# Replacing CREAM-CE with HTC-CE: The INFN–T1 experience

BY INFN-CNAF

2020/09/23

*Email:* dalpra@infn.it

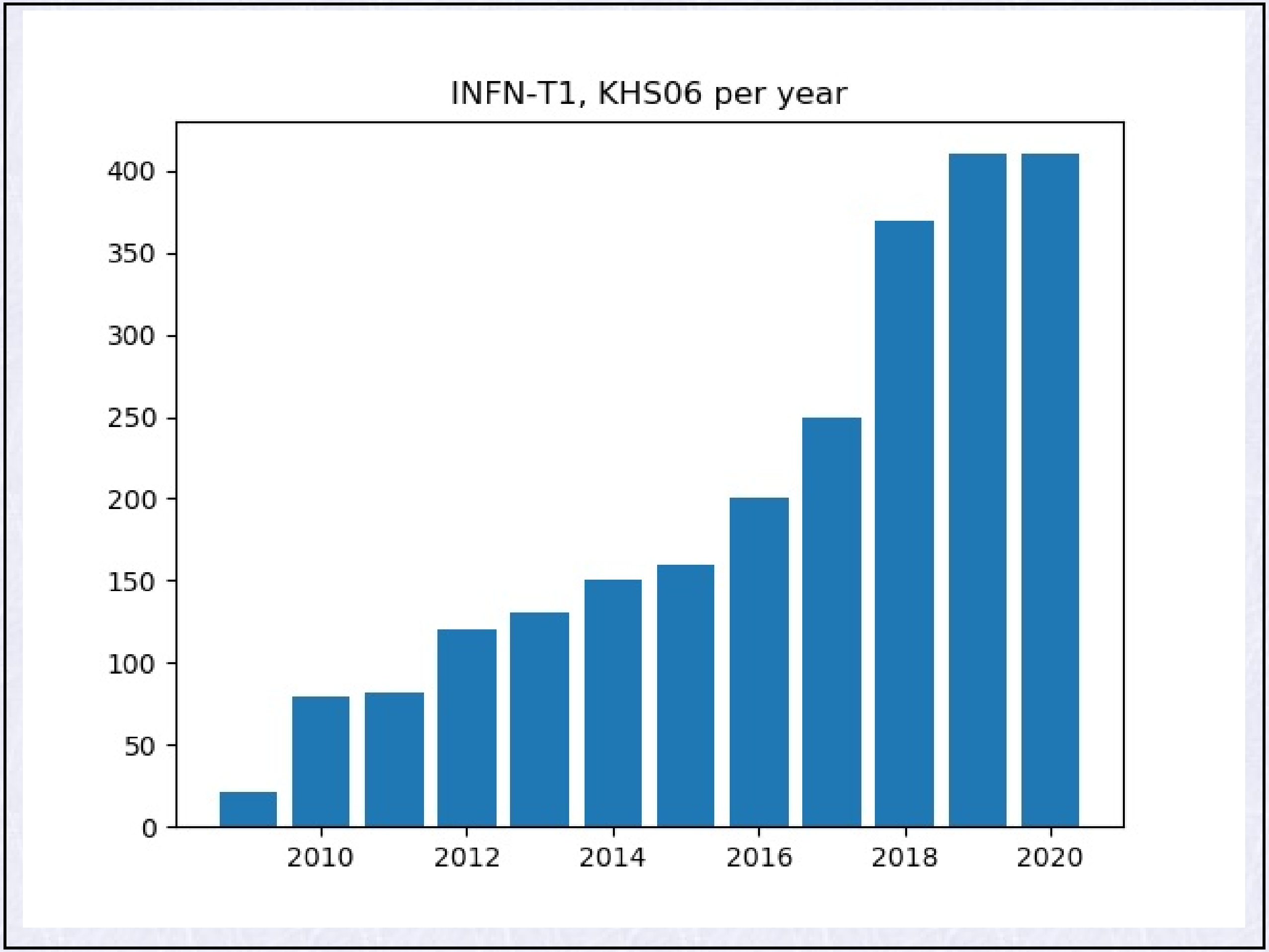
## The INFN-T1 computing farm

- $\sim 400$ KHS06, $35000$ slots, $850$ physical hosts
- $\sim 40$ User groups: $24$ Grid VOs, $\sim 25$ local
- Former: $(5+2) \times \mathrm{CREAM-CE/LSF\,9.1.3}$

  Current: $(6+1) \times \mathrm{HTC-CE/HTC}$ (from May 2020)

We report our experience with about migrating the new system, with focus on HTCondor-CE

# Our experience with CREAM-CE



- INFN-T1 had a quite exponential growth in the last $\geqslant 10$ years, like most HEP sites, ranging from $\sim$14KHS06 (2008) to $\sim$400KHS06 (2019).
- Our CREAM instances initially suffered overload and scalability problems.

