ITB Factory as Infosys

Jeff Dost (UCSD)
OSG Glidein Factory Operations

Problem

- Creating and maintaining site entry points in factory config is not automated enough
- Currently we add sites if a VO complains* there is a site missing from our config that they know they can run on
- Currently we change our config when site admins complain* they have updated something rendering our entry obsolete

* A small subset of changes can be determined without site or vo help by observing glidein validation failures and Hold errors

Problem

- Essentially we want to make factory config maintenance more automated, and less complaint driven!

Solution

- Periodically query available information systems:
  - BDII
  - OSG Collector (for OSG Sites running HTCondor CEs)
- Create a corresponding entry for **EVERY** Site CE in the GOC ITB factory as advertised in infosys
- VOs and Factory ops can then test CEs based on GOC ITB factory list
- Once entry appears suitable for production, we push it to the production factories
- Periodically compare factory configs to infosys to discover changes

Work Done

- Two tools in development:
  - **build_infocache** – queries each information system for every possible CE and queue for each VO we support, caches results to disk
  - **generate_xml** – generates a factory config file by creating corresponding xml entries for each queue in infocache

Available Demo

- Test factory instance at cabinet-10-10-5.t2.ucsd.edu can be queried to see auto generated entries:

```
$ condor_status -any -pool cabinet-10-10-5.t2.ucsd.edu
-const 'mytype=?="glidefactory" && stringlistmember("cms",
GLIDEIN_Supported_VOs, ",") && GLIDEIN_ResourceName=?
="DESY-HH"' -af EntryName
DESY-HH_grid-cr5_cms
DESY-HH_grid-cr4_cms
DESY-HH_grid-cr3_cms
DESY-HH_grid-cr2_cms
DESY-HH_grid-cr1_cms
DESY-HH_grid-cr0_cms
...
```

Available Demo

- Everything needed for a condor submit file is advertised:

```
$ condor_status -any -pool cabinet-10-10-5.t2.ucsd.edu
-const 'mytype=?="glidefactory" && EntryName=?="DESY-
HH_grid-cr1_cms"' -l
...
GLIDEIN_Gatekeeper = "grid-cr1.desy.de:8443/cream-pbs-cms"
GLIDEIN_GridType = "cream"
...
```

- Corresponding submit lines:

```
Universe = grid
Grid_Resource = cream grid-cr1.desy.de:8443/cream-pbs-cms
```

Notable Changes

- Entry names are now auto generated and based on infosys resource name instead of manually chosen:
  - Old: CMS_T2_US_UCSD_gw6
  - New: UCSDT2_osg-gw-6_default
- Names in supported VO list are as declared in infosys instead of manually chosen:
  - Old: CMS,glowVO,HCC,nanoHUB,SBGrid, …
  - New: cms,glow,hcc,nanohub,sbgrid …

Missing CMS pieces

- CMS specific attributes cannot be found in information systems (BDII, OSG Collector):
  - GLIDEIN_CMSSite
  - GLIDEIN_SEs*
- Historically we manually discover these by checking Dashboard

* some sites correctly report SE in BDII but not all

Proposal

- Factory ops to leave this mapping to CMS
- CMS periodically scans all of our entries and report to us a list of missing / changed GLIDEIN_CMSSite names
- Factory ops then sets updated CMSSite names accordingly
- Can we drop GLIDEIN_SEs?
  - Historically needed for CRAB2 but not WMAgent
  - Is it still used in CRAB3 / global FE matchmaking?

To Do List

- Implement ability to merge updated infosys changes into existing config
- Deal with scalability limits – poor factory child process management prevents us from having more than 500 entries in config, reported bug to gwms developers
- Deal with human limits – a config with > 500 entries becomes difficult to maintain over 4 production factories, working with gwms developers to be able to:
  - Split entries into multiple config files
  - Make it easier to version control factory configs