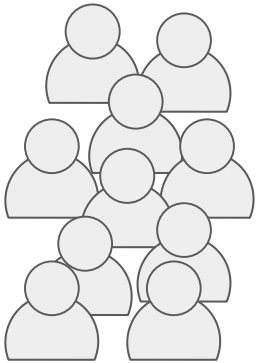
HTCondor-CE: Basics and Architecture

HTCondor Workshop 2019 - EU Joint Research Centre

Brian Lin
University of Wisconsin — Madison

The Pilot Overlay Model

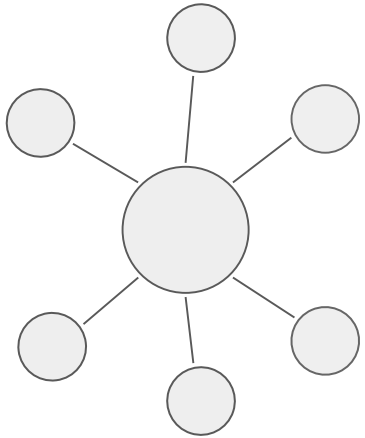


User Submit

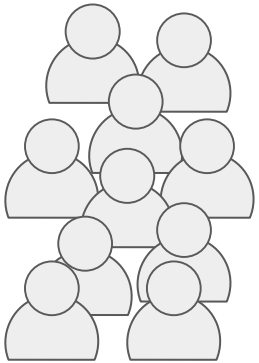
Pilot Factory

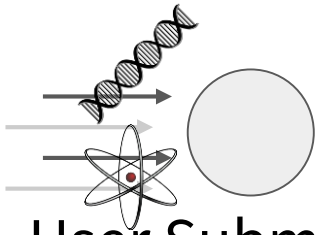
Compute Element

Local Batch System



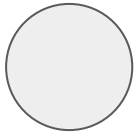
The Pilot Overlay Model



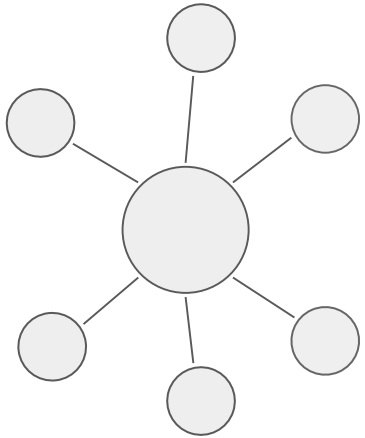