- These were mitigated by adopting `lsf-btools`, (U. Schwickerath, CERN) to directly query the LSF master via `C api` instead of parsing command line output and by sharing cached results with all the CEs.
- Several more local tunings and customizations were also added during time. We started with 6+2+2 CEs but could eventually reduce to 5+2 despite the site growth.

In the end, we were pretty satisfied as it became "The good CE™": after right tuning it was able to keep up with the site growth, stable and (almost) silent. This favoured a mental bias about moving from CREAM-CE to <other>-CE.

## Facts

- LSF license renewal would stop by end of 2018, then INFN remains license owner of the latest available version, without upgrades and support.
- We had decided already that we would move to HTCondor
- CREAM-CE support ends by end of 2020 (announced Feb 2019)

## The choice for a new CE

- HTCondor-CE looks like a natural bet in our case:
  - reduce the needed knowledge base
  - further CE developments would for sure be compliant with HTC evolution
- Needing to learn a new Batch System and a new CE, we chose a "twin pair".

**Moving to HTC-CE/HTC** Of course, we started with a

**Testbed cluster** (HTC-CE 3.1.0, Spring 2018)

- 1×HTC-CE on top of 1×CM/Collector, 3×WN, 16 slot each
- early manually submitted test grid-jobs as VO dteam
- Running regular CMS jobs by Sep. 2018. Soon after, also ALICE, ATLAS, LHCb had their pilot jobs working succesfully.
- Once configured: stable and smooth, can stay unattended.

## Experience with HTCondor-CE

### Installation and initial setup

- `htcondor-ce-*` RPMs currently available from the same repository of HTCondor (initially, distinct repos)
- Puppet modules available from CERN
- OSG documentation (very neat and clear, but OSG–oriented)
- Official HTC-CE docs (new)

### 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.

- Some work needed to adapt Puppet modules to our puppet/foreman system
- The online documentation (at the time) refers to `osg-configure` to finalize the setup, which of course we miss. CREAM used to be configured by editing a `siteinfo.def` text file and passing it to a `yaim` configurator tool.

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

  was useful to track early basic setup problems.

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

**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 valid certificates

**Argus.** Set up one or configure an existing one.

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

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

## Documenting work

A wiki for interested INFN sites was set up back then with a few details. A bit out of date now, updates in progress. Goal: providing notes for a manual setup.

