Introducing the HTCondor-CE

CHEP 2015

Presented by Edgar Fajardo

Introduction

- In summer 2012, OSG performed an internal review of major software components, looking for strategic weaknesses.
  - One highlighted area of concern was the gateway software (GRAM).
    - Decided to diversify; internally evaluated CREAM, HTCondor-CE, and staying with GRAM. Decision was to use HTCondor long-term.
- During 2013, major integration and beta testing was done.
- In 2014, large scale testing and transition started at largest, most complex sites.

What is HTCondor-CE?

- A **Compute Element** (CE) is OSG's entry point to a site's local cpu resources

- The core of a CE is the **gateway**, which provides:
  - **Routing**: Translation of a resource request to a batch system job (includes initial job submission, state tracking, and cleanup).
  - **Remote management**: A protocol for managing resource requests from remote clients.
  - Authentication and **Authorization**: Determining identity of the client and what actions they may perform.

- **NOTE**: In OSG, we do not submit jobs to CEs. We submit *resource requests*. At most CEs, these become jobs in the batch system.

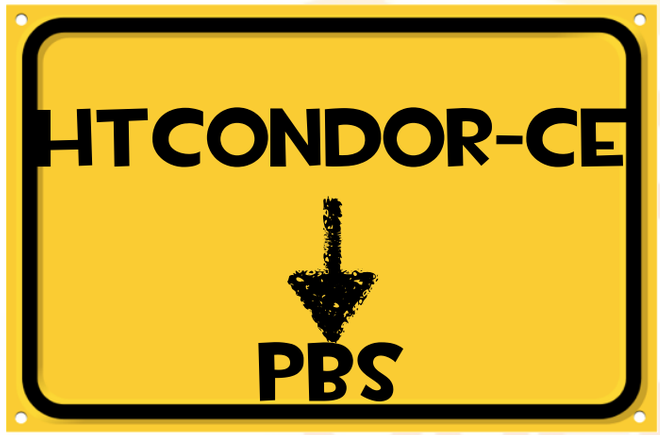
What is HTCondor-CE?

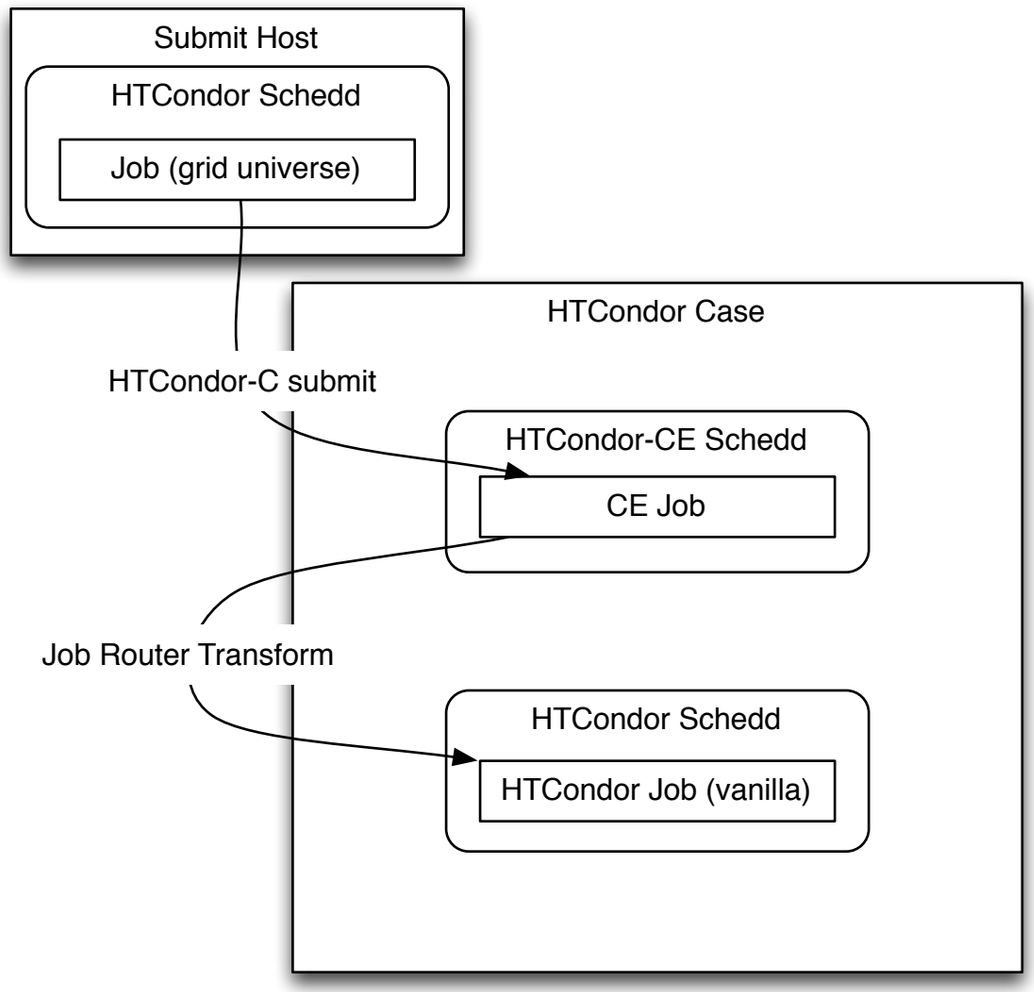
- The HTCondor-CE is a gateway implemented using a special configuration of the **HTCondor** software. Everything is:
  - HTCondor daemons.
  - HTCondor configurations.
  - HTCondor plugins.

