Source
CMS	Schedd	Factory Frontend
ATLAS	Schedd	PaNDA
ALICE	Schedd (JobRouter)	ALiEN (BDII)
LHCb	Direct	DIRAC (BDII)
ILC	Schedd	DIRAC (BDII)
COMPASS	Schedd	APF

Conclusion

In conclusion, we've been incredibly happy with HTCondor-CE.

It scales nicely, is a pleasure to manage and fits perfectly with our wider needs/plans for managing the Tier-0 batch farm.

"Everything is better with some condor in the system."

— Miron ~~Lannister~~ Livny

Thanks to

We'd like to thank the following people:

- Brain Bockelman for getting us set-up with the HTCondor-CE
- Brian Lin for his development work and help.
- OSG Software Team for their work on the stack.
- The HTCondor Team for producing the meat of what makes this work.

Any Questions?

www.cern.ch