



Glidein Factory Operations

The goal of this talk is to first define what tasks are performed in daily Factory operations, and then to describe the common procedures used to execute them.

Operational Tasks



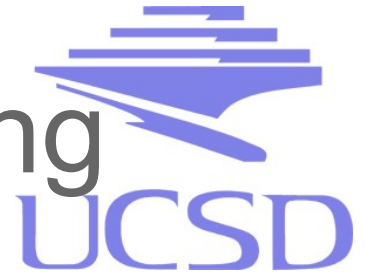
- Monitoring / Site Debugging
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - Upgrading Factory Condor
 - Upgrading Glidein Condor
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - Adding a New Frontend

Operational Tasks



- **Monitoring / Site Debugging**
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - Upgrading Factory Condor
 - Upgrading Glidein Condor
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - Adding a New Frontend

Monitoring / Site Debugging



- More will be covered in a separate talk
- Links to some useful monitoring pages:
 - glidein-
1.t2.ucsd.edu:8319/glidefactory/monitor/glidein_Production_v4_1/factoryStatus.html
 - glidein-
1.t2.ucsd.edu:8319/glidefactory/monitor/glidein_Production_v4_1/factoryStatusNow.html
 - [http://glidein-1.t2.ucsd.edu:8319/glidefactory/monitor/glidein_Production_v4_1/factoryCompletedStats.html](http://1.t2.ucsd.edu:8319/glidefactory/monitor/glidein_Production_v4_1/factoryCompletedStats.html)

Monitoring / Site Debugging



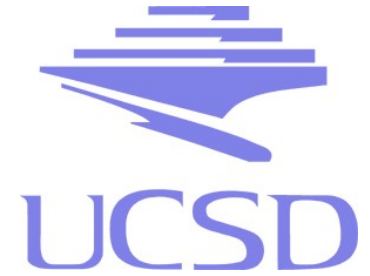
- Useful log locations:
 - Condor job logs:
 - ~/glideinsubmit/glidein_Production_v4_1/client_log/user_
<feuser>/entry_**<entry>**/*
 - GridManagerLogs:
 - /glidein01/condor/shared_log/*
 - Condor daemon logs:
 - /opt/glidecondor/condor_local/log/
 - GlideinWMS logs:
 - ~/glideinsubmit/glidein_Production_v4_1/log/*

Monitoring / Site Debugging



- A number of useful tools can be found in `~/glideinWMS/factory/tools/`:
 - **analyze_entries** – generates reports of aggregated successes / failures of glideins
 - **entry_q** – a wrapper for `condor_q` to query by entry name
 - **cat_StartdLog.py** – decompresses and outputs startd log from condor job error log
 - **proxy_info** – like `voms-proxy-info` plus needed `privsep` to access frontend proxies

Operational Tasks



- Monitoring / Site Debugging
- Updating Factory Configuration
 - **Adding a New Site Entry**
- Service Maintenance
 - Upgrading Factory Condor
 - Upgrading Glidein Condor
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - Adding a New Frontend

Some Basics First



- To stop the Factory:

```
gfactory@glidein-1 $ cd ~/glideinsubmit/glidein_Production_v4_1
gfactory@glidein-1 $ ./factory_startup stop
```

- To modify the config, edit **glideinWMS.xml**:

```
gfactory@glidein-1 $ vi ../glidein_Production_v4_1.cfg/glideinWMS.xml
```

- It is useful to diff against the current config:

```
gfactory@glidein-1 $ diff glideinWMS.xml
../glidein_Production_v4_1.cfg/glideinWMS.xml
```


Some Basics First



- To Reconfigure the Factory:

```
gfactory@glidein-1 $ ./factory_startup reconfig  
../glidein_Production_v4_1.cfg/glideinWMS.xml
```

- It good to make sure there are no running factory python processes before restarting:

```
gfactory@glidein-1 $ ps -u gfactory
```

