glideinWMS Frontend Installation

Part 1 – Condor Installation

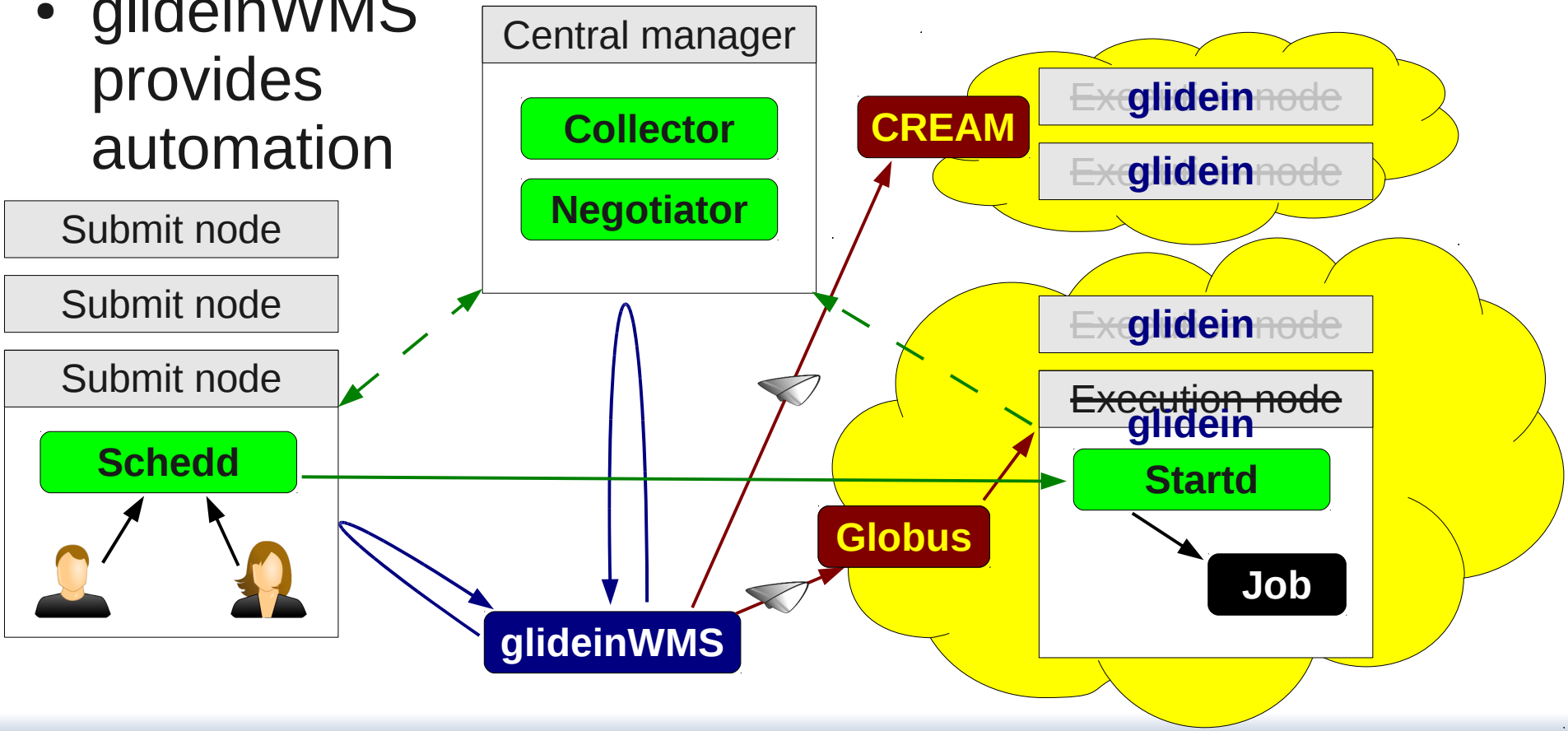
by Igor Sfiligoi (UCSD)

Overview

- Introduction
- Planning and Common setup
- Central Manager Installation
- Submit node Installation

