glideinWMS: Making Grid Simple for Users

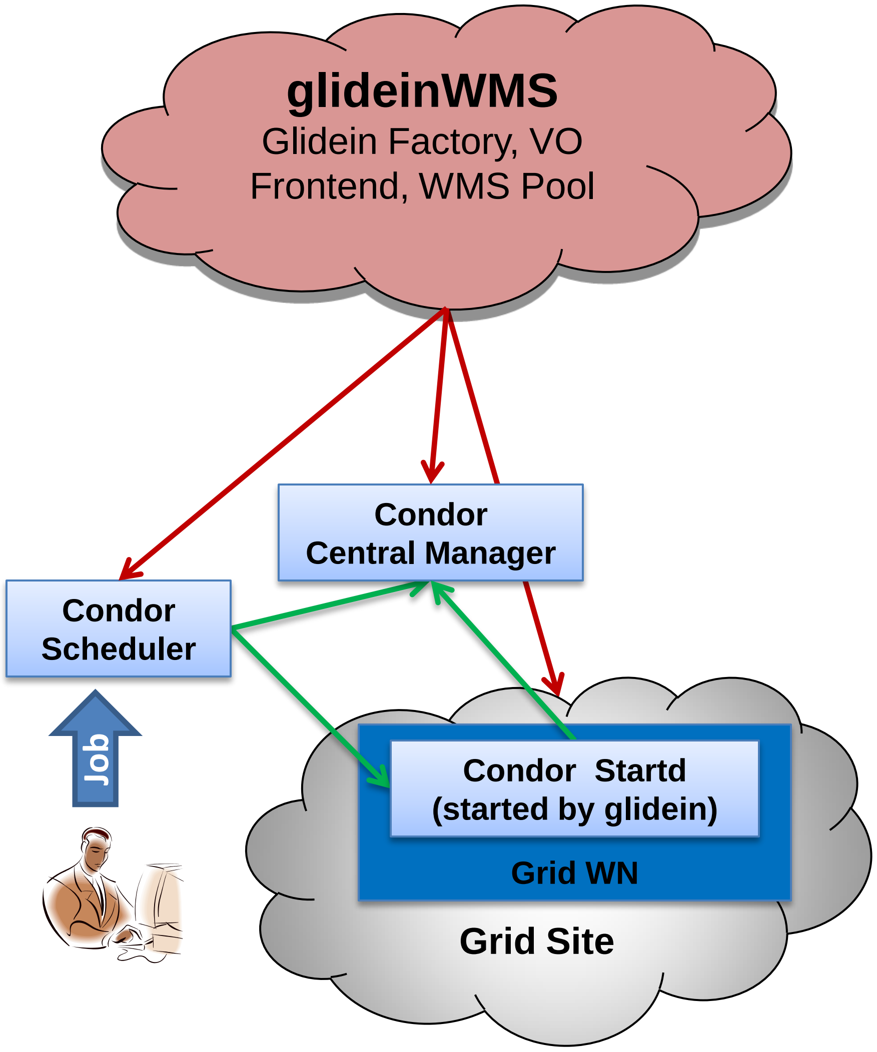
The glideinWMS –

- developed on top of the Condor system with just a thin layer on top of it
- Glidein Factory (GF) creates and submits glideins based on the demand
- Glideins running on the Grid worker nodes (WN) create condor pool of dynamically size to run user jobs.

From the user's point of view –

- glideinWMS is just a distributed Condor system
- shields the user from interfacing directly with the grid
- users with existing Condor-ready jobs can submit them to the Grid with minimal changes.
- the pilot can validate the node before running a user job; reducing the failure rate of user jobs.
- provides pseudo-interactive monitoring
- glideinWMS can prioritize jobs from different users.

UCSD currently hosts a factory that is open (upon request) to smaller OSG VOs.


glideinWMS
Glidein Factory, VO Frontend, WMS Pool
Condor Central Manager
Condor Scheduler
Job
Condor Startd (started by glidein)
Grid WN
Grid Site

Convert a Condor job to be run with glideinWMS

Existing JDF file:

```
[testuserparag@cmssrv99 testjobs]$ cat testjob.jdf

universe = vanilla

executable = system-info.sh
output = $(cluster).$(process)
error = err.$(cluster).$(process)
log = log.$(cluster).$(process)

should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT

Requirements = ( Disk > 0 && (Arch == "INTEL" || Arch ==
"X86_64"))

Queue
```

Modified JDF to run using glideinWMS

```
[testuserparag@cmssrv99 testjobs]$ cat testjob.jdf

universe = vanilla

executable = system-info.sh
output = $(cluster).$(process)
error = err.$(cluster).$(process)
log = log.$(cluster).$(process)

should_transfer_files = YES
when_to_transfer_output = ON_EXIT_OR_EVICT

+DESIRED_Sites = “ress_ITB_INSTALL_cms-xen9”

Requirements = (stringListMember(GLIDEIN_Site,
DESIRED_Sites)) && (Disk > 0 && (Arch == "INTEL" || Arch ==
"X86_64"))

Queue
```

1. Add Desired_Sites (used by glideinWMS) to the JDF. This is used by glideinWMS to determine potential Grid site and is configurable.
2. Modify the requirements to support Desired_sites.

Pseudo Interactive Monitoring

- User can securely interact with the job while it is running on the worker node.
- Glidein starts a condor startd dedicated for pseudo interactive monitoring.

glideinWMS tool	System tool	Description
glidein_cat.py	cat	Concatenate file on the worker node
glidein_ls.py	ls	List the directory on the worker node
glidein_ps.py	ps	Show the process list on the worker node
glidein_top.py	top	Provide dynamic real-time view of tasks on the worker node
glidein_gdb.py	gdb	Run a debugger
glidein_interactive.py	-	Run any command on the worker node.