- To Restart the Factory:

```
gfactory@glidein-1 $ ./factory_startup start
```

Adding a New Site Entry



- Assuming CE is known (e.g. from siteDB), do a preliminary BDII search:

```
gfactory@glidein-1 $ ldapsearch -p2170 -h exp-bdii.cern.ch -xLLL -b 'mds-vo-name=local,o=grid' "GlueCEUniqueID=osg-gw-4.t2.ucsd.edu*"
```

- Alternatively a shorthand script can be used:

```
gfactory@glidein-1 $ lds egi "osg-gw-4.t2.ucsd.edu*"
```

- “egi” may be replaced with “osg” to query from is.grid.iu.edu instead of exp-bdii.cern.ch

Adding a New Site Entry



- CE entries are returned with dn's of the form:

```
dn: GlueCEUniqueID=osg-gw-4.t2.ucsd.edu:2119/jobmanager-condor-  
default,Mds-Vo-name=UCSDT2,Mds-Vo-name=local,o=grid
```

- Fields to look for:

```
GlueCEImplementationName: Globus  
GlueCEPolicyMaxCPUtime: 1440  
GlueCEPolicyMaxWallClockTime: 1440  
GlueCEAccessControlBaseRule: VO:cms
```

Adding a New Site Entry



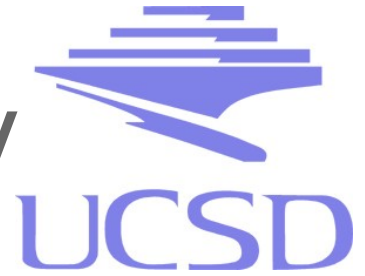
- Pick a queue from BDII that:
 - Supports the VOs you expect in the **AccessControlBaseRule** and has sufficiently large **MaxCPUTime** or **MaxWallClockTime** (smaller of the two)
 - *Note time is reported in minutes
 - Pick a similar existing entry from glideinWMS.xml to use as a template with the following suggestions:
 - If **GlueCEImplementationName: Globus:**
 - Choose an entry with **gridtype="gt2"**
 - Else if **GlueCEImplementationName: CREAM:**
 - Choose an entry with **gridtype="cream"**

Adding a New Site Entry



- Case **gridtype="gt2"**:
 - GlueCEUniqueID is always of the form:
<host>:2119/jobmanager-<batch>-<queue>
 - Change **gatekeeper** attribute to:
<host>:2119/jobmanager-<batch>
 - In **rsi** be sure to set appropriate (**queue=<queue>**)
*Note this isn't really used for condor batch systems so we leave it as (**queue=default**)

Adding a New Site Entry



- Case **gridtype="cream"**:
 - GlueCEUniqueID is always of the form:
<host>:8443/cream-<batch>-<queue>
 - Change **gatekeeper** attribute to:
**https://<host>:8443/ce-cream/services/CREAM2
<batch> <queue>**
 - *Note there should be no **rsI** for CREAM CEs

Adding a New Site Entry



- Some tips on setting **work_dir** attribute:
 - If the batch system is condor always set to **work_dir="Condor"**
 - If site is a non-condor OSG site set to **work_dir="OSG"**
 - If site is EGI you should run a test job to determine if \$TMPDIR is defined and exists on the Worker Node.
 - If defined, set to **work_dir="TMPDIR"**
 - Otherwise set to **work_dir="."** and make note in **comment** that TMPDIR is not defined

Adding a New Site Entry



- Other attribute changes:
 - Give the **entry** a unique **name** ideally using some combination of site name and gatekeeper hostname and / or queue
 - Unless you have reason to explicitly define it, remove the **schedd** attribute to let reconfig make the best choice for load balancing
 - Set **GLIDEIN_Max_Walltime** to $\text{MaxCPUTime} * 60 - 1800$
 - * (Or use **MaxWallClockTime** if it is smaller)
 - Set **GLIDEIN_ResourceName** to Mds-VO-name as listed in BDII (e.g. UCSDT2)

Adding a New Site Entry



- Other attribute changes:
 - Set **GLIDEIN_Site** to a unique name used to group multiple entries that belong to the same site (e.g. UCSD)
 - Ensure **GLIDEIN_Supported_VO**s only has list of frontends you want to advertise this entry to
 - Set **infosys_ref** to DN of BDII entry

Adding a New Site Entry



- CMS specific attributes:
 - **GLIDEIN_CMSSite** attribute must match cms site name in sitedb
 - Set **GLIDEIN_SEs** to SE listed in cms sitedb (found to generally be more accurate than BDII)
- Once you are satisfied with the new entry, reconfigure and restart the factory.

