Migrating INFN–T1 from CREAM-CE/LSF to HTCondor-CE/HTCondor

by Stefano Dal Pra

2019/07/05

Email: stefano.dalpra@cnaf.infn.it

INFN-T1, Current Status (Production)

- $\sim$ 400 KHS06, 35000 slots, 850 physical hosts
- $5 \times \mathrm{CREAM-CE/LSF\,9.1.3}$
- ~ 40 User groups: 24 Grid VOs, ~ 25 local

Moving to HTC-CE/HTC

We have two clusters right now

Testbed cluster (HTC-CE 3.1.0, HTC 8.6.13)

- 1$\times$HTC-CE on top of 1$\times$CM/Collector, 3$\times$WN, 16 slot each
- Running jobs submitted by the 4 LHC experiments from Sep. 2018
- Once configured: stable and smooth, can stay unattended.

Preprod cluster (HTC-CE 3.2.1, HTC 8.8.2)

- 3×HTC-CE on top of 1×CM/Collector, 15×WN, 16 slot each
- One more WN, with 2×K-40 GPUs (ongoing tests from VIRGO, ATLAS)
- Current latest stable versions (improved GPU support and monitoring)

We plan to begin production activity starting from next days, in this order:

1. LHC VOs
2. Grid VOs using a WMS (i.e. Dirac)
3. Other VOs
4. Local submitters

Experience with HTC-CE

Installation and initial setup; what was available

- `htcondor-ce-*` RPMs are now available from the same repository of HTCondor
- https://github.com/cernops/puppet-htcondor_ce. Puppet modules available from CERN (latest commit Dec. 2016)
- https://opensciencegrid.org/docs/compute-element/install-htcondor-ce/

  Documentation and guidelines (very neat and clear, but OSG–oriented)

First CE installation

It was a bit tricky, with some "trial and error". Help, good hints and assistance have been available from the HTCondor mailing list.

– Puppet modules not directly compliant with our puppet/foreman system (these have been adapted later)

– The online documentation refers to `osg-configure` to finalize the setup

- Final setup was done manually; mainly a matter of adapting GSI authentication/authorization.
- `ui-htc ~]$ condor_ce_trace --debug ce01-htc`

  The most useful tool to track down CE problems.

In the end, the main things to fix were about GSI auth* and, later, GIP.

lcmaps, voms. The same as with CREAM-CE, except for default name and location of a few files (voms-mapfile, x509 host certificates)

condor-mapfile. Adding a regexp to match certs of your site

After this, the CE should be able to deal with first job submissions

Argus. Set up one or configure an existing one.

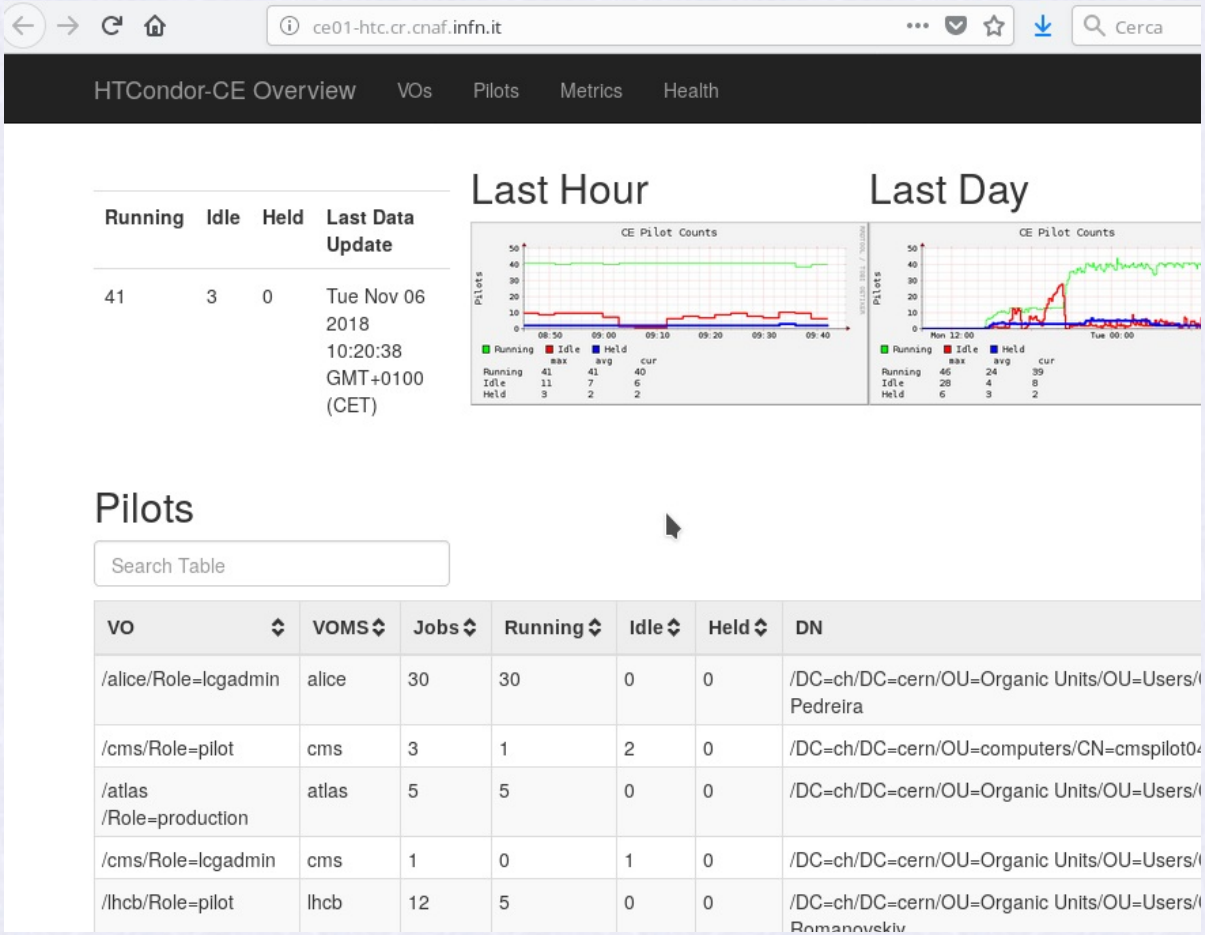
bdii. two configuration files from `htcondor-ce-bdii` rpm

