

The Glidein Service

Gideon Juve

juve@usc.edu



What are glideins?

- A technique for creating temporary, usercontrolled Condor pools using resources from remote Grid sites
 - 1. Grid jobs (called "glideins") are submitted to grid site using normal mechanisms (Globus, etc.)
 - 2. Glideins start Condor worker daemons on remote resources
 - 3. Glidein workers join user's Condor pool and are used to run application jobs

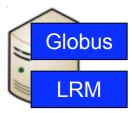


LOCAL SITE GRID SITE



User













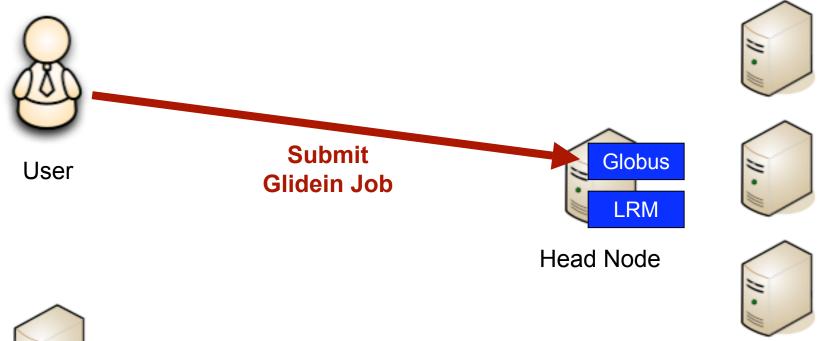
Condor Central Manager

To use glideins, the user runs a Condor central manager on a local machine that they control. This Condor pool will manage glidein resources allocated from a remote grid site.

Worker Nodes



LOCAL SITE GRID SITE





Condor Central Manager

The user submits a job to the grid site. This job (called a "glidein") will start Condor on the remote worker nodes. It is assumed that the Condor worker executables are pre-installed at the site.

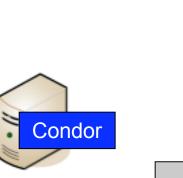
Worker Nodes



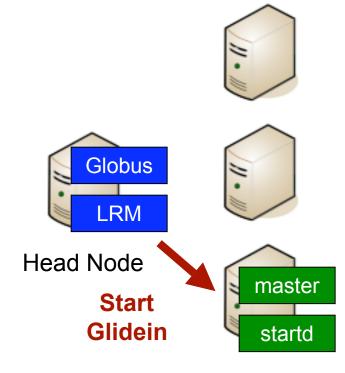
LOCAL SITE GRID SITE



User



Condor Central Manager



Worker Nodes

The glidein job configures and starts the Condor worker daemons on the grid site's worker nodes.

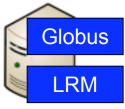


LOCAL SITE GRID SITE

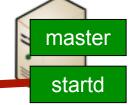


User





Head Node



Contact Central Manager



Condor Central Manager The Condor daemons contact the user's central manager and become part of the user's Condor pool.

Worker Nodes



LOCAL SITE GRID SITE

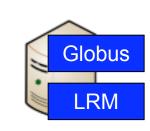


User





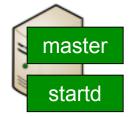
Condor Central Manager











Worker Nodes

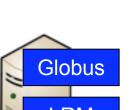
The user submits application jobs to their Condor pool. The jobs are matched with glidein resources.



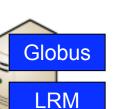
LOCAL SITE GRID SITE

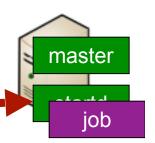


User



Head Node





Worker Nodes



Condor Central Manager

The application jobs are dispatched to the worker nodes for execution. Multiple application jobs can be executed on a single glidein resource.

Run Application Job



What are glideins good for?