User Submit

Compute Element

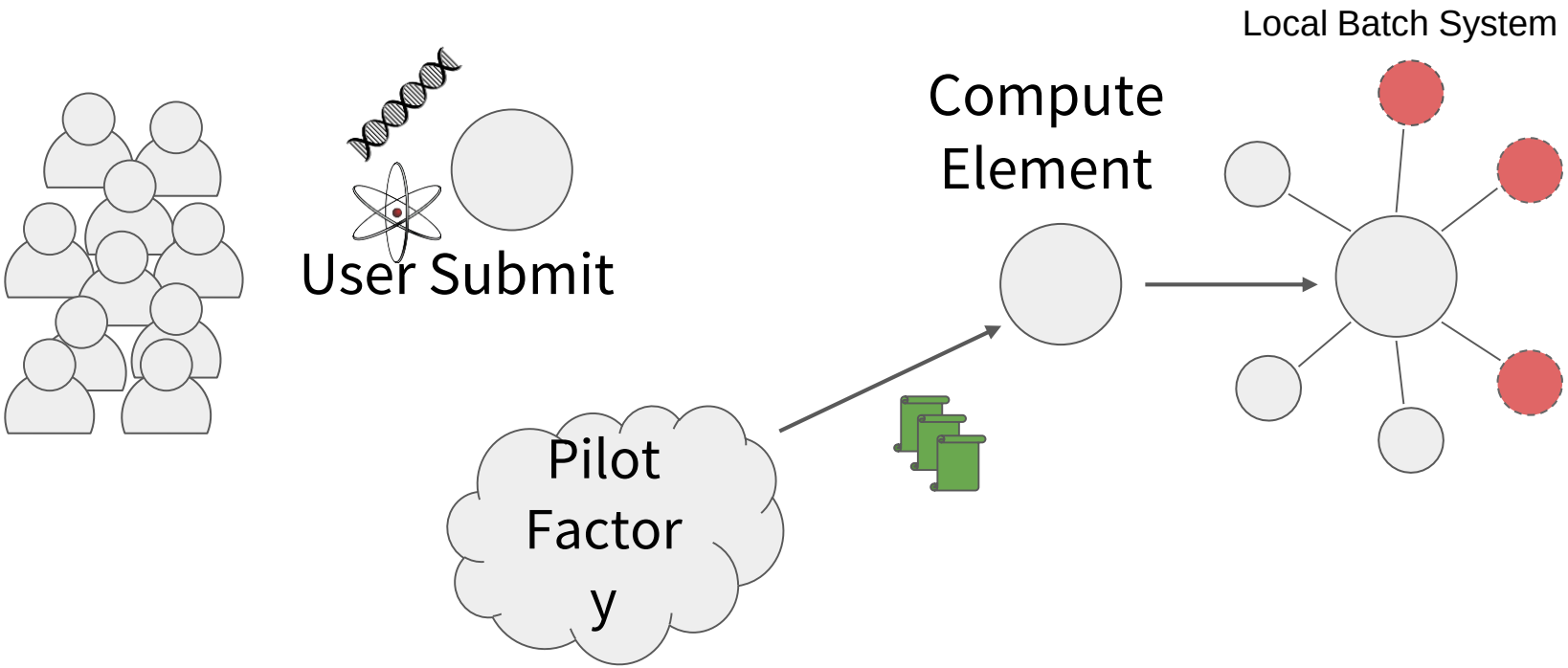


Local Batch System

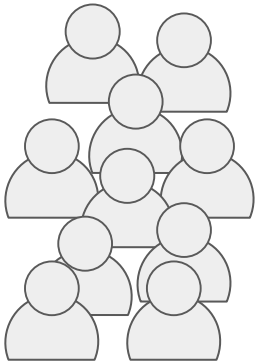


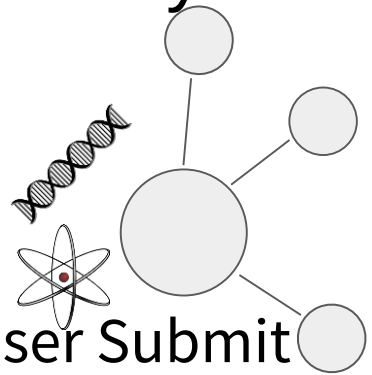
Pilot Factory

The Pilot Overlay Model



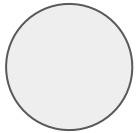
The Pilot Overlay Model



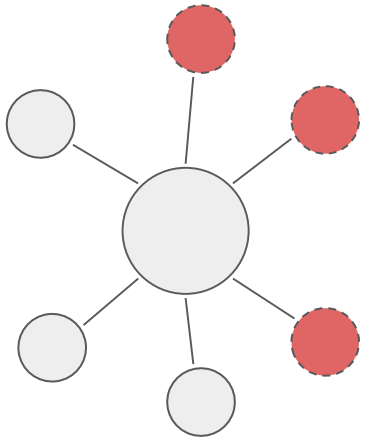


User Submit

Compute Element



Local Batch System

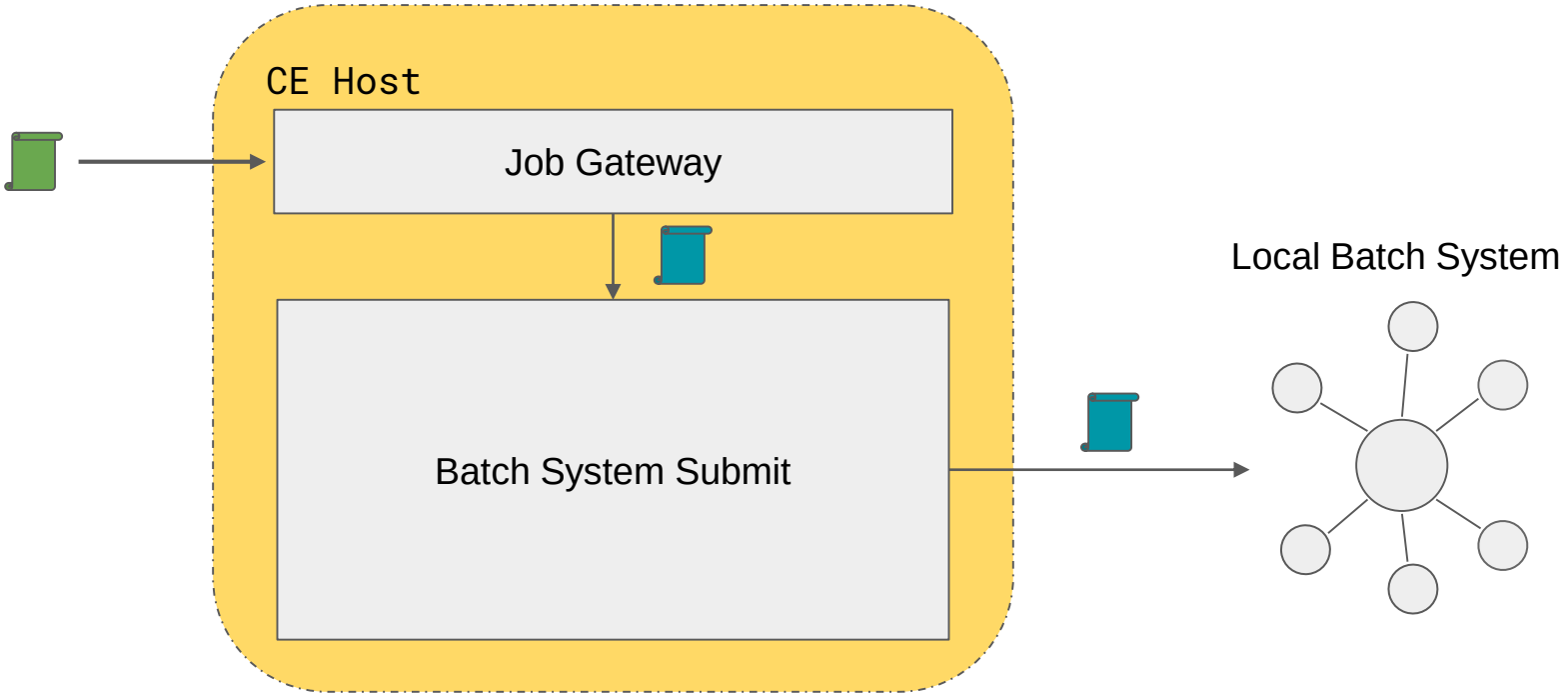


Pilot Factory

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