Operational Tasks



- Monitoring / Site Debugging
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - **Upgrading Factory Condor**
 - Upgrading Glidein Condor
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - Adding a New Frontend

Upgrading Factory Condor



- Go to the condor website and download tarballs:
 - <http://www.cs.wisc.edu/condor/downloads-v2/download.pl>
- For the UCSD factory we currently use **condor-<rel>-x86_rhap_5-stripped.tar.gz**

```
root@glidein-1 $ cd ~/Downloads
root@glidein-1 $ wget
http://parrot.cs.wisc.edu//symlink/20111104041502/7/7.6/7.6.4/8958702130f
10a30fbf9c98bea3eaf5b/condor-7.6.4-x86_rhap_5-stripped.tar.gz
```

Upgrading Factory Condor



- **Stop Factory:**

```
gfactory@glidein-1 $ cd ~/glideinsubmit/glidein_Production_v4_1
gfactory@glidein-1 $ ./factory_startup stop
```

- **Stop Condor either with init.d script:**

```
root@glidein-1 $ /etc/init.d/condor stop
```

- **Or manually:**

```
root@glidein-1 $ killall -9 condor_master condor_schedd condor_collector
condor_negotiator condor_procd condor_gridmanager gahp_server
root@glidein-1 $ killall condor_master
```

Upgrading Factory Condor



- Run upgrade script:

```
root@glidein-1 $ /root/glideinWMS/install/glidecondor_upgrade condor-7.6.4-x86_rhap_5-stripped.tar.gz
```

- Start Condor either with init.d script:

```
root@glidein-1 $ /etc/init.d/condor start
```

- Or manually:

```
root@glidein-1 $ /opt/glidecondor/start_condor.sh
```

- Monitor **top** and ensure %id is > 10 before continuing. This can take as long as 20 minutes because there are a lot of glideins the GridManagers have to reestablish connections with.

Upgrading Factory Condor



- Once the load has decreased, it is safe to restart the Factory:

```
gfactory@glidein-1 $ ./factory_startup start
```

Operational Tasks



- Monitoring / Site Debugging
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - Upgrading Factory Condor
 - **Upgrading Glidein Condor**
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - Adding a New Frontend

Upgrading Glidein Condor



- Go to the condor website and download tarballs:
 - <http://www.cs.wisc.edu/condor/downloads-v2/download.pl>
- For the glidein tarballs we currently use
 - **condor-<rel>-x86_rhap_5-stripped.tar.gz**
 - **condor-<rel>-x86_64_rhap_5-stripped.tar.gz**

```
gfactory@glidein-1 $ cd ~/Downloads
gfactory@glidein-1 $ wget
http://parrot.cs.wisc.edu//symlink/20111104041502/7/7.6/7.6.4/8958702130f
10a30fbf9c98bea3eaf5b/condor-7.6.4-x86_rhap_5-stripped.tar.gz
```

Upgrading Glidein Condor



- Go to appropriate prestage area and create glidein tarballs:

```
gfactory@glidein-1 $ cd
~/glideinsubmit/glidein_Production_v4_1.cfg/Prestage
gfactory@glidein-1 $ ~/glideinWMS/creation/create_condor_tarball
gfactory-2.5.2-condor-7.6.4-x86_rhap_5-stripped.tar.gz
~/Downloads/condor-7.6.4-x86_rhap_5-stripped.tar.gz
```

Upgrading Glidein Condor



- Edit **glideinWMS.xml**:
 - For **each tarball created** above find the previous condor version in the **<condor_tarballs>** section that had **version="default"**.
 - Note there are two entries, one with **version="<prev>"** and another with **version="default"**.
 - Create a new entry for **version="<new>"** but do not remove the entry for the previous version.
 - Change the entry with **version="default"** to the new one if you want it to be the new default version.
 - Do the same for special entries like **version="7.6.x"** as needed.

