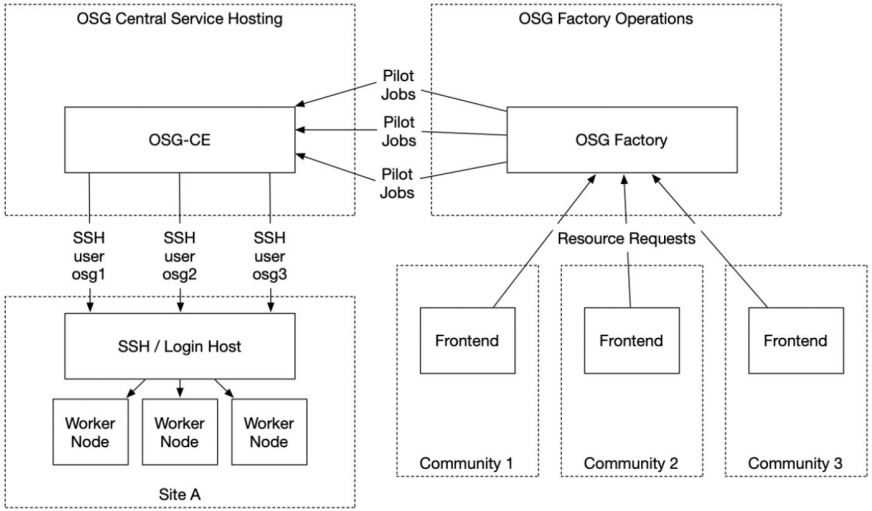
Hosted CE options/process and implementation

Marco Mascheroni, Jeff Dost

UC San Diego

The OSG pool: recap

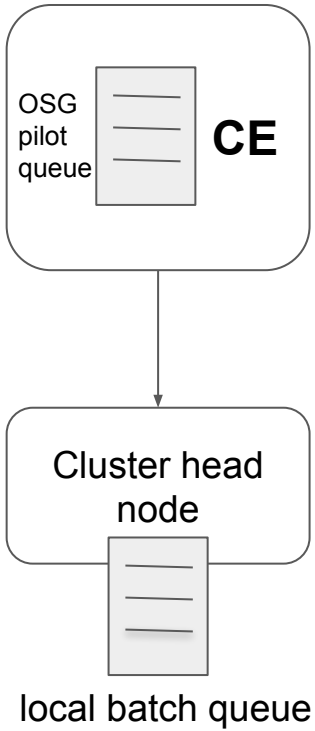
Manages a dynamic overlay pools that grows and shrinks based on user's request



- Based on the glideinWMS software
  - A Frontend component:
    - Looks at the *user* jobs in the community schedulers
    - Matches them over available Computing Element (CE)
    - Determines number of *pilot* jobs to submit
  - A pilot factory component:
    - Submit the pilot script to site CE
- Pilot jobs may run multiple user jobs
  - Configures and starts the condor startd daemon

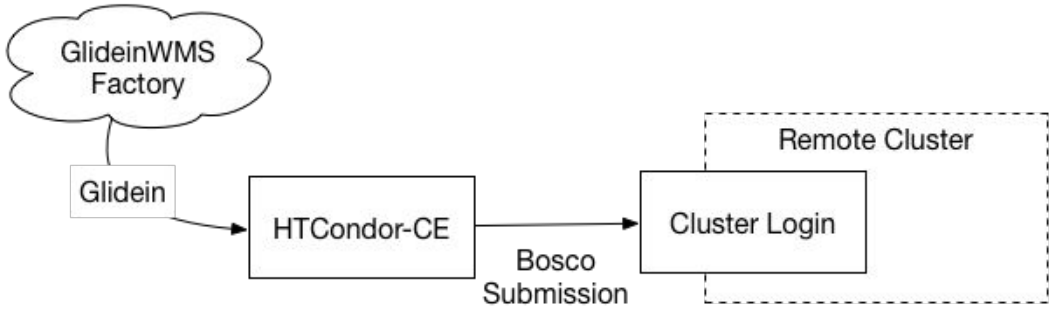
The Compute Element: a way into the cluster

- Set of services that provide access for **pilot** jobs to a local resource management system
  - jobs that arrive at a CE are **not end-user jobs**
- Takes care of **authentication, authorization** and delegation of jobs **to your** existing campus HPC/HTC **cluster**.
- Receives **jobs from the GWMS factory** and put them in a (condor) queue
- Responsible of **connect**ing **to** the **cluster head node** and submitting jobs
- **OSG will maintain and operate it** for you through the Hosted-CE initiative
- One Compute Element (CE) for each cluster contributing resources