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!

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

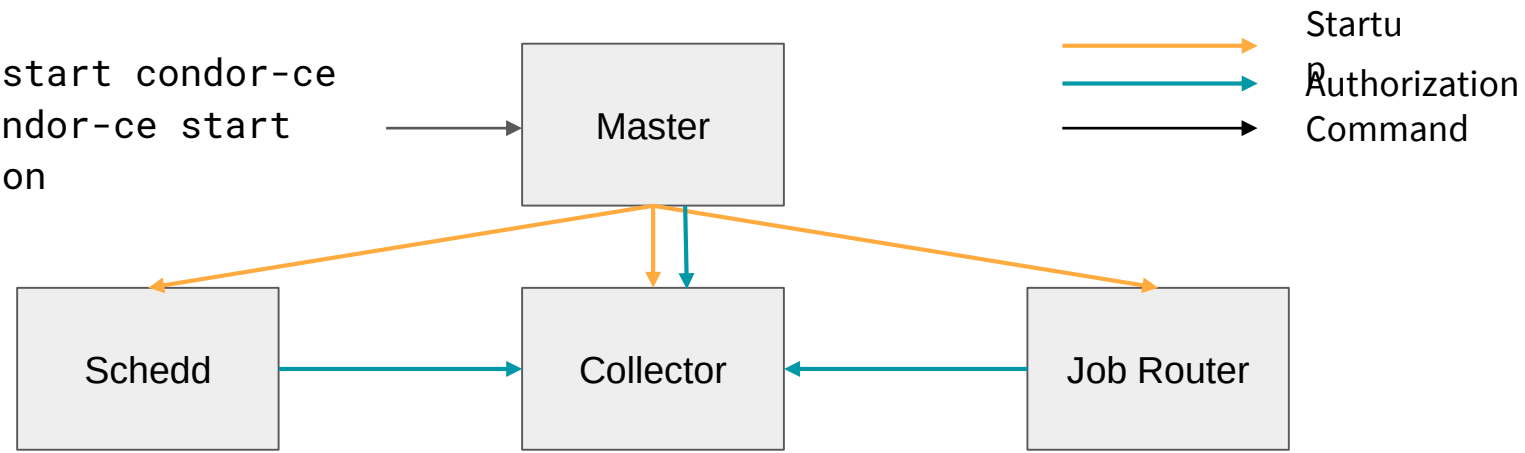
- Supports interaction with the following **resource layers**...
  - HTCondor batch systems directly
  - Slurm, PBS Pro/Torque, SGE, and LSF batch systems
  - Also with all of the above via SSH
- Non-HTCondor batch systems and SSH submission are supported via the HTCondor GridManager daemon and the Batch 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

