glideinWMS factory system setup

by Igor Sfiligoi, Jeff Dost (UCSD)

Overview

- Component overview
- Description of the components
- Hardware needs

Components of a factory

- The factory runs:
  - Condor
  - Web server
  - GlideinWMS
- It also needs:
  - OS
  - GSI security setup (cert & CAs)
  - Libraries (RRDTool, JavascriptRRD, ...)

Operating System

- GlideinWMS is supported on >= RHEL6 compatible systems
  - But may run on other *NIX systems as well Requires python 2.6 or above
- Root access is needed to install and operate some of the services
- Public TCP/IP is networking needed, both incoming and outgoing

Libraries

- Needed on top of standard OS libraries:
  - RRDTool: http://oss.oetiker.ch/rrdtool/
  - JavascriptRRD – available from Sourceforge Untarring suffice, but RPM install also available http://sourceforge.net/projects/javascriptrrd/
- Both available in OSG yum repo and come in as glideinWMS dependencies

Web server

- Used for both delivering payload to worker nodes and for monitoring
- No dynamic content served
  - Only static files
- glideinWMS rpm brings in httpd as dependency
  - also adds glideinWMS specific config into /etc/httpd/conf.d/

GSI setup

- CA & CRLs needed
  - Both for communication with frontend and resource providers
- Typically installed via OSG repo (but other means acceptable)
  https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallCertAuth
- Host cert needed for communication with frontends
- Full Grid client software recommended
  https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallOSGClient
  - Not essential, but useful for operations

Condor services

- Condor is used for
  - Collector for communication with frontend
  - Condor-G for communication with resource providers
    - Just a regular Schedd from user point of view Jobs are of the “Grid universe”
- Everything started and controlled by a Condor master
  - As usual for any Condor setup
- Also available from the OSG yum repo

Condor Collector

- Essentially a whiteboard
  - Factory and frontend advertise and read ClassAds
  - Also used to identify the schedds
- Authorization
  - FS/UID based for local services (factory&schedds)
  - GSI based for network access
    - Using host cert for own identification
    - Must whitelist the remote DNs that are allowed
    - Must provide a DN->ID mapping
  - Each classad has the identity (ID) of advertiser

Condor-G

- Many schedds
  - For improved scalability
  - Named schedd_glideins<n>@<hostname>
- Schedds only accessible from the local node
  - Schedd really just the interface
    - Condor-G does all the work
  - Schedd must run as root
    - needs to switch UIDs for security reasons

glideinWMS

- The glideinWMS rpms can be found in the OSG yum repo
- The factory will run as non-privileged **gfactory** user
  - gfactory has “Condor superuser” privileges
  - gfactory will uid switch to the relevant FE user before submitting on their behalf
- VO frontend user linux accounts must be created and gfactory must be authorized to uid switch to them in the following:
  - /etc/condor/privsep_config

Hardware needs

- HW needs to scale with the number of sites supported
  - Number of glideins is a 2nd order effect
  - Disk space scales primarily with number of glideins
- OSG factory experience
  - Uses ~12G of RAM
  - Requires SSD; spinning disks are too IO limited (space: <100G)

Summary

- RHEL6-compatible Linux recommended
- glideinWMS, Condor, and their dependencies are all available in the OSG Yum repos
- Requires incoming and outgoing TCP/IP
- Fast disks are preferred

Pointers

- glideinWMS development team is reachable at glideinwms-support@fnal.gov
- The official project Web page is http://glideinwms.fnal.gov
- OSG glidein factory at UCSD

  http://http://www.t2.ucsd.edu/twiki2/bin/view/UCSDTier2/OSGgfactory
  http://gfactory-1.t2.ucsd.edu/factory/monitor/

Acknowledgments

- glideinWMS is a CMS-led project developed at FNAL
- glideinWMS factory operations at UCSD is sponsored by OSG
- The funding comes from NSF, DOE and the UC system