HTCondor-CE: Basics and Architecture

HTCondor Workshop 2019 - EU Joint Research Centre
Brian Lin
University of Wisconsin — Madison
The Pilot Overlay Model

User Submit

Pilot Factory

Compute Element

Local Batch System
The Pilot Overlay Model

User Submit

Pilot Factor y

Compute Element

Local Batch System
The Pilot Overlay Model

User Submit

Pilot Factory

Compute Element

Local Batch System
The Pilot Overlay Model

Pilot Overlay Model Diagram:
- User Submit
- Compute Element
- Local Batch System
- Pilot Factor

Diagram Illustrates:
- User interaction with the system
- Compute Element's role
- Interaction with Local Batch System
What is a CE?

- A compute element (CE) serves as the entry point to your local compute resources
  - Exposes a **remote API** for resource acquisition
  - Provides authentication and **authorization** of remote clients
  - Interacts with the **resource layer** (i.e. batch system)
- A CE is made up of a thin layer of **job gateway** software installed on a host that can submit to and manage jobs on your local batch system
- Designed to support the pilot job overlay model (i.e. resource provisioning requests) and is generally not intended for direct user submission
Compute Element Architecture

CE Host -> Job Gateway

Job Gateway -> Batch System Submit

Batch System Submit -> Local Batch System
HTCondor as a Job Gateway

HTCondor-CE is HTCondor configured as a job gateway

- Same HTCondor binaries, ClassAds, and configuration language to provide the **remote API**
- Relevant tools wrapped to use the HTCondor-CE configuration (e.g., `condor_ce_q`, `condor_ce_status`, etc.)
- Separate `condor-ce` service
HTCondor-CE + HTCondor Batch System

- Two sets of HTCondor daemons
  - Two sets of configuration:
    `/etc/condor-ce/config.d/` and `/etc/condor/config.d/`
  - Two sets of logs:
    `/var/log/condor-ce/` and `/var/log/condor/`

- Note the lack of the `condor_negotiator` for the CE set of daemons. HTCondor-CE doesn’t manage any worker nodes so it doesn’t need to do matchmaking!

```bash
# pstree
[...]
├── condor_master ├── condor_collector
│                  │── condor_negotiator
│                  │── condor_procd
│                  │── condor_schedd
│                  │── condor_shared_port
│                  │── condor_startd
│                  └── condor_job_router
│                        └── condor_procd
│                        └── condor_schedd
│                        └── condor_shared_port
└── condor_master ─── condor_collector
    └── condor_procd
    └── condor_schedd
    └── condor_shared_port

[...]
```
HTCondor as a Job Gateway

- By default, provides GSI authentication (authN) and uses HTCondor security for authorization
- HTCondor-CE 4 iterates on the default authentication model:
  - GSI authN is still supported but SciTokens is preferred if presented by a client (and you’re using a SciTokens-enabled HTCondor binaries)
  - HTCondor-CE daemons authenticate with each other using filesystem (i.e. Unix user) authN instead of GSI!
- Schedd AuditLog is used to record modifications to the job queue
- Payload jobs are also audited if incoming pilots report back to the HTCondor-CE’s collector daemon (e.g. GlideinWMS)
HTCondor as a Job Gateway

- Supports interaction with the following **resource layers**...
  - HTCondor batch systems directly
  - Slurm, PBS Pro/Torque, SGE, and LSF batch systems
  - Also with all of the above via SSH
- Non-HTCondor batch systems and SSH submission are supported via the HTCondor GridManager daemon and the Batch ASCII Language Helper Protocol (BLAHP)
  - Takes the routed job and further transforms it into your local batch’s JDL
  - Specific Job ClassAd attributes result in batch system specific directives, e.g. the Queue attribute results in **#SBATCH --partition ...** for Slurm
  - Queries the local batch job to pass along state updates back along the job chain
Job Router Daemon

- The Job Router is responsible for taking a job, creating a copy, and changing the copy according to a set of rules
  - When running an HTCondor batch system, the copy is inserted directly into the site batch schedd. Otherwise, the copy is inserted back into the CE schedd
  - Each chain of rules is called a “job route” and is defined by a ClassAd
  - Job routes reflect a site’s policy
- Once the copy has been created, attribute changes and state changes are propagated between the source and destination jobs
- Can be configured to match jobs to routes using round-robin or first-match (the default) strategies
HTCondor-CE Daemons

- systemctl start condor-ce
- service condor-ce start
- condor_ce_on

Diagram:
- Master
- Schedd
- Collector
- Job Router
HTCondor-CE + HTCondor Batch System