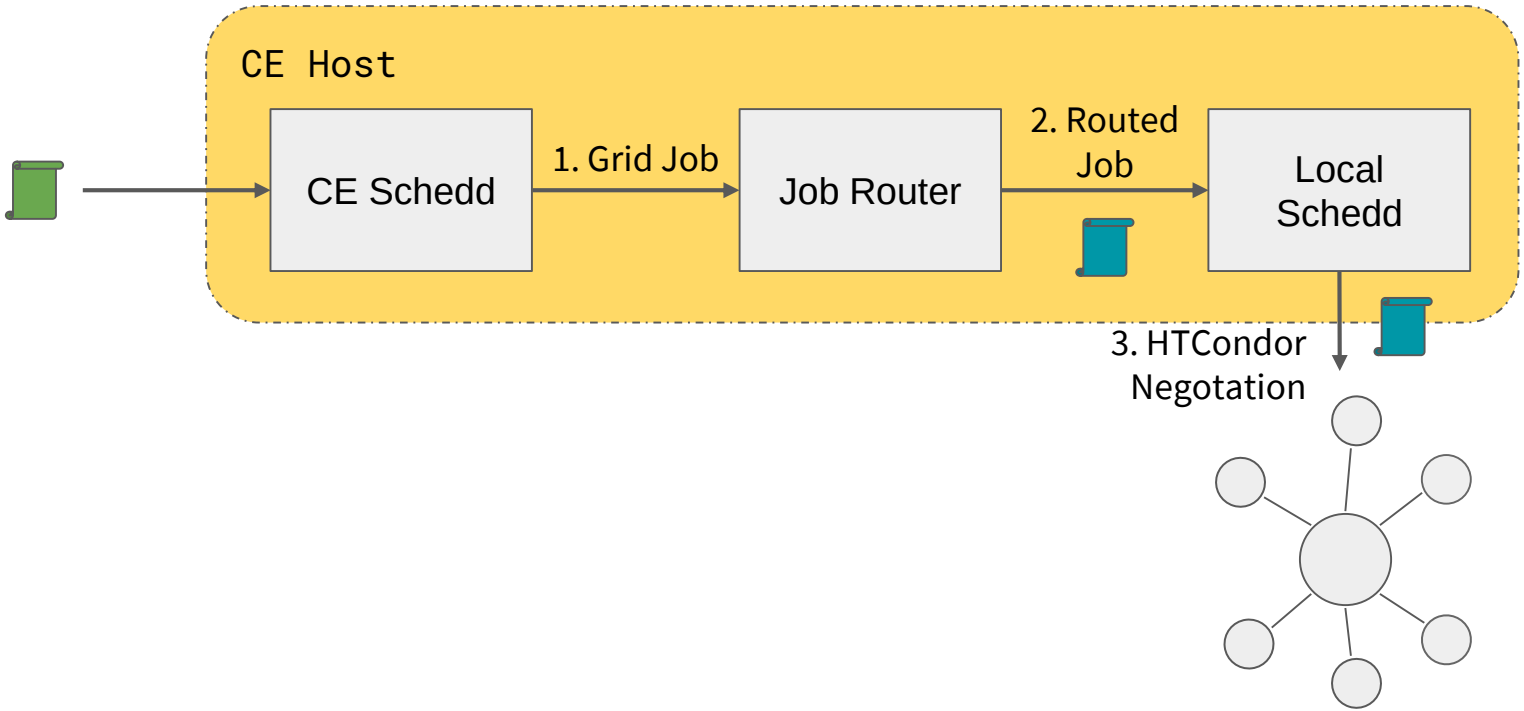
- The Job Router is responsible for taking a job, creating a copy, and changing the copy according to a set of rules
  - When running an HTCondor batch system, the copy is inserted directly into the site batch schedd. Otherwise, the copy is inserted back into the CE schedd