- Running short jobs on the grid
 - Condor can dispatch jobs faster than Globus
- Bypassing site scheduling policies
 - Max submitted/running jobs
 - Priority for large jobs
- Avoiding competition for resources
 - Glideins reserve resources for multiple jobs
- Reducing load on head node/LRM
 - Fewer Globus jobmanagers polling for status



Other Approaches

- Advance Reservations
 - Ask the scheduler for exclusive access to resources
 - Not supported at many sites
 - Typically managed by site administrator (not users)
 - Users are charged a premium for resources
 - Unused reservations cannot be returned
- Task Clustering
 - Group multiple, independent jobs together
 - Can delay the release of some jobs
 - May reduce parallelism
- Not mutually exclusive
 - Can use a combination of techniques



Glidein Service

- GT4 grid service for running glideins
 - Automates the installation and configuration of Condor on grid site
 - Simplifies the complex setup and configuration required to run glideins
- Separate setup and provisioning steps
 - Create "sites" for remote installation and setup of Condor executables
 - 2. Create "glideins" for resource allocation



LOCAL SITE

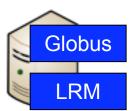




User



Globus Container



Head Node





Worker Nodes



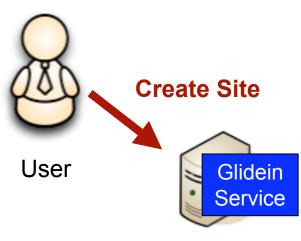
Condor Central Manager

The setup is similar to regular glideins. In addition to the Condor central manager, the user runs a Globus container that hosts the Glidein Service.

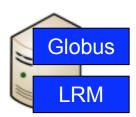


LOCAL SITE

GRID SITE



Globus Container













Condor Central Manager The user creates a new site by sending a request to the Glidein Service. The request specifies details about the grid site including: job submission information, file system paths, etc.





GRID SITE



User



Globus Head Node Container









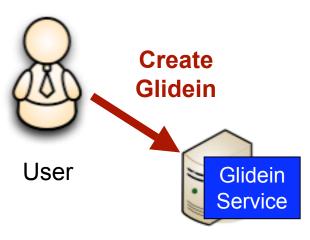
Condor Central Manager

The Glidein Service installs Condor executables on a shared file system at the grid site. The appropriate executables are automatically selected and downloaded from a central repository based on architecture, and operating system.

Worker Nodes

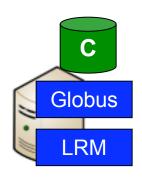


LOCAL SITE



Globus Container

GRID SITE



Head Node





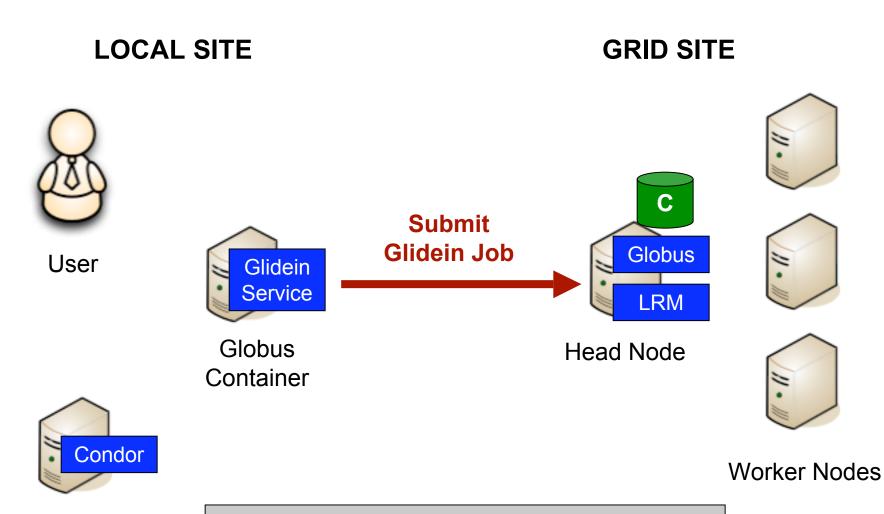




Condor Central Manager The user allocates resources by submitting glidein requests to the Glidein Service. The request specifies the number of hosts/CPUs to acquire and the duration of the reservation.