1. Grid Job
2. Routed Job
3. HTCondor Negotiation
HTCondor-CE + Non-HTCondor Batch System

- Since there is no local batch system schedd, jobs are routed back into the CE schedd as “Grid Universe” jobs
- Grid universe jobs spawn a Gridmanager daemon per user with log files: /var/log/condor-ce/GridmanagerLog.<user>
- Requires a shared filesystem across the cluster for pilot job file transfers
HTCondor-CE + Non-HTCondor Batch System

1. Grid Job
2. Routed Job
3. Start GridManager
4. qsub, sbatch, etc.

CE Host -> CE Schedd -> Job Router

Grid Manager
HTCondor-CE + HTCondor + Non-HTCondor

1. Grid Job
2. Routed Job
3a. HTCondor Negotation
3b. Start GridManager
4. qsub, sbatch, etc.
HTCondor-CE + SSH

- Using BOSCO ([https://osg-bosco.github.io/docs/](https://osg-bosco.github.io/docs/)), HTCondor-CE can be configured to submit jobs over SSH
  - Requires SSH key-based access to an account on a node that can submit and manage jobs on the local batch system
  - Requires shared home directories across the cluster for pilot job file transfer
- The Open Science Grid (OSG) uses HTCondor-CE over SSH to offer HTCondor-CE as a Service (a.k.a. Hosted CE) for small sites
- Can support up to ~10k jobs concurrently
HTCondor-CE + SSH

1. Grid Job
2. Routed Job
3. Start Gridmanager
4. SSH
5. qsub, sbatch, etc.

CE Host
CE Schedd
Job Router
Gridmanager
Submit/Head Node

EC-JRC - HTCondor-CE: Basics and Architecture
HTCondor-CE Requirements

- Open port (TCP) 9619
- Shared filesystem for non-HTCondor batch systems for pilot job file transfer
- CA certificates and CRLs installed in `/etc/grid-security/certificates/`
- VO information installed in `/etc/grid-security/vomsdir/`
- Ensure mapped users exist on the CE (and across the cluster)
- Minimal hardware requirements
  - Handful of cores
  - HTCondor backends should plan on ~½ MB RAM per job
  - Expecting high rates of jobs? HTCondor-CE `SPOOL` dir should live on an SSD
    - Default: `/var/lib/condor-ce/spool` (condor_ce_config_val -v SP00L)
- For example, our Hosted CEs run on 2 vCPUs and 2GB RAM
HTCondor-CE Information Systems

- HTCondor-CE offers a simple information service using the built-in HTCondor View feature to report useful grid information
  - Contact information (hostname/port)
  - Access policy (authorized virtual organizations)
  - What resources can be accessed?
  - Debugging info (site batch system, site name, versions) for humans

- Each HTCondor-CE in a grid can be configured to report information to one or more HTCondor-CE Central Collectors
  - Under the hood, CE Schedd attributes are published to the Central Collector(s)
  - There are still some OSG smells here: tooling and default status output keys off of OSG_* attributes
## HTCondor-CE Information Systems

```bash
$ condor_ce_info_status
```

<table>
<thead>
<tr>
<th>Name</th>
<th>CPUs</th>
<th>Memory</th>
<th>MaxWallTime</th>
<th>AllowedVOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>OU_OSCER_ATLAS_2650</td>
<td>20</td>
<td>32768</td>
<td>4320</td>
<td>atlas, dosar</td>
</tr>
<tr>
<td>OU_OSCER_ATLAS_2670</td>
<td>24</td>
<td>65536</td>
<td>4320</td>
<td>atlas, dosar</td>
</tr>
<tr>
<td>R510</td>
<td>8</td>
<td>24576</td>
<td>1440</td>
<td>osg, cms</td>
</tr>
<tr>
<td>R730xd</td>
<td>12</td>
<td>32768</td>
<td>1440</td>
<td>osg, cms</td>
</tr>
<tr>
<td>OUHEP_ITB_1</td>
<td>4</td>
<td>6144</td>
<td>1440</td>
<td>atlas, dosar</td>
</tr>
<tr>
<td>CancerComputer_MinneUE</td>
<td>12</td>
<td>24576</td>
<td>1440</td>
<td>osg, sbgrid, mis</td>
</tr>
<tr>
<td>USCMS-FNAL-WC1</td>
<td>8</td>
<td>16384</td>
<td>2850</td>
<td>&quot;cms&quot;</td>
</tr>
<tr>
<td>AGLT2-Dell1</td>
<td>8</td>
<td>16000</td>
<td>4300</td>
<td>atlas</td>
</tr>
<tr>
<td>AGLT2-Dell2</td>
<td>8</td>
<td>16000</td>
<td>4300</td>
<td>atlas</td>
</tr>
</tbody>
</table>