  - Each chain of rules is 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
```



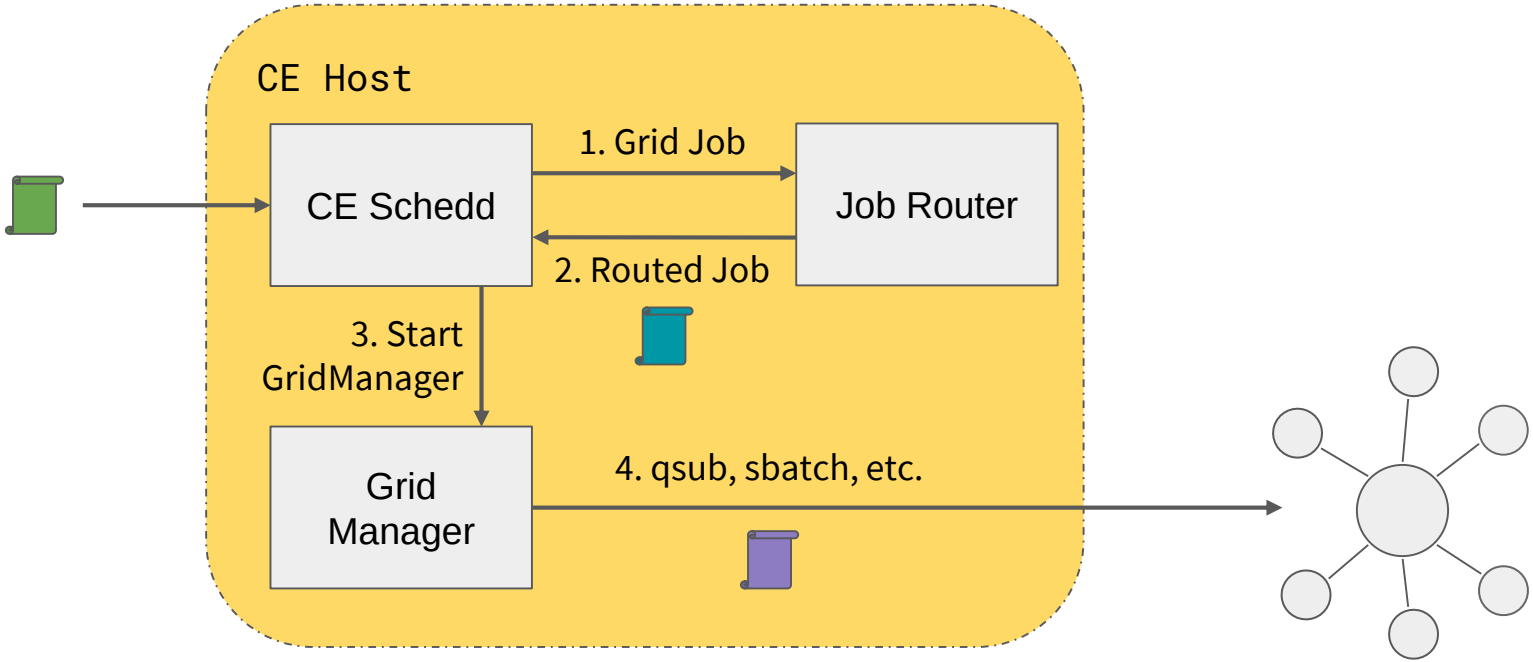
HTCondor-CE + HTCondor Batch System



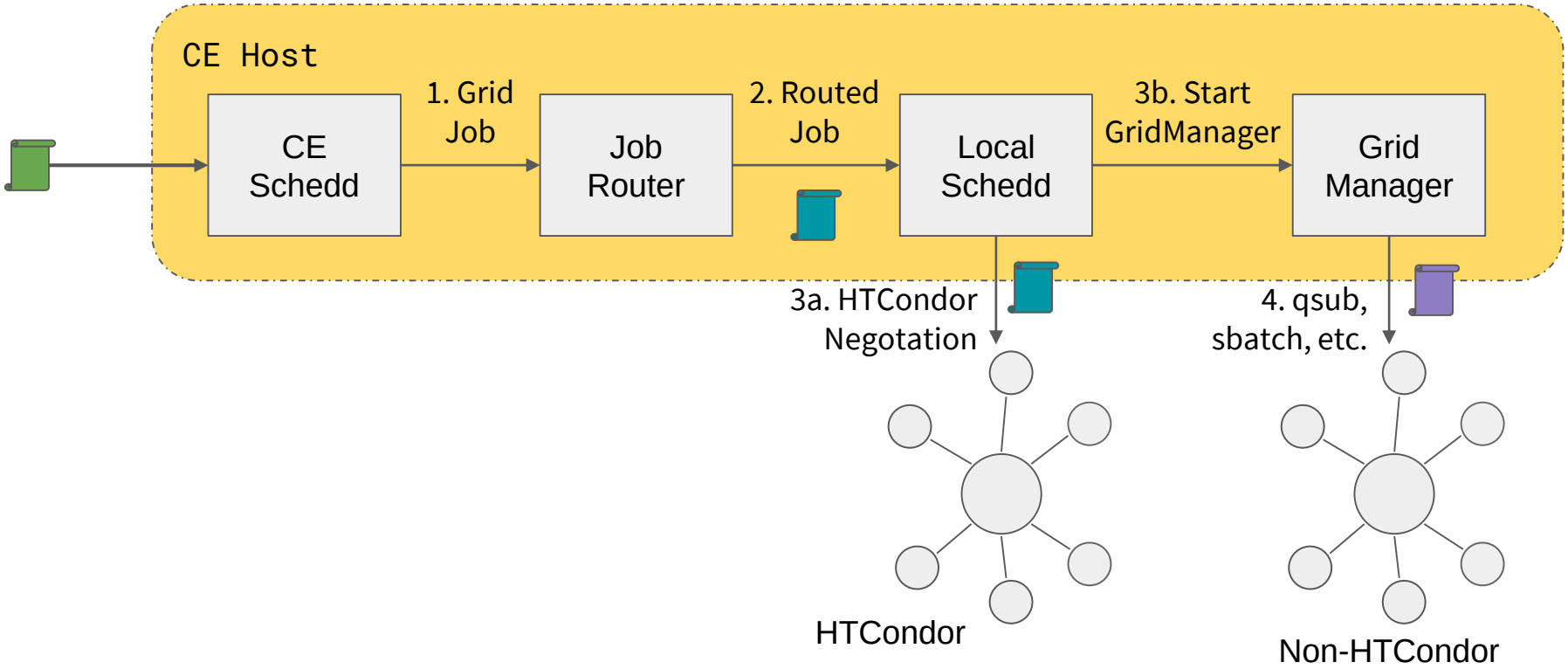
HTCondor-CE + Non-HTCondor Batch System

- 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