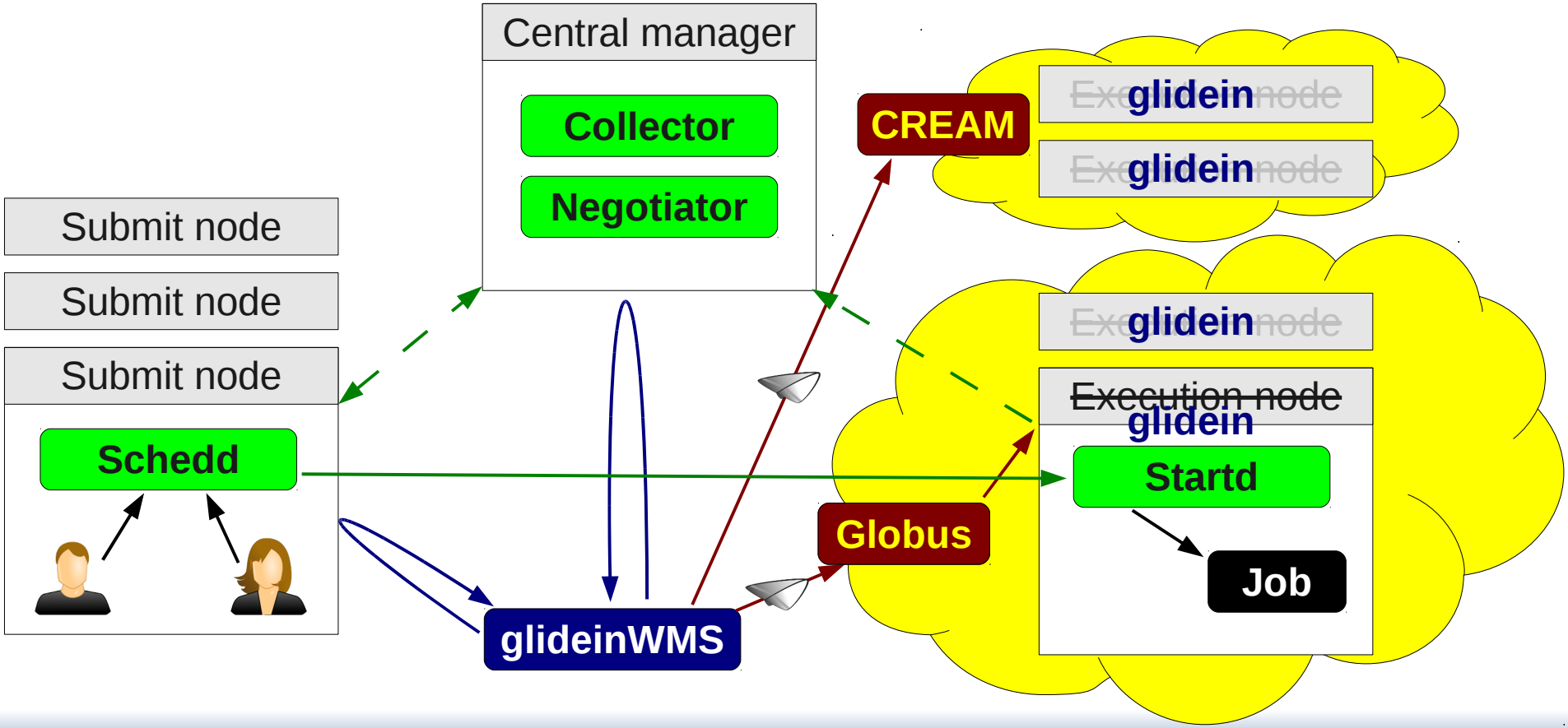
Refresher - Glideins

- A glidein is just a properly configured Condor execution node submitted as a Grid job
  - glideinWMS provides automation



Refresher - Glideins

- The glideinWMS triggers glidein submission
  - The “regular” negotiator matches jobs to glideins



Bottom line

Condor is king!