[...]
### HTCondor-CE Information Systems

```
$ condor_status -schedd -pool collector.opensciencegrid.org:9619
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Machine</th>
<th>RunningJobs</th>
<th>IdleJobs</th>
<th>HeldJobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE01.CMSAF.MIT.EDU</td>
<td>CE01.CMSAF.MIT.EDU</td>
<td>412</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>CE02.CMSAF.MIT.EDU</td>
<td>CE02.CMSAF.MIT.EDU</td>
<td>669</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CE03.CMSAF.MIT.EDU</td>
<td>CE03.CMSAF.MIT.EDU</td>
<td>1532</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>atlas-ce.bu.edu</td>
<td>atlas-ce.bu.edu</td>
<td>2568</td>
<td>1627</td>
<td></td>
</tr>
<tr>
<td>bgk01.sdcc.bnl.gov</td>
<td>bgk01.sdcc.bnl.gov</td>
<td>5</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>bonner06.rice.edu</td>
<td>bonner06.rice.edu</td>
<td>7</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>brown-osg.rcac.purdue.edu</td>
<td>brown-osg.rcac.purdue.edu</td>
<td>2</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>ce01.brazos.tamu.edu</td>
<td>ce01.brazos.tamu.edu</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>ce01.ific.uv.es</td>
<td>ce01.ific.uv.es</td>
<td>1012</td>
<td>265</td>
<td>0</td>
</tr>
<tr>
<td>ce1-vanderbilt.sites.opensciencegrid.org</td>
<td>ce1-vanderbilt.sites.opensciencegrid.org</td>
<td>310</td>
<td>194</td>
<td>4</td>
</tr>
<tr>
<td>ce2-vanderbilt.sites.opensciencegrid.org</td>
<td>ce2-vanderbilt.sites.opensciencegrid.org</td>
<td>1162</td>
<td>357</td>
<td>0</td>
</tr>
<tr>
<td>cit-gatekeeper.ultralight.org</td>
<td>cit-gatekeeper.ultralight.org</td>
<td>76</td>
<td>33</td>
<td>2</td>
</tr>
<tr>
<td>cit-gatekeeper2.ultralight.org</td>
<td>cit-gatekeeper2.ultralight.org</td>
<td>96</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>cit-gatekeeper3.ultralight.org</td>
<td>cit-gatekeeper3.ultralight.org</td>
<td>98</td>
<td>34</td>
<td>3</td>
</tr>
</tbody>
</table>
```

[...]

September 26, 2019  
EC-JRC - HTCondor-CE: Basics and Architecture
HTCondor-CE Information Systems

$ condor_status -schedd -pool collector.opensciencegrid.org:9619 -json
[
{
  "AddressV1": "[[ p="primary"; a="18.12.1.31"; port=9619; n="Internet"; spid="323298_41ac_3"; noUDP=true; ], [ p="IPv4"; a="18.12.1.31"; port=9619; n="Internet"; spid="323298_41ac_3"; noUDP=true; ]],
  "AuthenticatedIdentity": "ce01.cmsaf.mit.edu@daemon.opensciencegrid.org",
  "AuthenticationMethod": "GSI",
  "Autoclusters": 0,
  "CollectorHost": "CE01.CMSAF.MIT.EDU:9619",
  "CondorPlatform": "$CondorPlatform: X86_64-CentOS_7.5 $",
  "CurbMatchmaking": false,
  "DaemonCoreDutyCycle": 0.04549036158372677,
  "DaemonStartTime": 1569321031,
  "DetectedCpus": 16,
  "DetectedMemory": 24094,
  "FileTransferDownloadBytes": 0.0,
  [...]
}
HTCondor-CE Information Systems

Data from 109 CEs reporting to the OSG Central Collector

- HTCondor: 65.1%
- Slurm: 22.0%
- PBS/Torque: 11.9%
- SGE: 0.9%
Why Use HTCondor-CE

- If you are using HTCondor for batch:
  - One less software provider - same thing all the way down the stack.
  - HTCondor has an extensive feature set - easy to take advantage of it (i.e., Docker universe).
- Regardless, a few advantages:
  - Can scale well (up to at least 16k jobs; maybe higher).
  - Declarative ClassAd-based language.
- But disadvantages exist:
  - Non-HTCondor backends are finicky outside PBS and Slurm.
  - Declarative ClassAd-based language.
Getting Started with HTCondor-CE

- Available as RPMs via HTCondor (and OSG) Yum repositories
- Start installation with documentation available via htcondor-ce.org

The HTCondor-CE software is a job gateway based on HTCondor for Compute Elements (CE) belonging to a computing grid (e.g. European Grid Infrastructure, Open Science Grid). As such, HTCondor-CE serves as an entry point for incoming grid jobs — it handles authorization and delegation of jobs to a grid site's local batch system.

Supported batch systems include:

- Grid Engine
- HTCondor
- LSF
- PBS/Torque
- Slurm