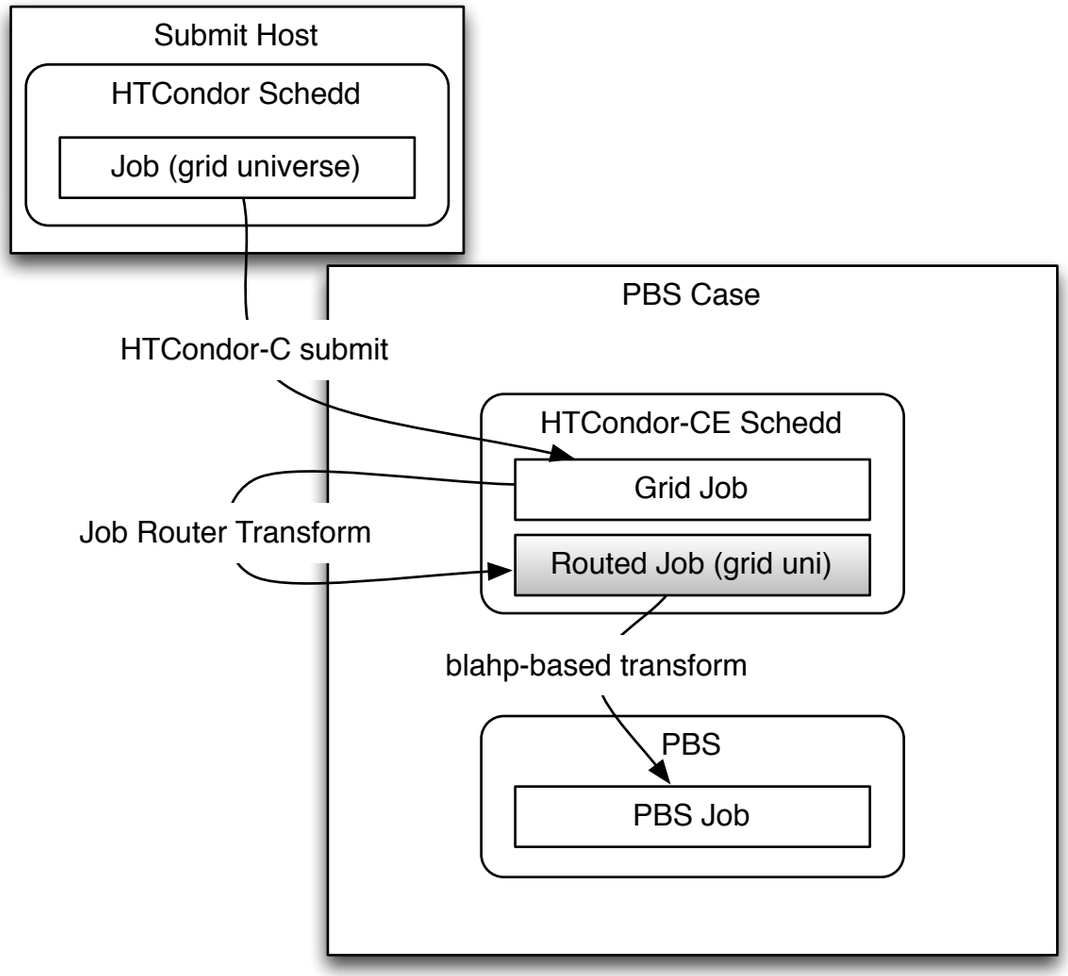
HTCondor
High Throughput Computing

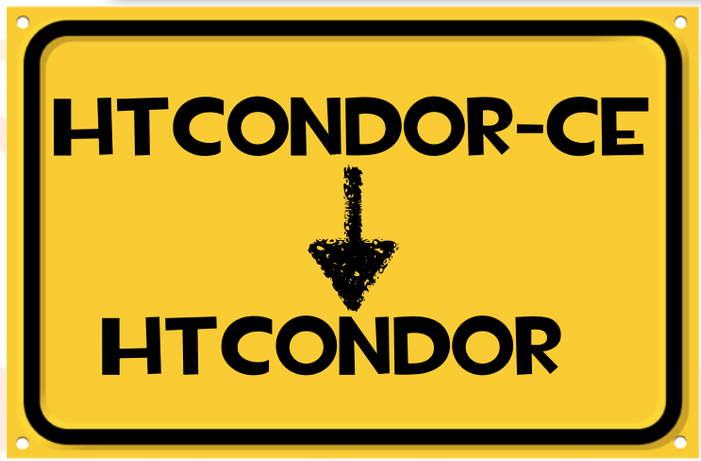
- Authentication is done with **GSI**; authorization is done with **LCMAPS**.
- Remote submit protocol is **HTCondor-C**.
- The 'heart' of customizing jobs is the **JobRouter**, a declarative transform language.
  - Interface with local batch system is **blahp** / Condor-G.

HTCondor-CE in a slide









Transformation Details

- **Job Router**: Transform requests to jobs (localize at CE)
- **BLAHP**:
  - Submit jobs to non-HTCondor batch systems (PBS, SGE, SLURM, etc.)
  - blahp is the executable which then calls, for example, qstat / qsub / qdel.
  - blahp has another layer of customization if, for example, you need to tweak qsub arguments. Most useful things can be done via the JobRouter transform.

Job Route Example

```
JOB_ROUTER_ENTRIES = \
   [ \
     GridResource = "batch pbs"; \
     TargetUniverse = 9; \
     name = "Local_PBS_cms"; \
     default_queue = "cms"; \
     Requirements = target.x509UserProxyVOName =?= "cms"; \
   ] \
   [ \
     GridResource = "batch pbs"; \
     TargetUniverse = 9; \
     name = "Local_PBS_other"; \
     default_queue = "other"; \
     Requirements = target.x509UserProxyVOName =!= "cms"; \
   ]
```

More details/recipes for the routes:
https://twiki.grid.iu.edu/bin/view/Documentation/Release3/JobRouterRecipes

Hate declarative languages? Script-based callout also available!

Why HTCondor-CE? STRATEGIC

- **Local expertise** - HTCondor developers are within “shouting distance” from OSG software team.

- **Unify** submission, CE, and opportunistic computing platforms - Condor-G, Condor-CE, and BOSCO - **onto a single software stack**.
  - At least one HTCondor piece - the **blahp** - overlaps with CREAM CE.

- **Minimize** OSG effort needed. HTCondor-CE is a *configuration* of HTCondor: all non-configuration bugs are a problem for the HTCondor team.

- **No new external dependency**. We depend on the HTCondor team regardless of the CE; with any other solution, we have an additional external dependency.

Why HTCondor-CE?
TECHNICAL

	HTCondor-CE	GRAM
Best max running jobs	16k*	10k
Network Port usage (per running job	2	4

- Management: CE layer is visible and available for interaction. “condor_*” toolset works with CE.