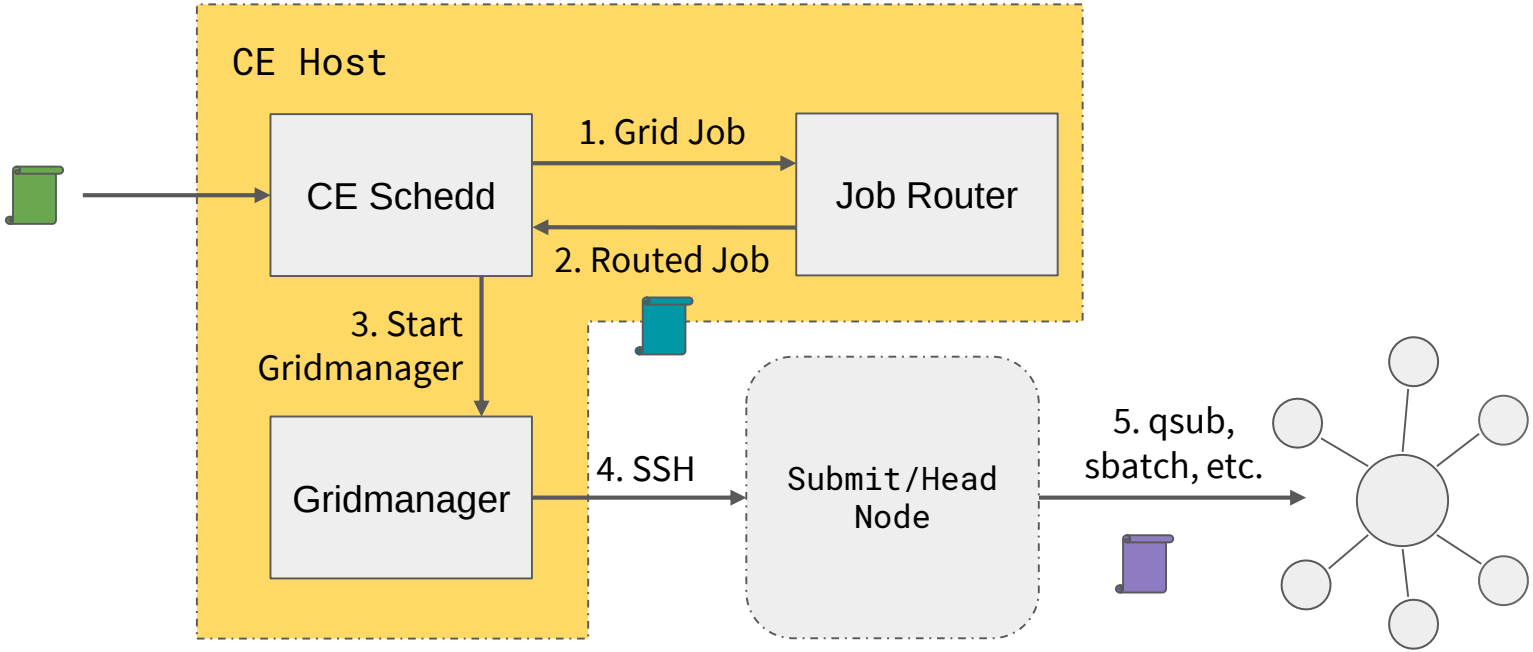
HTCondor-CE + HTCondor + Non-HTCondor



HTCondor-CE + SSH

- Using BOSCO (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



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 SPOOL`)
- 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