- Before:

```
<condor_tarball arch="default" os="default"
tar_file="/home/gfactory/glideinsubmit/glidein_Production_v4_1.cfg/Presta
ge/gfactory-2.5.2-condor-7.6.3-x86_rhap_5-stripped.tar.gz"
version="7.6.3"/>
<condor_tarball arch="default" os="default"
tar_file="/home/gfactory/glideinsubmit/glidein_Production_v4_1.cfg/Presta
ge/gfactory-2.5.2-condor-7.6.3-x86_rhap_5-stripped.tar.gz"
version="default"/>
```

- After:

```
<condor_tarball arch="default" os="default"
tar_file="/home/gfactory/glideinsubmit/glidein_Production_v4_1.cfg/Presta
ge/gfactory-2.5.2-condor-7.6.3-x86_rhap_5-stripped.tar.gz"
version="7.6.3"/>
<condor_tarball arch="default" os="default"
tar_file="/home/gfactory/glideinsubmit/glidein_Production_v4_1.cfg/Presta
ge/gfactory-2.5.2-condor-7.6.4-x86_rhap_5-stripped.tar.gz"
version="7.6.4"/>
<condor_tarball arch="default" os="default"
tar_file="/home/gfactory/glideinsubmit/glidein_Production_v4_1.cfg/Presta
ge/gfactory-2.5.2-condor-7.6.4-x86_rhap_5-stripped.tar.gz"
version="default"/>
```

Upgrading Glidein Condor



- Some more **glideinWMS.xml** tips:
 - 64-bit tarballs should have **arch="x86_64"** but 32-bit tarballs should have **arch="default"**
 - Similarly if the tarball is for sites using rhel5, set **os="default"**. Otherwise change accordingly.
 - If the site entry does not specify **CONDOR_ARCH**, **CONDOR_OS**, or **CONDOR_VERSION** in its **<attrs>** section, it will use the tarball that specifies **"default"** for the corresponding tarball entries.

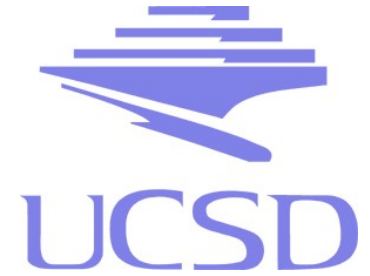
Upgrading Glidein Condor



- Reconfigure and restart factory:

```
gfactory@glidein-1 $ cd ~/glideinsubmit/glidein_Production_v4_1
gfactory@glidein-1 $ ./factory_startup stop
gfactory@glidein-1 $ ./factory_startup reconfig
../glidein_Production_v4_1.cfg/glideinWMS.xml
gfactory@glidein-1 $ ./factory_startup start
```

Operational Tasks



- Monitoring / Site Debugging
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - Upgrading Factory Condor
 - Upgrading Glidein Condor
 - **Obtaining / Updating GlideinWMS code**
- VO Frontend Registration and Support
 - Adding a New Frontend

Obtaining GlideinWMS



- We use **git** to obtain and update glideinWMS code
- To obtain glideinWMS from git:

```
gfactory@glidein-1 $ git clone http://cdcvs.fnal.gov/projects/glideinwms  
glideinWMS
```

- At UCSD we currently run glideinWMS version **branch_v2_5_2plus_ucsd_factory**:

```
gfactory@glidein-1 $ cd glideinWMS  
gfactory@glidein-1 $ git checkout branch_v2_5_2plus_ucsd_factory
```