Worker Nodes





Condor Central Manager

The Glidein Service translates the user's request into a glidein job, which is submitted to the grid site.



LOCAL SITE



User

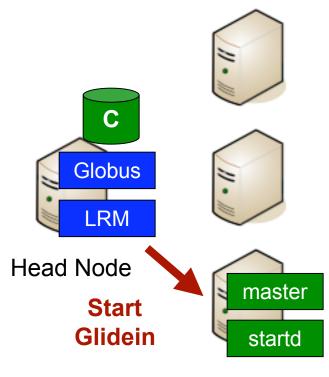


Globus Container



Condor Central Manager

GRID SITE



Worker Nodes

The glidein job starts Condor daemons on the worker nodes of the grid site.



LOCAL SITE

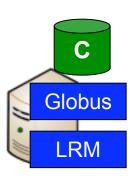
GRID SITE



User



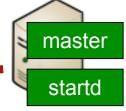
Globus Container



Head Node







Contact Central Manager

Worker Nodes

Condor Central Manager

Condor

The Condor daemons contact the user's Condor central manager and become part of the user's resource pool.



LOCAL SITE

GRID SITE



User

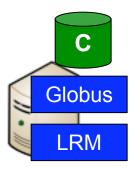




Condor Central Manager



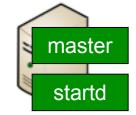
Globus Container



Head Node







Worker Nodes

The user submits application jobs to their Condor pool. The jobs are matched with available glidein resources.



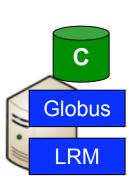
LOCAL SITE GRID SITE



User

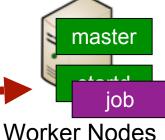


Globus Container



Head Node





Run Application Job

Condor Central Manager

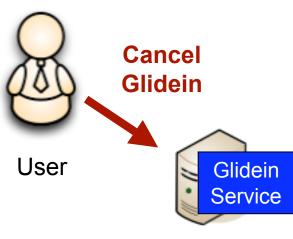
Condor

The application jobs are dispatched to the remote workers for execution.