- Packaging: HTCondor provides many customization hooks - and doesn’t overwrite changes on upgrade!

*Limit in the test is the batch system memory, which maxes out at 16k.

CE layer visible!

bbockelm — root@red:~ — ssh — 92×24

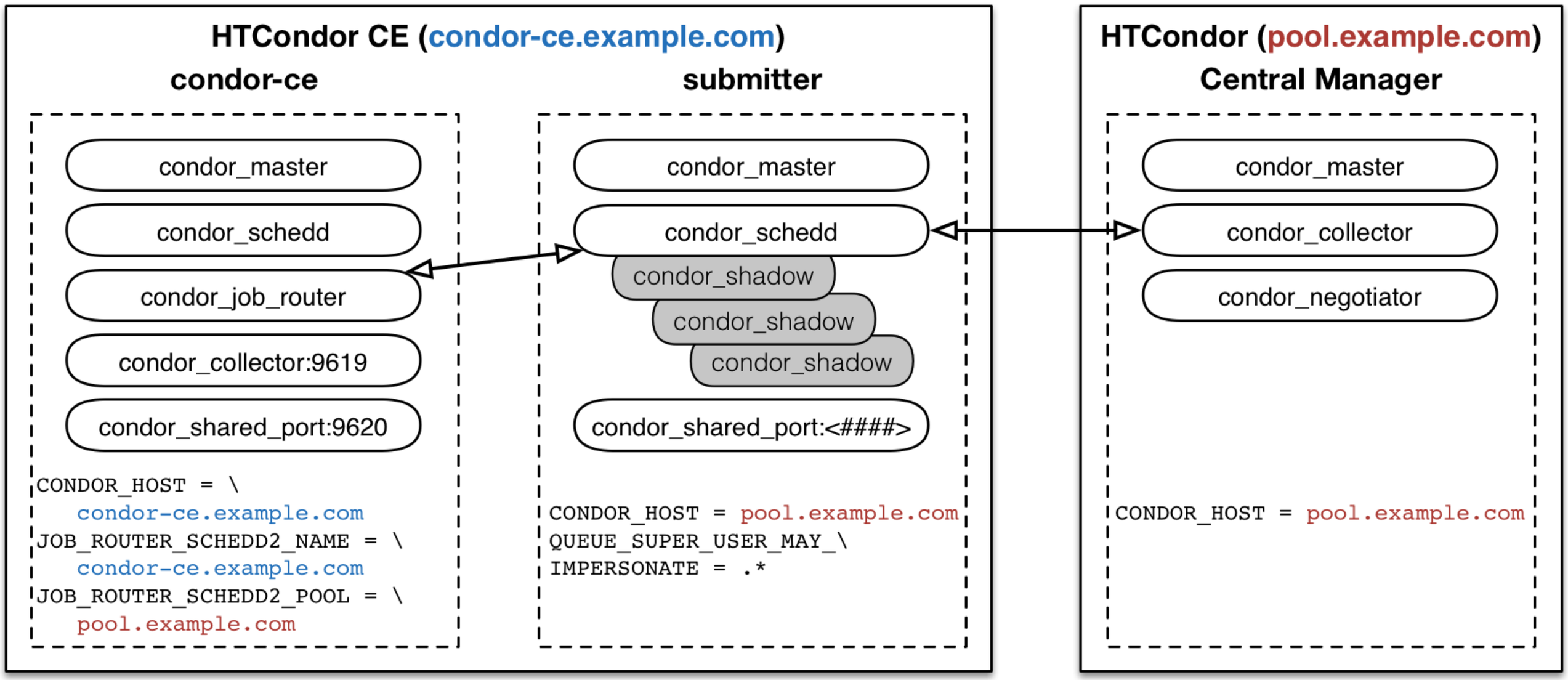
```
360622.0   uscmsPool2557   4/9  07:23   0+00:13:02 R  0   0.0  glidein_startup.sh
360623.0   uscmsPool2553   4/9  07:23   0+00:00:00 I  0   0.0  glidein_startup.sh
360624.0   uscmsPool2556   4/9  07:29   0+00:00:00 I  0   0.0  glidein_startup.sh
360625.0   uscmsPool2558   4/9  07:30   0+00:00:00 I  0   0.0  glidein_startup.sh
360626.0   uscmsPool2558   4/9  07:30   0+00:00:00 I  0   0.0  glidein_startup.sh
360627.0   uscmsPool2555   4/9  07:30   0+00:00:00 I  0   0.0  glidein_startup.sh
360628.0   uscmsPool2557   4/9  07:34   0+00:00:00 I  0   0.0  glidein_startup.sh
360629.0   uscmsPool2557   4/9  07:34   0+00:00:00 I  0   0.0  glidein_startup.sh
360630.0   uscmsPool2557   4/9  07:34   0+00:00:00 I  0   0.0  glidein_startup.sh
360631.0   uscmsPool2557   4/9  07:34   0+00:00:00 I  0   0.0  glidein_startup.sh
360632.0   uscmsPool2557   4/9  07:34   0+00:00:00 I  0   0.0  glidein_startup.sh
360633.0   uscmsPool2557   4/9  07:37   0+00:00:00 I  0   0.0  glidein_startup.sh
360634.0   uscmsPool2557   4/9  07:37   0+00:00:00 I  0   0.0  glidein_startup.sh
360635.0   uscmsPool2557   4/9  07:37   0+00:00:00 I  0   0.0  glidein_startup.sh
360636.0   uscmsPool2557   4/9  07:37   0+00:00:00 I  0   0.0  glidein_startup.sh
360637.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh
360638.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh
360639.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh
360640.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh
360641.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh
360642.0   uscmsPool2557   4/9  07:41   0+00:00:00 H  0   0.0  glidein_startup.sh

3090 jobs; 10 completed, 2 removed, 1110 idle, 1481 running, 487 held, 0 suspended
[root@red ~]# condor_ce_q | tail -n 24
```