(glideinWMS just a small layer on top)

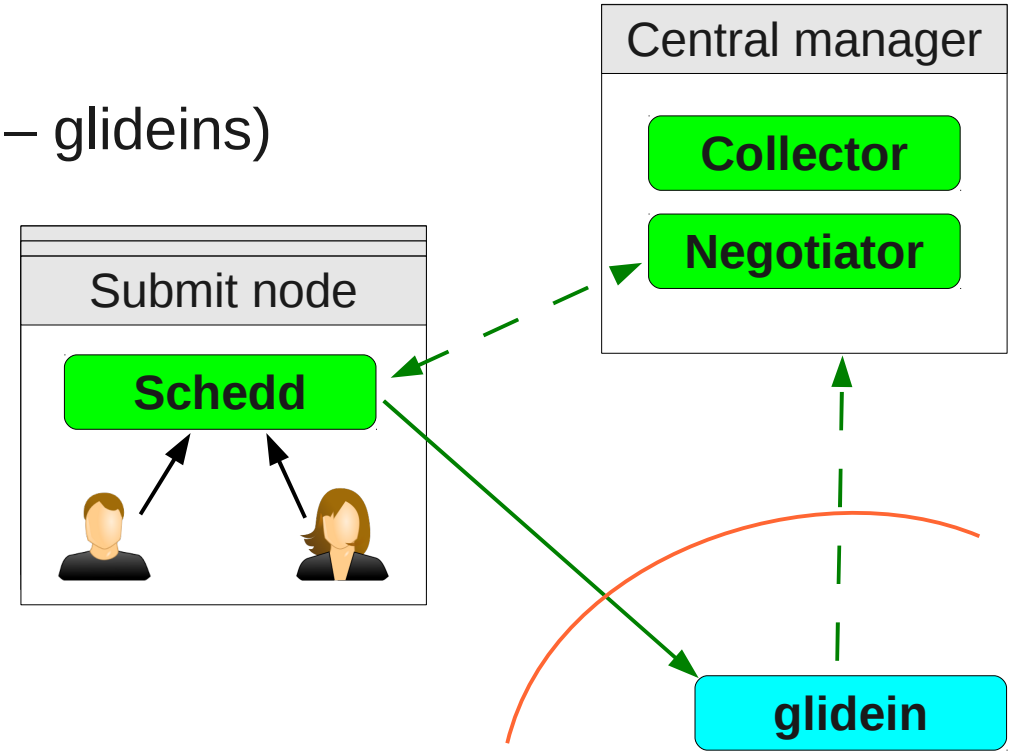
Condor installation

- Proper Condor installation and configuration the most important task
  - Condor will do most of the work
  - … and is thus the most resource hungry
- GlideinWMS installation almost an afterthought
  - Although it does require proper security config of Condor
  - GlideinWMS installation proper will be described in a separate talk

Planning
and
Common setup

Refresher - Condor

- Two main node types
  - Submit node(s)
  - Central manager
  - (execute nodes are dynamic – glideins)
- **Public TCP/IP networking needed**
- GSI used for network security



Planning the setup

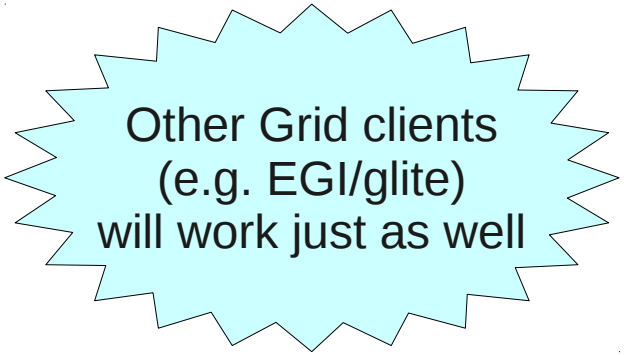
- In theory, all Condor daemons can be installed on a single node
- However, if at all possible, **put Central Manager on a dedicated node**
  - i.e. do not use it as a submit node, too
  - **Both for security and stability reasons**
- You may want/need more than one submit node
  - Depends on expected use and available HW
  - You do need at least one, though

Common system considerations

- Condor is supported on a wide variety of platforms
  - Including Linux (e.g. RHEL5), MacOS and Windows
  - Linux recommended in OSG (and assumed in the rest of talk)
- GSI security requires
  - Host or service certificate
  - CAs & CRLs
    - Typically delivered via OSG RPMs (but other means acceptable)
      https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallCertAuth
  - Full Grid Client software recommended (for ease of ops)
    https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallOSGClient

OSG Grid Client

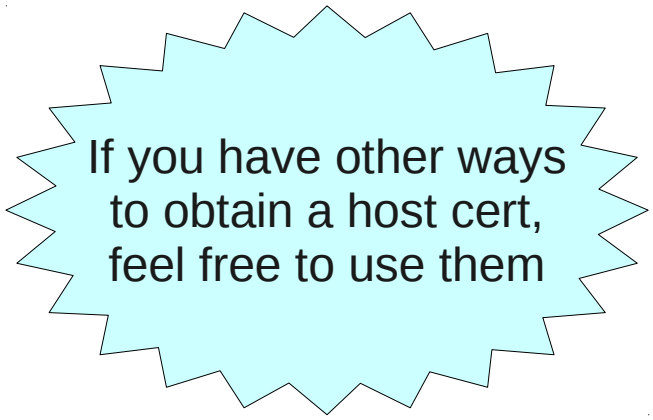
- Requires RHEL5-compatible Linux
  - RHEL6 support promised for early 2012
- Procedure in a nutshell
  - Add EPEL and OSG RPM repositories to sys conf.
  - `yum install osg-ca-certs`
  - `yum install osg-client`
  - Enable CRL fetching crontab



https://twiki.grid.iu.edu/bin/view/Documentation/Release3/InstallOSGClient

Requesting a host certificate

- OSG provides a script to talk to DOEGrids
  https://twiki.grid.iu.edu/bin/view/Documentation/Release3/GetHostServiceCertificates
- Procedure in a nutshell
  - Install OSG client
  - `yum install osg-cert-scripts`
  - `cert-request` …
  - Wait for email
  - `cert-retrieve` …
  - cp into /etc/grid-security/



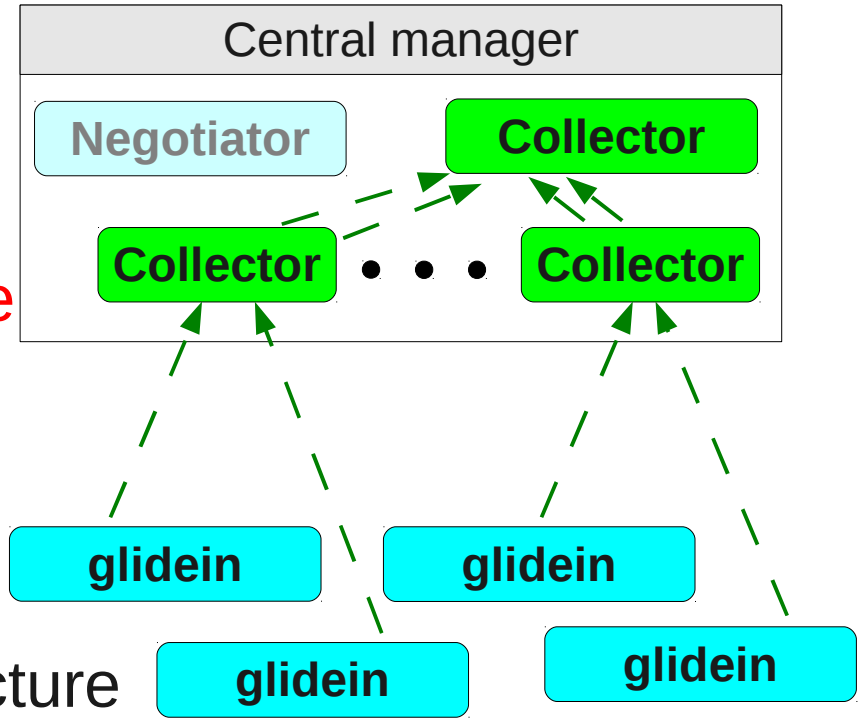
Condor
Central Manager

Refresher - Central Manager

- Two (groups of) processes
  - Collector
  - Negotiator
- The Collector defines the Condor pool
  - Knows about all the glideins it owns
  - Knows about all the schedds
- The Negotiator does the matchmaking
  - Decides who gets what resources

