A user perspective
Outlook

- Grid computing overview
- The pilot paradigm
- Introducing Condor glideins
- glideinWMS description
- glideinWMS in real life
- Conclusions
Portrait of a scientist

• Needs many computing cycles to analyze his data
  – More than can get from a personal desktop
• Wants to spend most of his time thinking about the scientific problems
  – Computing is just a tool
Portrait of the Grid

• Resources grouped in independent pools
  – Each with its own set of rules

• Resources in different pools configured differently
  – Users expected to adapt
We have a problem!

- Scientists are forced to spend a significant amount of time thinking about computing
  - And every time a new site is added, the process starts again
- Time spent on computing problems is subtracting time available for scientific thinking!
Let's make the Grid uniform

... by creating an overlay over the Grid sites
- Hiding differences between sites
- Making the Grid look as a single, uniform pool
The pilot paradigm

- A possible solution
  - Use pilot jobs to create the overlay
The pilot paradigm (continued)

- Never send user jobs directly
  - Send pilot jobs to create the overlay, instead
- When a pilot lands on a Grid worker node
  - Validates Grid resource
  - Prepares the environment
  - Pulls a user job
- Hides Grid heterogeneity
  - Users see a fairly uniform computing pool
Concor glideins

http://www.cs.wisc.edu/condor/

- Condor is based on a distributed architecture
- Condor glideins are Grid jobs that start regular Condor daemons
  - i.e. they are pilots implemented with Condor

Submit job

Get result

http://www.cs.wisc.edu/condor/
Submitting glideins

- Condor provides only a basic command line glidein submission tool
  - Good for trying out glideins
  - But not meant to be used as a glidein factory
- A few groups developed glidein factories
  - CDF has the CDF-specific GlideCAF
  - USCMS@FNAL is developing the glideinWMS
Introducing the glideinWMS

http://www.uscmsg.org/SoftwareComputing/Grid/WMS/glideinWMS/

• An autonomous glidein submission system
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins
• glideinWMS composed of six logical pieces:
  – A Condor central manager (collector + negotiator)
  – One or more Condor submit machines
  – A glideinWMS collector
  – One or more VO frontends
  – One or more glidein factories
  – The glideins
glideinWMS architecture

• glideinWMS composed of six logical pieces:
  – A Condor central manager (collector + negotiator)
  – One or more Condor submit machines
  – A glideinWMS collector
  – One or more VO frontends
  – One or more glidein factories
  – The glideins
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins
glideinWMS architecture

- glideinWMS composed of six logical pieces:
  - A Condor central manager (collector + negotiator)
  - One or more Condor submit machines
  - A glideinWMS collector
  - One or more VO frontends
  - One or more glidein factories
  - The glideins
Condor handles user jobs

- A glidein Condor pool is still a Condor pool
  - Just a very dynamic one
- All Condor features available
  - ClassAds
  - Fair share
  - Group quotas
- Users really don't know about the glideinWMS
Glidein submission

• glideinWMS processes are responsible **only** for startd startup
  – A glidein just configures and starts it
  – Once started, startd has full control

• Glidein factory administrator handles Grid heterogeneity
Glidein submission

• Based on the principle of constant pressure
  – As long as there are enough waiting jobs in the queue, a fixed number of glideins are kept at each suitable Grid site

• Works nicely for systems with lots of waiting jobs
  – Will waste resources on seldom used systems
Glidein submission

- Glidein submission is a collaborative work
  - VO frontend decides how many glideins to submit
  - Glidein factory actually does the submission
  - WMS collector is used for message passing
- Condor-G used for submission to Grid sites
Security considerations

- GlideinWMS **requires** security over the wire
  - WAN network connections cannot be blindly trusted!
- All network traffic features integrity checks
  - Prevents man-in-the-middle attacks
- GSI authentication (X509 certificates/proxies) used for all interactions with Condor daemons over the network
  - Based on access lists (like gridmap file, but with regex)
  - Attributes not used right now
    - But will probably need it
Security considerations

• Startd not running as a privileged user
  – Cannot change UID by itself when starting user job
  – Malicious user job could hijack the startd if running under the same UID

• Condor interfaced to gLEexec
  – gLEexec allows to change UID given user proxy
  – Users protected from other users running on the same node
  – Startd protected from the user job
Proxy handling

- A single identity used to submit all the glideins
  - Should have a pilot role
- Condor ships user proxy to worker node
  - User can use it access other resources
- Proxy lifetime management and renewal not handled by glideinWMS
  - Users expected to do it
User job monitoring

- **Good monitoring a must for most users**
- Condor provides a plethora of monitoring tools
  - Most useful are `condor_q` and `condor_status`
  - Third parties provide additional Condor monitoring tools
- glideinWMS provides tools for pseudo-interactive monitoring
  - `ls`, `cat`, `top` on the worker nodes
- The glidein factory also maintains a basic Web based graphical view
  - plus machine readable XML and rrd data
glideinWMS monitoring

• Good monitoring a must for most administrators, too

• Condor-G provides some tools
  – Mostly condor_q

• The glidein factory maintains a rich Web based graphical view
  – plus machine readable XML and rrd data

• Glideins return comprehensive logs
  – Useful for low level debugging
  – But require some expertise to browse though
Status of glideinWMS

• Should be usable out of the box for most users
  – CMS is using it since v1.1

• Still in active development phase
  – More monitoring
  – More automated error checking
  – More automated error recovery
  – Better integration with other systems

• Condor also an evolving product
Glidein deployments in HEP

- CMS using glideins for production jobs at FNAL
  - And across all seven T1s
- CMS used them for analysis jobs in CCRC08
  - Across 40 T2s
- CDF and MINOS using them for user analysis
glideinWMS in numbers

• Deployed systems
  – CMS@FNAL stable 3k glideins for the past 6 months
  – CMS@CCRC up to 4k glideins over 40 sites globally
  – CDF average 2k glideins with 100s of users for past 2 years (by using the GlideCAF)

• glideinWMS Tested on a dedicated test pool, scaled without major problems to
  – 10k glideins at any time
  – 100k user jobs queued
glideinWMS support

- Developed by USCMS team at Fermilab
- Released under the FermiTools license
  - A modified BSD license
    http://fermitools.fnal.gov/about/terms.html
- Support to non-CMS users available on best-effort bases
glideinWMS contact info

GlideinWMS home page:
http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

Condor home page:
http://www.cs.wisc.edu/condor/

email: sfiligoi@fnal.gov
Conclusions

- The average scientist should not be exposed directly to the Grid
  - Computing related overhead too high
- Glideins can hide the Grid complexity and make it look as a uniform computing pool
- Several HEP collaborations are happily using glideins in the real life
  - Other communities could benefit as well
  - glideinWMS is an easy path there