Can query the CE itself with condor_ce_q!

Provides visibility to CE status information and errors.

Running Daemons



~~Information Service~~

- Prior information services in OSG were not useful for provisioning. Other services describe the state and configuration of batch queues. We want a *provisioning information* service that describes:
  - **Resource types**. "I can allocate you a 8-core resource with 16 GB of RAM."
    - Note we don't try to enumerate the number of resources of each type - utilized or available.
  - **How to access resources**. "To get at resource type A, set attribute foo='bar' in request."
  - Want to include both standard attributes (Cores, RAM, Disk) and VO-custom ones (IO intensity, VO queue name, job type).
- To meet these requirements, we needed to start from scratch.

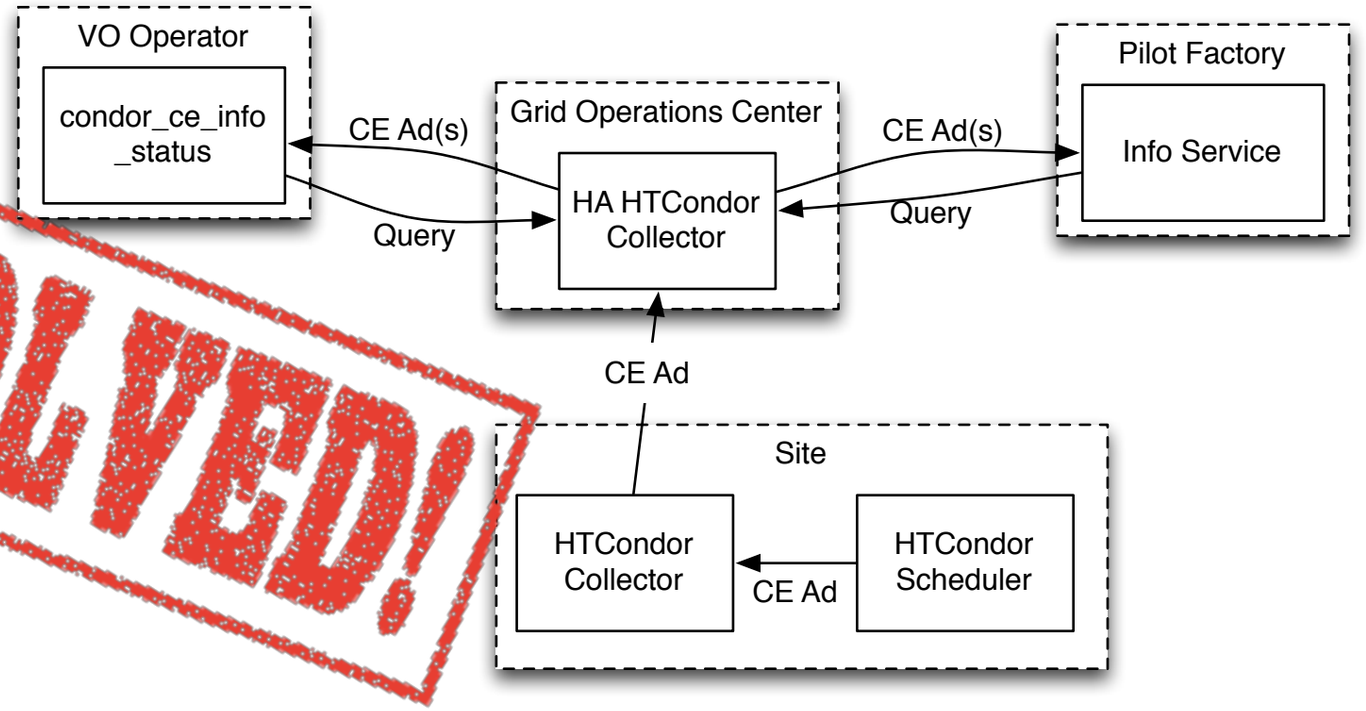
HTCondor-CE Collect

- Basic idea: reuse HTCondor components to provide the info service.

- Describe provisioning information in a single HTCondor ClassAd.

- Setup a condor_collector centrally. Use built-in forwarding capabilities from site collector to central one.

- Use standard condor querying tools to interact with central collector.



- For more information, see https://indico.fnal.gov/getFile.py/access?contribId=19&sessionId=8&resId=0&materialId=slides&confId=8580

Upcoming Activity

- A few strategic directions:
  - Flesh out the blahp support for LSF & SGE when used via HTCondor.
    - Getting blahp to work with LSF has been an **epic battle**.
  - Make (HTCondor-CE) - (HTCondor) = smaller.
    - Goal is always to keep the HTCondor-CE “config-only”. Still Mostly True.
  - Take better advantage of existing HTCondor features; get HTCondor team to implement new ones (**Docker universe**).
  - Continue refinements - especially in terms of ease-of-configuration and ease-of-customization.