Condor Collector – considerations

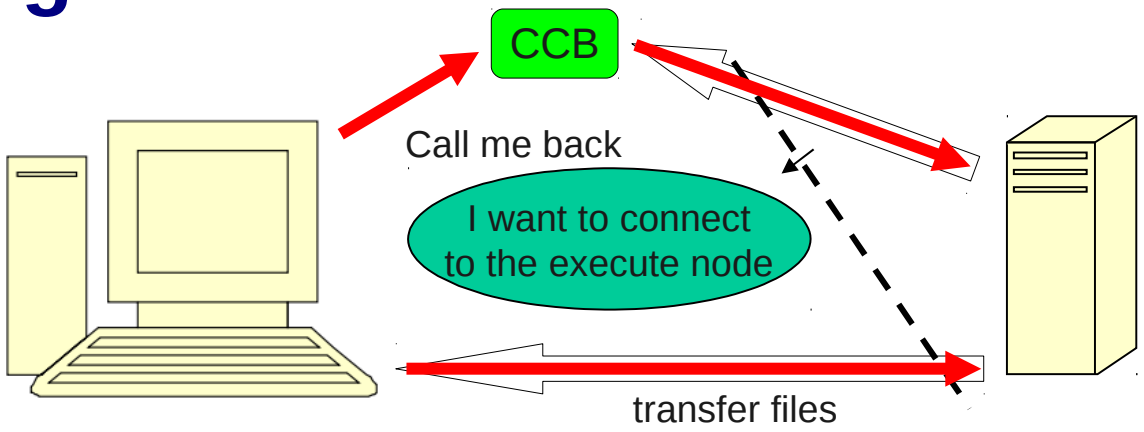
- The Collector is the **repository of all knowledge**
  - All other daemons report to it
  - Including the glideins, who get its address at run-time
- Must process lots of info
  - One update every 5 mins from each and every daemon
  - With strong security → expensive
- Typically deployed as a **tree of collectors**
  - All security handled in leafs
  - Top one still has the complete picture



CCB – An additional cost

- The Condor collectors are also acting as CCBs
  - Each glidein will open 5+ long-lived TCP sockets
- Make sure you have enough file descriptors
  - Default OS limit is 1024 per process
- Plan on having
  one CCB per 100 glideins

Leafs in the
tree of collectors



High availability
(theory)

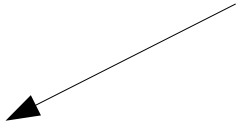
- Central manager can be a single point of failure
  - If it dies, the Condor pool dies with it!
- To avoid this, one can deploy multiple CMs
  - All daemons will advertise to 2 (or more) Collectors

Currently not supported by glideinWMS

  - All CMs will have the same view of the world
- There can only be one Negotiator, though
  - One negotiator will be Active, all others in standby
  - More details on Condor man page
    http://www.cs.wisc.edu/condor/manual/v7.6/3_11High_Availability.html#SECTION004112000000000000000

Hardware needs

- Tree of collectors spreads the load over multiple processes
  - So several CPUs come handy
- Negotiator single threaded
  - Will benefit from fast CPU
- Memory usage not terrible
  - O(100k) per glidein to store ClassAds
  - Concrete CMS example: 25k glideins ~ 6G memory
- Negligible disk IO



System considerations

Minimize risk due to Condor bugs

- Does **not** need to run as root (although it can)
  - Make sure the host cert is readable by that user
- Must be on the **public IP network**
  - Each collector listens on its own well defined port, must be reachable by all glideins (WAN)
  - Negotiator has a dynamic list port, must be reachable by submit nodes (schedds)
- Will use a large number of network sockets
  - **Will overwhelm most firewalls**
  - Consider disabling stateful firewalls (e.g. iptables)

Must open firewall at least for these

Security considerations

- Cannot be firewalled → endpoint security
  - GSI security used (i.e. x509 certs) for networking
  - Limit administrative rights to local users (FS auth)
- The Collector is **central trust point** of the pool
  - The DNs of all other daemons are whitelisted here, including:
    - Schedds
    - Glideins (i.e. pilot proxies)
    - Clients (e.g. glideinWMS Frontend)

Installing the CM

- Two major burdens (for basic install)
  - Collector tree
  - Security setup
- The glideinWMS installer helps with both
  - Starting from Condor tarball
  - As any user (e.g. as non-root)
  - Highly recommended
- RPM install also an option
  - Easy to keep up-to-date (i.e. yum update)
  - But you will need to configure by hand
  - And will run as root

