Monitoring and Ops Tools Demo

These slides summarize some techniques shown in the live ops demo

Factory Tools

- Useful tools for daily operations can be found in:

  ~/glideinWMS/factory/tools/analyze_entries

- We highly recommend putting this in your path
- These tools should generally be run from the directory of the current factory instance, e.g.

  ~/glideinsubmit/glidein_Production_v4_1

Factory Tools

- Tools used in demo:
  - analyze_entries
  - entry_q
  - proxy_info
  - cat_StartdLog.py

Analyze Entries

- **analyze_entries** is a useful tool for generating reports on glidein successes / failures over a specified time period.
- Example:

```
gfactory@glidein-1 $ analyze_entries -x 24 -s waste
```

- **-x** specifies the time period. It can be 2, 24, or 168 hours
- **-s** specifies which column to sort by
- **-m** can optionally be used to give totals only, broken down by frontends

entry_q

- **entry_q** is a wrapper for **condor_q** but accepts an entry name.
- This is useful since you do not need to specify schedd or necessary constraints
- Example:

```
gfactory@glidein-1 $ entry_q CMS_T2_UK_SGrid_RALPP_hep206_grid -const
'jobstatus==5'
```

- **entry_q** accepts any additional arguments that a normal **condor_q** would accept.

entry_q

- To discover the actual schedd as well as the constraint used by **entry_q** use the **-printschedd** argument directly after listing the entry
- Example:

```
gfactory@glidein-1 $ entry_q CMS_T2_UK_SGrid_RALPP_hep206_grid
-printschedd -const 'jobstatus==5'
```

- Sample output:

```
Schedd: schedd_glideins14@glidein-1.t2.ucsd.edu
Constraint: '((GlideinFactory=?="UCSD")&&(GlideinName=?
="Production_v4_1")&&(GlideinEntryName=?
="CMS_T2_UK_SGrid_RALPP_hep206_grid"))&&(JobStatus==5)'
...
```

Show Glidein Proxy Info

- **proxy_info** can be used to obtain proxy information from the pilot proxies.
- You can't do a normal **voms-proxy-info** due to privilege separation. **proxy_info** gets around this by using the **condor_root_switchboard**
- Example:

```
gfactory@glidein-1 $ source /opt/vdt/setup.sh
gfactory@glidein-1 $ proxy_info ferenci info -all <proxy path>
```

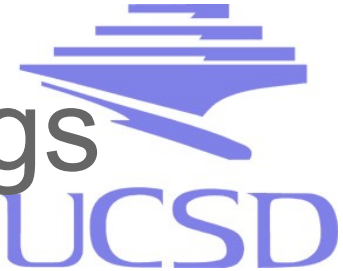
- In addition to **info -all**, **proxy_info** can also do useful commands like **ls -ltr** or **cat** on the path given

Show Glidein Proxy Info

- It is useful to do an **entry_q** first to find out the path of the pilot proxy used by the glidein in question:

```
gfactory@glidein-1 $ entry_q CMS_T2_US_Wisconsin_cms01 -const 'Owner=?
="ferenci"' -format '%s\n' x509userproxy |sort |uniq -c
```

Reading Glidein Startd Logs



- **cat_StartdLog.py** along with other **cat_*Log.py** tools in ~/glideinWMS/factory/tools/ can be used to read the glidein condor daemon logs sent from the WN.
- Example:

```
gfactory@glidein-1 $ cat_StartdLog.py
client_log/user_ferenci/entry_HCC_US_UConn_gluskap/job.76301.5.err
```

- The argument is the condor error log of the glidein

Synchronizing Multiple Factories

- Use the **clone_glidein** tool
- Example:

```
gfactory@glidein-1 $ ~/glideinWMS/creation/clone_glidein -exclude
GLIDEIN_Site CERN  -out glideinWMS.xml.test -merge yes -other
glideinWMS.xml.ucsd glideinWMS.xml
```

- Specify **-other** to pass in the alternate config of the config you want to obtain changes from
- Specify **-out** to name the generated output file.
- **-exclude** can be used to prevent entries with certain attributes from being added to the new file