```
$ condor_ce_info_status
Name                        CPUs    Memory  MaxWallTime AllowedVOs

OU_OSCER_ATLAS_2650         20       32768           4320 atlas, dosar
OU_OSCER_ATLAS_2670         24       65536           4320 atlas, dosar
R510                        8        24576           1440 osg, cms
R730xd                      12       32768           1440 osg, cms
OUHEP_ITB_1                 4        6144            1440 atlas, dosar
CancerComputer_MinneUE      12       24576           1440 osg, sbgrid, mis
USCMS-FNAL-WC1              8        16384           2850 "cms"
AGLT2-Dell1                 8        16000           4300 atlas
AGLT2-Dell2                 8        16000           4300 atlas
[...]
```

HTCondor-CE Information Systems

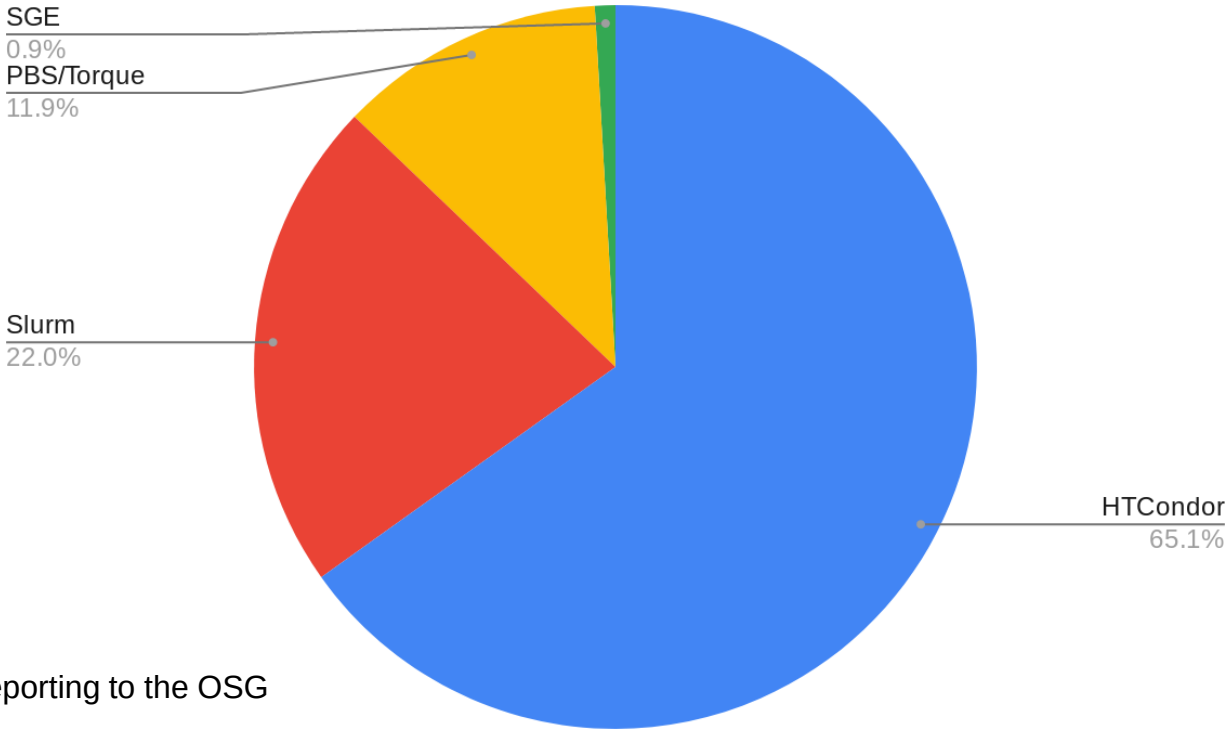
```
$ condor_status -schedd -pool collector.opensciencegrid.org:9619
Name                                    Machine                                 RunningJobs   IdleJobs   HeldJobs