Hosted-CE initiative

- OSG configures and maintains an HTCondor-CE on behalf of the site
- Special configuration of HTCondor-CE that can submit jobs to a remote cluster over SSH (through BOSCO)
- Provides a simple starting point for opportunistic resource owners that want to start sharing resources on OSG with minimal effort
  - Just need to allow SSH access to a submit node in your cluster



- Currently maintaining 17 CEs for various institutions

Hosted-CE requirements

- Existing compute cluster with a supported batch system running on a supported operating system
  - Local batch schedulers supported: HTCondor, LSF, PBS and TORQUE, SGE, and Slurm
  - OS Requirements: Red Hat Enterprise 6 and 7 and compatible platforms, for 64-bit Intel architectures. Select rebuilds of RHEL are also supported (CentOS 6, CentOS 7, Red Hat Enterprise Linux 6, Red Hat Enterprise Linux 7, Scientific Linux 6, Scientific Linux 7)
- Outbound network connectivity from the compute nodes (they can be behind NAT)
- Shared file system between the cluster head node and the compute nodes
- Temporary scratch space on each worker node

Requesting a Hosted-CE

- Requires site admins to fill in two forms (like you did)
  - One with "anagraphic" information
  - One with specific "configuration" information
- Upon completion OSG will contact you and ask you to grant access to a specific ssh key
  - We ask you to create 20 accounts on the cluster head node (from osg01 to osg20)
  - Correspond to the different communities (plus some spare)
  - The ssh key is shared among the 20 accounts, but each CE has its own
- As soon as the ssh connection works OSG will install the required software on the cluster head node

Software running on the Cluster Head Node

- The CE will start a GAHP server on the cluster head node once it receives jobs
  - The Grid Ascii Helper Protocol (GAHP) is the standard protocol used for communication with CEs on the Grid
- The CE will then talk to the GAHP server on the head node and perform the required operations
  - Submit new jobs (slurm_submit.sh, pbs_submit.sh, ...)
  - Query the status of existing jobs (slurm_status.sh, pbs_status.sh)

```
root      40988  0.0  0.0 137360  4324 ?        Ss   11:01   0:00  \_ sshd: ligouser [priv]
ligouser  40992  0.0  0.0 137360  2036 ?        S    11:01   0:00  |   \_ sshd: ligouser@notty
ligouser  40993  0.0  0.0 144956  3180 ?        Ss   11:01   0:00  |       \_ /home/ligouser/bosco/glite/bin/batch_gahp
root      41149  0.0  0.0 137360  4324 ?        Ss   11:01   0:00  \_ sshd: ligouser [priv]
ligouser  41153  0.0  0.0 137828  2536 ?        S    11:02   0:00      \_ sshd: ligouser@notty
ligouser  41158  0.0  0.0 106124  1440 ?        Ss   11:02   0:00          \_ /bin/bash -l -c echo Allocated port 46057 for remote
forward to 1>&2 ; CONDOR_CONFIG=~/bosco/glite/etc/condor_config.ft-gahp ~/bosco/glite/bin/condor_ft-gahp -f
ligouser  41302  0.0  0.0  52356  4788 ?        S    11:02   0:00              \_ /home/ligouser/bosco/glite/bin/condor_ft-gahp -f
```

Software installed on the head node (details)

- Everything goes under the *bosco* directory
- The *sandbox* dir contains the OSG pilot and the certificates:

```
[osg01@xxx]$ ls bosco/sandbox/fb08/fb08fcb4/hosted-ce21.grid.uchicago.edu_9619_hosted-ce21.grid.uchicago.edu#82514.0#1568645702
condor_exec.exe  credential_osg-flock-grid-iu-edu_OSG_gWMSFrontend.main_291573
```

payload (the pilot script)

credential used to connect to the overlay pool

Dirname contains hosted-CE hostname and condor jobis

- The *glite* directory contains the actual software that is executed to submit jobs to the local cluster:

Bin contains the software executables

```
[osg01@XXX ~]$ ls bosco/glite/
bin  etc  lib  log  share
```

Cluster head node queue

- Jobs reaches the cluster queue after the *_submit.sh script is executed
- Sample slurm output:

```
67892         bl_547c98+ slurm-bat+       osg          4      RUNNING       0:0
67893         bl_daac41+ slurm-bat+       osg          4      RUNNING       0:0
67894         bl_8f7053+ slurm-bat+       osg          4      RUNNING       0:0
67895         bl_956cd9+ slurm-bat+       osg          4      RUNNING       0:0
67896         bl_4eb96c+ slurm-bat+       osg          4      RUNNING       0:0
67897         bl_1160dc+ slurm-bat+       osg          4      RUNNING       0:0
67898         bl_6de8e0+ slurm-bat+       osg          4      RUNNING       0:0
67899         bl_3fe9b5+ slurm-bat+       osg          4      RUNNING       0:0
67900         bl_3223ce+ slurm-bat+       osg          4      PENDING       0:0
67901         bl_3a4c6f+ slurm-bat+       osg          4      PENDING       0:0
```

bl_nnnnn+ is a file written by slurm_submit.sh

What is done on the nodes

- The process run on the worker nodes is the pilot script
- Making sure the resources acquired are configured properly. The pilot executes a set of validation scripts.
  - Factory for general validation
    - Check condor version, collector setup, X509 proxy cert validation, publish information about the node
  - Frontend for experiment specific checks
    - Singularity validation, Cvmfs and OS checks, Benchmark, Network and squid proxy setup, CPU and memory checks
- Connecting to the user pool
  - Once we know the nodes are ok, the pilot script writes out a condor_config file and starts the startd daemon