HTCondor-CE and Docker Universe

- JobRouter allows you to inject arbitrary attributes into the routed job for HTCondor sites.
    - This allows admins to control which HTCondor features or options are turned on for a given user's job.
- At Nebraska, we've been very interested in containerization efforts; one observation are chroots are hard to create!
    - Docker provides similar container features *but* provides tooling for easy-to-create environments.
- Next HTCondor release will allow the site to launch a Docker container.
    - Sites can easily modify HTCondor-CE routing to have pilots launched inside Docker containers.
    - We can then share a single Docker image for worker nodes. "One less thing" for sites to maintain.

HTCondor-CE (Local) Collector

- We've always wanted more information about payload jobs.
  - Who's running? What are they running? Are they using CPU efficiently?

- In the next HTCondor-CE release, the CE will allow pilots to send startd ads (representing the payload jobs). The CE admin can **view the payload activity with condor_status**.

- In the next gWMS release, the pilot will send these ads automatically.

Parting Thoughts

- HTCondor-CE is a fresh approach on gateway technology - focusing solely on **resource provisioning** - built on top of the **foundation of HTCondor**.
  - Reason for switching include both organizational and technical.
  - Tries to re-envision both the concept of a gateway and how information services interact.

- Transition has been ongoing within OSG for the larger sites. In 2015, we will start tackling smaller ones.

- Moving forward, we would like the HTCondor-CE to shrink in size (becoming more and more “just HTCondor”) and continue to benefit from new features in the base software.