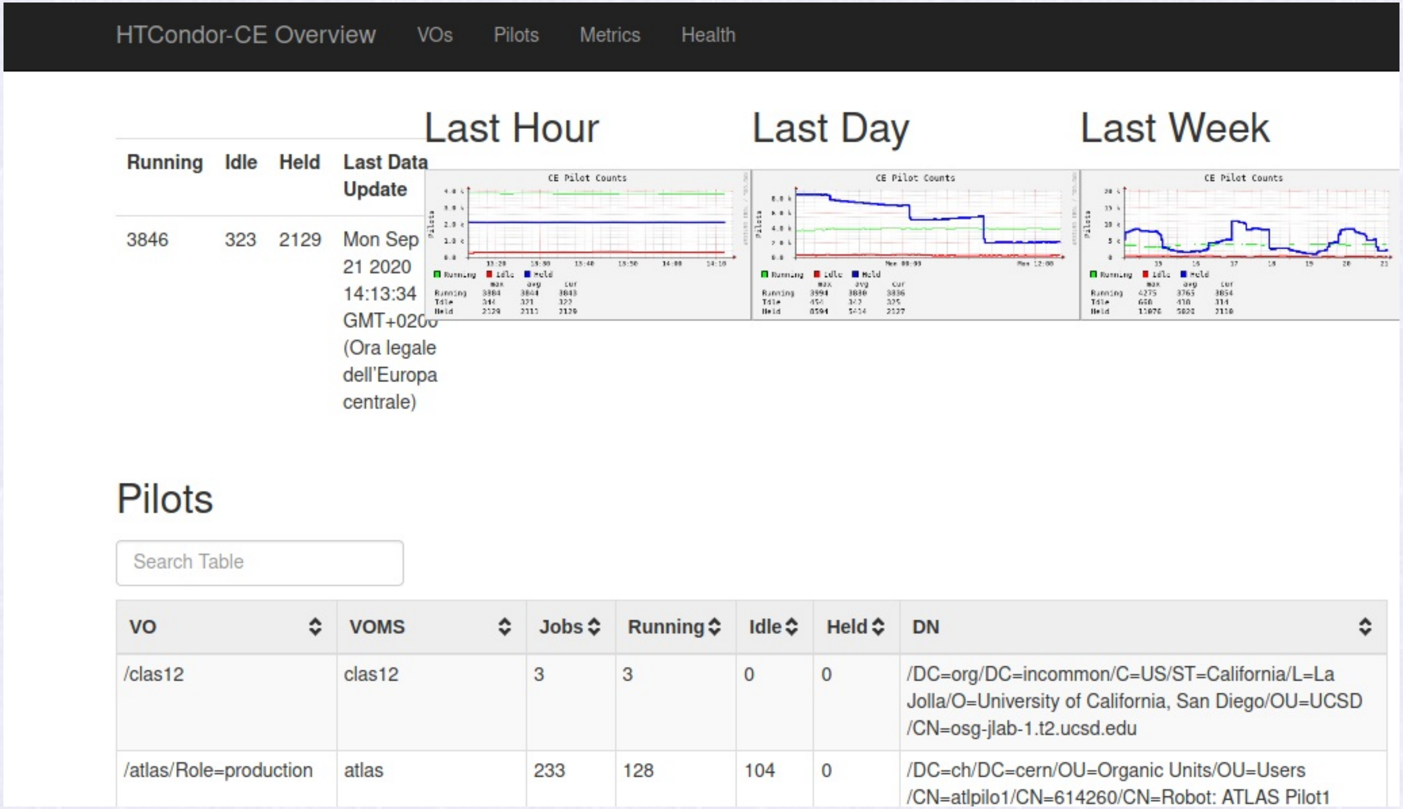
## Recent updates

A few more Italian Grid sites recently installed their HTC-CE instances. Just to name two:

- **INFN-BARI** Tier2, Using $2 \times$ HTC-CE with CMS for a few months now, adding ALICE soon. They use ARGUS for auth*zn; after rpm install they only had to adapt `/etc/condor-ce/config.d/99-local.conf`
- **INFN-LNL** Succesful tests with HTC-CE, production activity with CMS only expected to start soon

# Monitoring

HTCondor-CE comes with a small web tool (CEView) providing a simple interface for monitoring the activity of the CE.



| Running | Idle | Held | Last Data Update |
|---|---|---|---|
| 3846 | 323 | 2129 | Mon Sep 21 2020 14:13:34 GMT+0200 (Ora legale dell'Europa centrale) |

Last Hour

Last Day

Last Week

## Pilots

Search Table

| VO | VOMS | Jobs | Running | Idle | Held | DN |
|---|---|---|---|---|---|---|
| /clas12 | clas12 | 3 | 3 | 0 | 0 | /DC=org/DC=incommon/C=US/ST=California/L=La Jolla/O=University of California, San Diego/OU=UCSD /CN=osg-jlab-1.t2.ucsd.edu |
| /atlas/Role=production | atlas | 233 | 128 | 104 | 0 | /DC=ch/DC=cern/OU=Organic Units/OU=Users /CN=atlpilo1/CN=614260/CN=Robot: ATLAS Pilot1 |

# HTCondor-CE use cases

A couple of use cases useful to evaluate HTC-CE capabilities.

## 1. Accessing GPUs

as an exercise to get more confident with the CE I tried to configure Grid access to GPU resources. Turns out that this could be done quite easily.

## At client side:

In the condor submit file:

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

## At HTC-CE side:

In the HTCondor-CE JOB_ROUTER_ENTRIES:

```
[name = "condor_pool_atlas";
 TargetUniverse = 5;
 Requirements = target.x509UserProxyVOName =?= "atlas" &&\
                    (target.queue =?= "atlas_cuda");
 set_request_GPUs = 1;
 eval_set_WantGPU = true;
...]
```

## Accounting GPU usage

there are classAds in the job history file which can be tracked in the accounting usage records:

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

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

# GPU jobs activity (early Grid jobs, Apr 2019)

| 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 |

# Regular GPU activity from March 2020

```
acct=> SELECT <ugly sql query omitted>;

Month | N  |   WCT_h    |  username   |   exechosts
-------+----+------------+-------------+---------------
    3 |  9 | 370:27:29  | pilatlas030 | hpc-200-06-07
    4 | 19 | 869:17:56  | pilatlas030 | hpc-200-06-07
    5 | 34 | 866:19:18  | pilatlas030 | hpc-200-06-07
    6 | 37 | 1051:49:07 | pilatlas030 | hpc-200-06-07
    7 | 47 | 876:05:23  | pilatlas030 | hpc-200-06-07
    8 | 14 | 00:01:39   | virgousr1   | hpc-200-06-07
    8 |  3 | 00:00:33   | virgousr2   | hpc-200-06-07
    8 | 13 | 263:16:20  | pilatlas030 | hpc-200-06-07
    9 | 15 | 00:05:50   | virgousr1   | hpc-200-06-07
    9 |  9 | 96:31:19   | pilatlas030 | hpc-200-06-07
(10 rows)
```

## 2. HTCondor-CE as a frontend for HPC (Slurm)

CNAF was involved in a project to investigate “transparent Grid access” to HPC resources. We selected HTCondor-CE to interface with those resources (Jun 2019).

HTC-CE 3.2.2 (coupled with condor-8.8.2) was installed on a Slurm Submit Node (owned by CINECA) and made reachable to submitters at the standard 9619 TCP port. Completely manual setup, working on a “foreign host” without privileges.

- No WLCG environment (yaim, puppet, EGI repos,...)
- unprivileged only access: rpm install upon request, edit permission granted for a few config files, a few sudo commands for CE service management
- GSI (voms, argus, poolaccount) installed in the CE host itself