CE01.CMSAF.MIT.EDU                      CE01.CMSAF.MIT.EDU                            412          7          6
CE02.CMSAF.MIT.EDU                      CE02.CMSAF.MIT.EDU                            669          1          0
CE03.CMSAF.MIT.EDU                      CE03.CMSAF.MIT.EDU                           1532          7          0
atlas-ce.bu.edu                         atlas-ce.bu.edu                              2568       1627
            27
bgk01.sdcc.bnl.gov                      bgk01.sdcc.bnl.gov                              5          2          0
bonner06.rice.edu                       bonner06.rice.edu                               7          1          8
brown-osg.rcac.purdue.edu               brown-osg.rcac.purdue.edu                       2          8          0
ce01.brazos.tamu.edu                    ce01.brazos.tamu.edu                            2          1          4
ce01.ific.uv.es                         ce01.ific.uv.es                              1012        265          0
ce1-vanderbilt.sites.opensciencegrid.org ce1-vanderbilt.sites.opensciencegrid.org      310        194          4
ce2-vanderbilt.sites.opensciencegrid.org ce2-vanderbilt.sites.opensciencegrid.org     1162        357          0
cit-gatekeeper.ultralight.org           cit-gatekeeper.ultralight.org                  76         33          2
cit-gatekeeper2.ultralight.org          cit-gatekeeper2.ultralight.org                 96         32          8
cit-gatekeeper3.ultralight.org          cit-gatekeeper3.ultralight.org                 98         34          3
[...]
```

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 $",
  "CondorVersion": "$CondorVersion: 8.6.13 Oct 30 2018 $",
  "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

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