Note: they are in the condor config dir, not condor-ce. Glue2 only.

A wiki for interested INFN sites is being set up with details.

Monitoring

HTCondor-CE comes with a small web tool (CEView) providing a simple interface for monitoring the activity of the CE. Enabled by editing `05-ce-view.conf`.



Accounting

We are using our own custom accounting system for six years now, so we have been considering about adapting it.

Accounting with LSF

- 

- We collect a few more data for internal use: job exit status, WN name (this is then mapped to HS06 of the node), requested resources,...
- If we can collect the same data from HTC-CE, we can re-use the other components.

Accounting with HTC-CE

First attempt: HTCondor python bindings as an alternative to `condor_history`. It works but have had timeouts at times. Enabling `PER_JOB_HISTORY_DIR` seems to be the right thing to do.

- `PER_JOB_HISTORY_DIR=/var/lib/gratia/data/`
- One accounting text file per job, `history.<jobid>` with <key> = <value> pairs, one per line.
- Each file is self consistent. It has both grid and batch data: no need for blah records, no need to find matches between sets of grid and batch records.
- python: `jobfile2dict(fn)` read a file, return a dict. The needed k/v are used to forge an INSERT query to add a tuple into our accounting DB. We collect the same set of keys (Job ClassAds in HTCondor terms) that the apel HTCondor parser collects, and a few more.
- After parsing and insert, the file is moved to a backup directory (prevents double counting).

Apel records obtained as a SQL VIEW:

```
acct=> select * from apelhtjob where ‘‘Processors’’=8 limit 1;
+-------------------------------------------------------------
Site                        | INFN-T1
SubmitHost                  | ce02-htc.cr.cnaf.infn.it#7737.0#1555345220
MachineName                 | htc-2.cr.cnaf.infn.it
Queue                       | cms
LocalJobId                  | 7737
LocalUserId                 | pilcms006
GlobalUserName              | /[...]CN=cmspilot04/vocms080.cern.ch
FQAN                        | /cms/Role=pilot/Capability=NULL
VO                          | cms
VOGroup                     | /cms
VORole                      | Role=pilot
WallDuration                | 41848
CpuDuration                 | 40549
Processors                  | 8
NodeCount                   | 1
StartTime                   | 1555345239
EndTime                     | 1555350470
InfrastructureDescription | APEL-HTC-CE
InfrastructureType          | grid
ServiceLevelType            | HEPSPEC
ServiceLevel                | 10.000
```

Requiring GPUs

Client side:

In the condor submit file:

```
request_GPUs = 1
requirements = (TARGET.CUDACapability >= 1.2) &&\
(TARGET.CUDADeviceName =?= "Tesla K40m") &&\
$(requirements:True)
```

HTC-CE side:

In the HTCondor-CE `JOB_ROUTER_ENTRIES`:

```
[name = "condor_pool_atlas";
 TargetUniverse = 5;
 Requirements = target.x509UserProxyVOName =?= "atlas";
 copy_requirements = "original_requirements";
 set_requirements = original_requirements;
...]
```

Accounting GPU usage

there are keys in the job histfile:

```
AssignedGPUs = “CUDA0"
GPUsProvisioned=1
```

Tracking these (or newer) keys

```
acct=> SELECT COUNT(*) AS “N", sum(runtime) as “WCT", username, exechosts
acct-> FROM htjob WHERE gpu=1 GROUP BY username,exechosts;
```

Which yields:

N	WCT	username	exechosts
5	4	atlas220	hpc-200-06-07
5	5	dteam039	hpc-200-06-07
3	3	sdalpra	hpc-200-06-07
17	28	virgo050	hpc-200-06-07

Conclusions

- HTC-CE works best with pilot jobs
- Currently, a few "gaps" in the documentation for non OSG people
- Can be seen as a "thin layer" on top of HTCondor:
- most of the desired behaviours are to be obtained by configuring HTCondor services, at CE side and/or batch side
- JobRouting is a very important mechanism to deal with when managing a working HTC-CE. Need some practice to get more confident with its configuration.