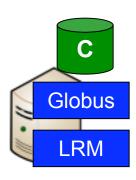


LOCAL SITE

GRID SITE



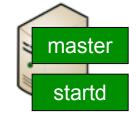
Globus Container









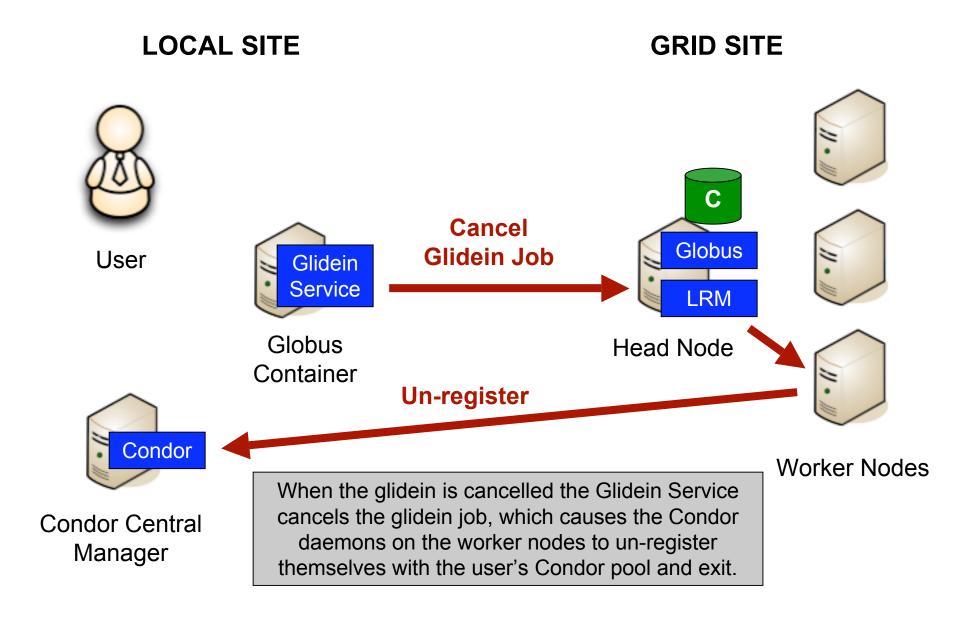


Worker Nodes



Condor Central Manager When the user is finished with their application they cancel their glidein request (or let it expire automatically).

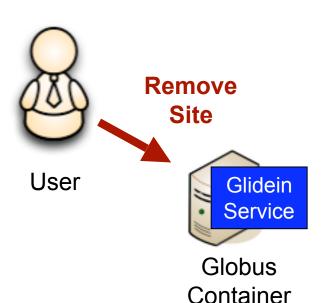


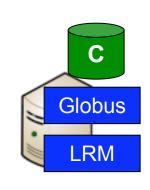




LOCAL SITE

GRID SITE







Head Node

Worker Nodes



Condor Central Manager The user can submit multiple glidein requests for a single site. When the user is done with the site they ask the Glidein Service to remove the site.





GRID SITE



User



Globus Container



Head Node





Worker Nodes



Condor Central Manager

Finally, the Glidein Service removes Condor from the shared file system at the grid site. This removes all executables, config files, and logs.



Features

- Auto-configuration
 - Detect architecture, OS, glibc -> Condor package
 - Determine public IP
 - Generate Condor config file
- Multiple interfaces
 - Command-line
 - SOAP
 - Java API
- Automatic resubmission of glideins
 - Indefinitely, N times, until date/time
- Notifications
- History tracking



Challenges

- Firewalls / Private IPs
 - Hinder communication between glideins and user's pool
 - Potential solutions
 - GCB
 - VPN?
 - Run central manager on head node (restrictions)
- Shared File System
 - Currently required for Glidein Service to stage Condor executables



Command Line Example

```
$ glidein create-site --site-name mercury --condor-version 7.0.0 \
          --environment GLOBUS LOCATION=/usr/local/qlobus-4.0.1-r3 \
          --install-path /home/ac/juve/glidein \
          --local-path /gpfs scratch1/juve/glidein \
          --staging-service 'gt2 grid-hg.ncsa.teragrid.org/jobmanager-fork' \
          --glidein-service 'gt2 grid-hg.ncsa.teragrid.org/jobmanager-pbs'
$ glidein list-site
ID
        NAME
                       CREATED
                                      LAST UPDATE
                                                      STATE
                                                                MESSAGE
3
                       08-27 09:16
                                      08-27 10:17
                                                      READY
                                                                Installed
        mercury
$ glidein create-glidein --site 3 --count 1 --host-count 1 --num-cpus 1 \
          --wall-time 30 --condor-host juve.isi.edu
$ glidein list-glidein
ID
     SITE
                  CONDOR HOST
                                         WTIME
                                                 CREATED
                                                                                     MESSAGE
                                  SLOTS
                                                              LAST UPDATE
                                                                           STATE
                                          30
     mercury (3) juve.isi.edu
                                                 08-27 09:23 08-27 10:24
                                                                           RUNNING
                                                                                      Running
$ condor status
Name
                   OpSys
                              Arch
                                     State
                                                Activity LoadAv Mem
                                                                      ActvtyTime
                                    Unclaimed Idle
                                                        0.100 1024 0+00:00:05
tg-c408.ncsa.terag LINUX
                              IA64
                     Total Owner Claimed Unclaimed Matched Preempting Backfill
           IA64/LINUX
                                                                              0
                                       0
                                                          0
                                                                              0
               Total
```