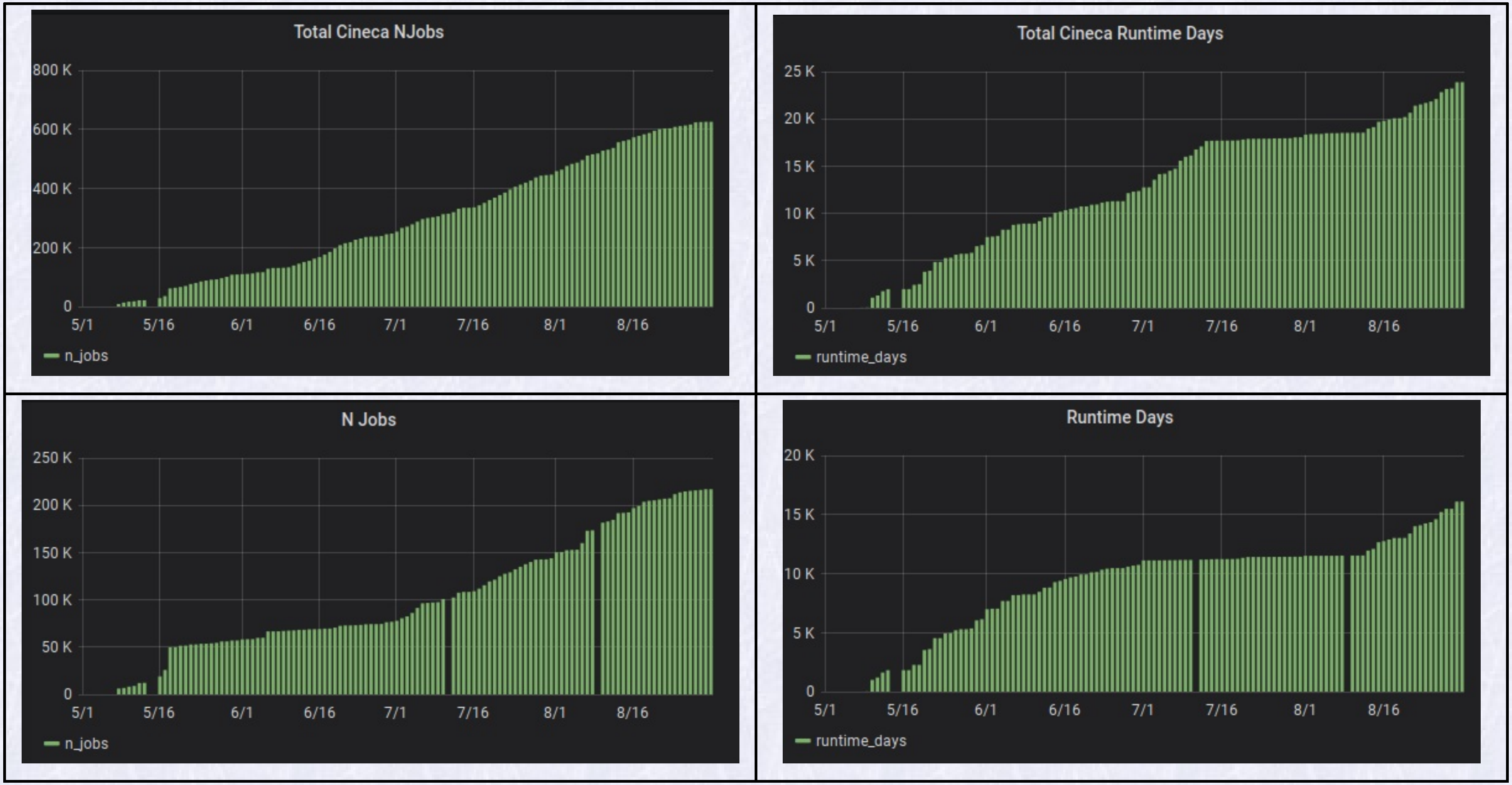


**Figure 1.** cumulative number of jobs and runtime for the whole HPC cluster and for CMS. see CHEP2019, Integration of CINECA HPC to CNAF

# WallClockTime accounting from the HTC-CE/Slurm

```
 [sdalpra0@rxyzl06 ~]$ condor_ce_history -cons '((GriddobStatus =?=
"COMPLETED") && (JobStartDate =!= undefined) && (RoutedToJobId =?= undefined))' -af:j x509UserProxyVOName '((CompletionDate - JobStartDate))'
'((CompletionDate - JobStartDate) * 68)/3600.0' Owner JobStartDate CompletionDate

255723.0 cms 170296 3216.7 a07cms02 1583488889 1583659185

253973.0 atlas 65 1.2 a07atl00 1583488841 1583488906

253856.0 atlas 70 1.3 a07atl00 1583488728 1583488798

252042.0 cms 155394 2935.2 a07cms02 1583333394 1583488788
```

## Use cases summary

We could succeed with two very different and non standard applications using an instrument with which I still was rather new. That confirms that HTC-CE has very “broad range” capabilities.

# HTC-CE deployment at CNAF

- We currently have $6\times$ CEs for INFN-T1 plus $1$ for INFN-LHCB-T2 (served by the same underlying HTCondor instance)

- we diversified the deployment of our CE instances: $(2+1)\times\mathrm{VM}$ oVirt, $2\times\mathrm{VM}$ VMware, $2\times$ bare metal (one having SSD for `/var`)

- New: $2\times\mathrm{HV}$ in oVirt share $2\times\mathrm{NVMe}$ disk via GlusterFS; the CEs in oVirt access their `SPOOL` from there. We plan to move some CEs there in the future.

- No overload observed so far under ordinary workload (very far from being overloaded, actually).

## Problems we have had

- condor_ce jobs remaining on hold ("HTCondor-CE held job due to expired user proxy") and never purged (fix: htcondor-ce-3.4.3).

  In the meanwhile we were manually purging such jobs:

  ```
  condor_ce_rm -cons '(JobStatus == 5 ) &&\
   (time() - x509UserProxyExpiration > 4 * 3600)'
  ```

- Submit Node (not a CE) exhausting space in /var dir

  **Reason**: jobs with input files too large (MAX_TRANSFER_INPUT_MB = 10). Transfer from Schedd to Exec Node fails, job put on hold in the Schedd.

  **Solution**: reject submission when input filesize too large:

  ```
  SUBMIT_REQUIREMENT_NAMES = InputFileTooLarge
  SUBMIT_REQUIREMENT_InputFileTooLarge = DiskUsage < 15000
  SUBMIT_REQUIREMENT_InputFileTooLarge_REASON = "input file too large"
  ```

## Conclusions

- Few months after replacing CREAM, The HTCondor-CE confirms itself as being our new "The good CE™"
- Official documentation for non OSG people now exists
- Can be seen as a "thin layer" on top of HTCondor (learn one, master two)
- Can easily deal with a wide range of heterogeneous scenarios
- most of the desired behaviours are to be obtained by configuring HTCondor services, at CE side and/or batch side
- JobRouting is the key mechanism to deal with when managing a HTC-CE. Need some practice to become confident with its configuration.