Process tree on the worker node

```
condor_startd -f
  └─condor_starter -f -a slot1 test-006.t2.ucsd.edu
      └─condor_exec.exe /var/lib/condor/execute/dir_146125/condor_exec.exe -v std -name gfactory_instance -entry OSG_US_UCSD_TEST_CE_004 -clientname test.test -schedd...
          └─condor_startup. /var/lib/condor/execute/dir_146125/glide_q5Xf8G/main/condor_startup.sh glidein_config
              └─condor_master -f -pidfile /var/lib/condor/execute/dir_146125/glide_q5Xf8G/condor_master2.pid
                  ├─condor_procd -A /var/lib/condor/execute/dir_146125/glide_q5Xf8G/log/procd_address -L /var/lib/condor/execute/dir_146125/glide_q5Xf8G/log/ProcLog -R ...
                  └─condor_startd -f
```

- The first processes (condor_startd and condor_starter) are batch system dependant
  - Could be slurm, pbs, etc, depends on your batch system
- The condor_exec.exe is the pilot job (a bash script)
  - It eventually starts another condor_startd process that connects to the overlay pool
  - It will run multiple “user jobs”, possibly from different users
- Will see more processes once user jobs start
  - As children of the inner startd

Monitor information (Factory)

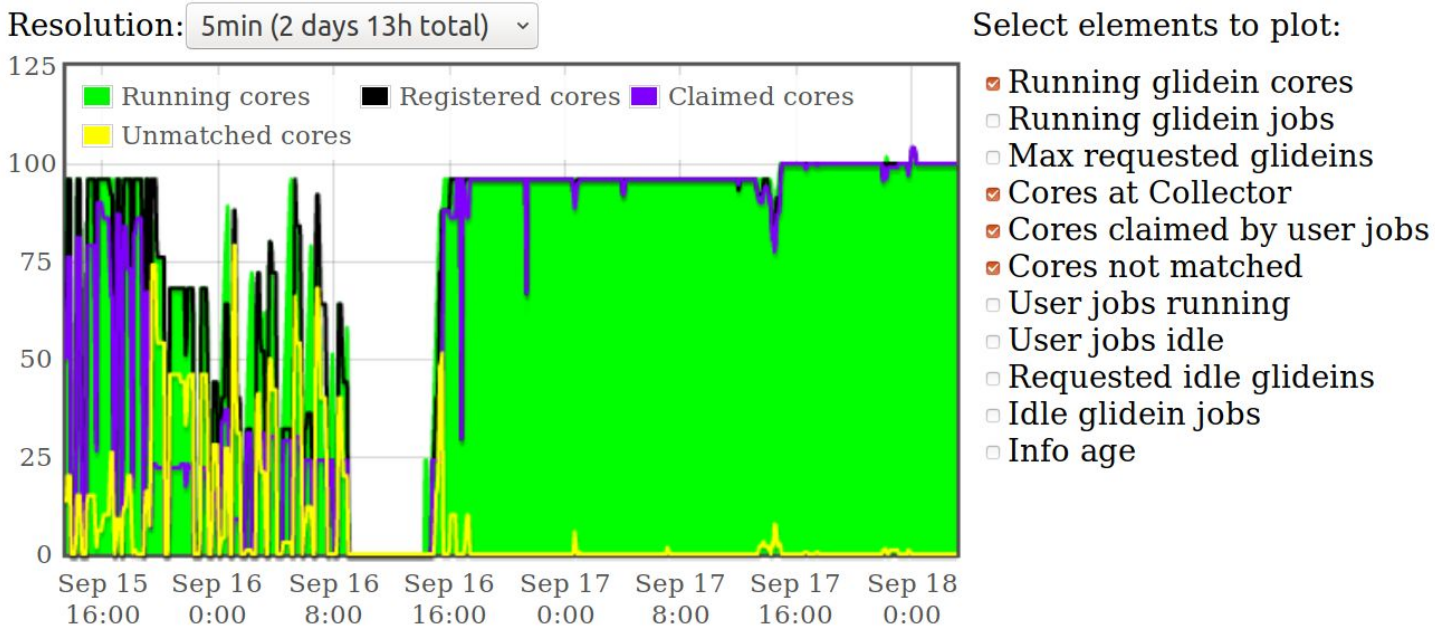
Available here. Shows:

Cores **running**

Cores able to **registered** back to the overlay pool

Cores **claimed** and running user jobs

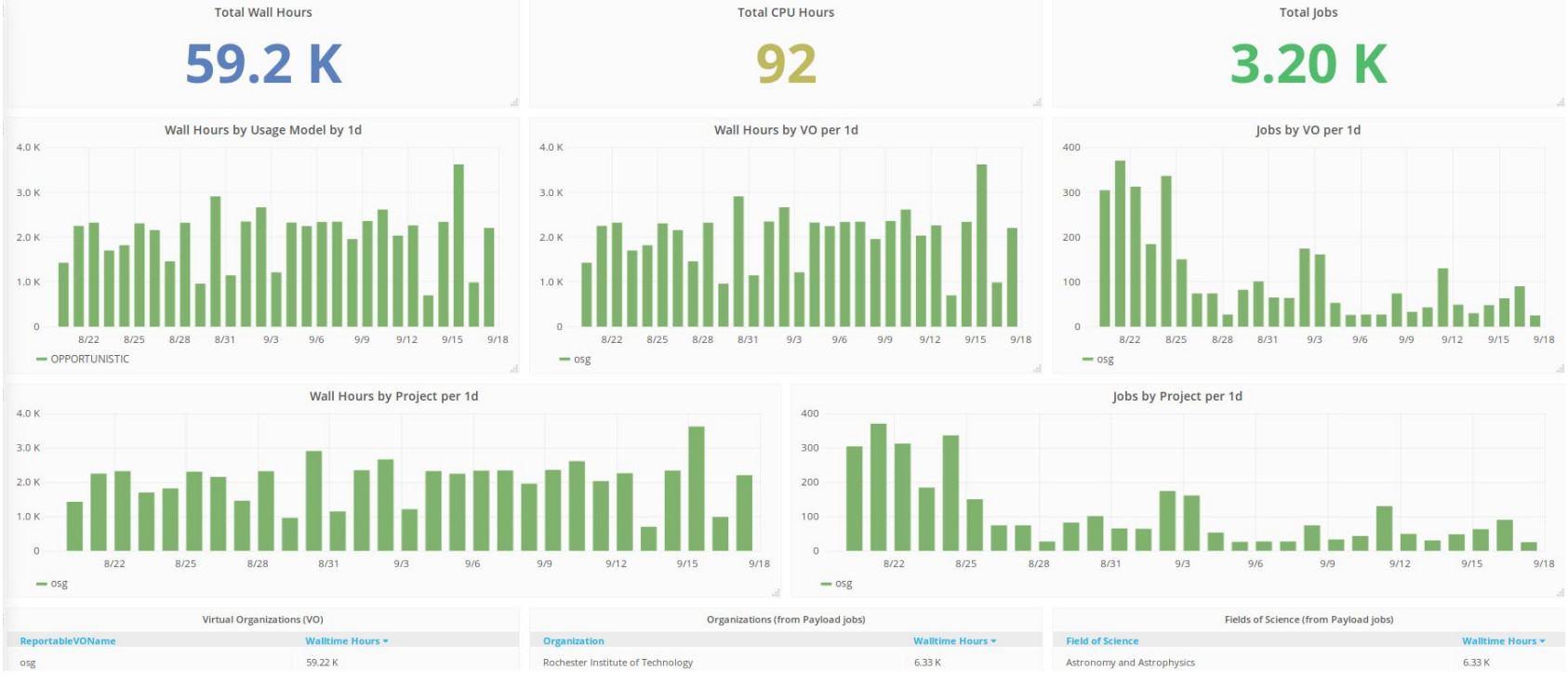
Registered cores waiting for user jobs (**unmatched**)



N.B.: Running > Registered = Claimed + Unmatched

Monitor information (GRACC)

Can be found here. Collects pilot information.



Remarks

- Institutions are still the owner of the resources
- Admins can do disruptive actions if OSG jobs are affecting the cluster
  - Kill OSG jobs
  - Cap them (although we are already capping them in the gwms factory)
  - Ban osgNN users (or disable ssh access)
- Please let us know if you do something long/disruptive
  - Don't need to know small interventions (e.g.: killing few jobs or even the entire queue once in a while, short downtimes, etc.)
  - Contact help@opensciencegrid.org if you completely interrupt OSG contribution for more than one day (e.g. downtime, disable SSH access or otherwise ban osgNN users from running)
- We will put the CE in downtime if you foresee a long intervention

Contacts information

Communication with the OSG team goes through freshdesk or help@opensciencegrid.org