Updating GlideinWMS



- Before updating glideinWMS first shut down the Factory:

```
gfactory@glidein-1 $ cd ~/glideinsubmit/glidein_Production_v4_1
gfactory@glidein-1 $ ./factory_startup stop
```

- Check to make sure there are no running factory python processes:

```
gfactory@glidein-1 $ ps -u gfactory
```

- Update code from git:

```
gfactory@glidein-1 $ cd ~/glideinWMS
gfactory@glidein-1 $ git stash
gfactory@glidein-1 $ git pull
gfactory@glidein-1 $ git stash pop
```

Updating GlideinWMS



- **IMPORTANT** reconfigure the factory after updating the code
 - Updates contained in glideinWMS/creation will **only** be propagated to the factory instance if reconfig is run.

```
gfactory@glidein-1 $ cd ~/glideinsubmit/glidein_Production_v4_1
gfactory@glidein-1 $ ./factory_startup reconfig
../gideinsubmit/glidein_Production_v4_1.cfg/glideinWMS.xml
```

- Restart factory:

```
gfactory@glidein-1 $ ./factory_startup start
```

Operational Tasks



- Monitoring / Site Debugging
- Updating Factory Configuration
 - Adding a New Site Entry
- Service Maintenance
 - Upgrading Factory Condor
 - Upgrading Glidein Condor
 - Updating GlideinWMS code
- VO Frontend Registration and Support
 - **Adding a New Frontend**

Adding a New Frontend



- Create new user:

```
root@glidein-1 $ useradd fecernitb
```

- Add user to **/etc/condor/privsep_config**:

```
valid-target-uids = fecms : fesleep : ... : fecernitb  
valid-target-gids = feglow : fecms : ... : fecernitb
```

- Run **glidecondor_addDN** to authenticate with condor:

```
root@glidein-1 $ /root/glideinWMS/install/glidecondor_addDN -daemon 'CERN  
ITB Frontend Marian Zvada marian.zvada@cern.ch'  
/DC=ch/DC=cern/OU=computers/CN=vocms157.cern.ch fecernitb
```

Adding a New Frontend



- **IMPORTANT** reconfigure the condor_collector after running **glidecondor_addDN**

```
root@glidein-1 $ killall -HUP condor_collector
```

- If this step is not performed any requests from the new frontend will be mapped to user **anonymous** and will be rejected by the factory collector!

Adding a New Frontend



- Add new frontend to glideinWMS.xml:

```
</frontends>
...
  <frontend name="CMSV0-vocms157-ITB" comment="Contact: Marian
Zvada marian.zvada@cern.ch, Ignas Butenas ignas.butenas@cern.ch"
identity="fecernitb@glidein-1.t2.ucsd.edu">
  <security_classes>
    <security_class name="frontend" username="fecernitb"/>
  </security_classes>
</frontend>
...
</frontends>
```

- Reconfigure and restart the Factory

Adding a New Frontend



- Notify the Frontend Admin that registration is complete as well as the following:
 - The mapped frontend identity (e.g. fecernitb@glidein-1.t2.ucsd.edu)
 - Security name (e.g. CMSVO-vocms157-ITB)
 - Factory Collector host: glidein-1.t2.ucsd.edu
 - Factory DN:
 - /DC=org/DC=doegrids/OU=Services/CN=glidein-1.t2.ucsd.edu
 - Factory identity: gfactory@glidein-1.t2.ucsd.edu
 - Supported_VO **factory query_expr**:
 - stringListMember("CMS",GLIDEIN_Supported_VO)

Frontend Support



- For support we recommend frontend admins to contact us at: osg-gfactory-support@physics.ucsd.edu

Acknowledgements



- This work is partially sponsored by
 - the US Department of Energy under Grant No. DE-FC02-06ER41436 subcontract No. 647F290 (OSG), and
 - the US National Science Foundation under Grants No. PHY-0612805 (CMS Maintenance & Operations), and OCI-0943725 (STCI).
- Special thanks to the glideinWMS and Condor teams