Easy-to-use update cmdline tool available, too

Unless you hack the startup script

Collector tree setup

- In a nutshell
  - For each secondary collector:
    - Tell Master to start a collector on different port
    - repeat
  - Forward ClassAds to main Collector

```
...
COLLECTORxxx = $(COLLECTOR)
COLLECTORxxx_ENVIRONMENT = "_CONDOR_COLLECTOR_LOG=$(LOG)/CollectorxxxLog"
COLLECTORxxx_ARGS = -f -p yyyy
DAEMON_LIST = $(DAEMON_LIST) COLLECTORxxx
...

# forward ads to the main collector
# (this is ignored by the main collector, since the address matches itself)
CONDOR_VIEW_HOST = $(COLLECTOR_HOST)
```

x N

Security setup (1)

- In a nutshell
  - Configure basic GSI (i.e. point to CAs and host cert)
  - Set up authorization (i.e. switch to whitelist)
  - Whitelist all DNs
  - Enable GSI
- DN whitelisting a bit annoying
  - Must be done in two places
    - in condor_config, and
    - in condor_mapfile
  - glideinWMS provides a cmdline tool

Security setup (2)

```
# condor_config.local
# Configure GSI
CERTIFICATE_MAPFILE=/home/condor/glidecondor/certs/condor_mapfile
GSI_DAEMON_TRUSTED_CA_DIR=/etc/grid-security/certificates
GSI_DAEMON_CERT = /home/condor/.globus/hostcert.pem
GSI_DAEMON_KEY  = /home/condor/.globus/hostkey.pem

# Force whitelisting
DENY_WRITE = anonymous@*
DENY_ADMINISTRATOR = anonymous@*
DENY_DAEMON = anonymous@*
DENY_NEGOTIATOR = anonymous@*
DENY_CLIENT = anonymous@*
ALLOW_ADMINISTRATOR = $(CONDOR_HOST)
ALLOW_WRITE = *
USE_VOMS_ATTRIBUTES = False # use only pilot DN, not FQAN

# list all DNs
...
GSI_DAEMON_NAME=$(GSI_DAEMON_NAME),DNxxx
...

# enable GSI
SEC_DEFAULT_AUTHENTICATION_METHODS = FS,GSI
SEC_DEFAULT_AUTHENTICATION = REQUIRED
SEC_DEFAULT_ENCRYPTION = OPTIONAL
SEC_DEFAULT_INTEGRITY = REQUIRED
# optionally, relax client and read settings
```

```
# condor_mapfile
...
GSI "^DNxxx$" UIDxxx
...
GSI (.*) anonymous
FS (.*) \1
```

x N

Also enable local auth

Installing with Q&A installer

~/glideinWMS/install$ ./glideinWMS_install
...
Please select: **4**
[4] User Pool Collector
...
Where do you have the Condor tarball? **/home/condor/Downloads/condor-7.6.4-x86_rhap_5-stripped.tar.gz**
Where do you want to install it?: [/home/condor/glidecondor] **/home/condor/glidecondor**
If something goes wrong with Condor, who should get email about it?: **me@myemail**
Do you want to split the config files between condor_config and condor_config.local?: (y/n) [y] **y**
...
Do you want to get it from VDT?: (y/n) **y**
Do you have already a VDT installation?: (y/n) **y**
Where is the VDT installed?: **/etc/osg/wn-client**
...
Will you be using a proxy or a cert? (proxy/cert) **cert**
Where is your certificate located?: **/home/condor/.globus/hostcert.pem**
Where is your certificate key located?: **/home/condor/.globus/hostkey.pem**
My DN = 'DN1'
...
DN: **DNxxx**
nickname: [condor001] **uidxxx**
Is this a trusted Condor daemon?: (y/n) **y**

} x N

You can also add the DNs as an independent step

...
DN:
How many slave collectors do you want?: [5] **200**
What name would you like to use for this pool?: [My pool] **MyVO**
What port should the collector be running?: [9618] **9618**

Maintenance

- If you need to add more DNs, use
  - cmdline tool `glidecondor_addDN`

```
~/glideinWMS/install$ ./glidecondor_addDN -daemon "DN of Schedd A" "DNA" UIDA
Configuration files changed.
Remember to reconfig the affected Condor daemons.
```

- To upgrade the Condor binaries, use
  - cmdline tool `glidecondor_upgrade`

```
~/glideinWMS/install$ ./glidecondor_upgrade ~/Downloads/condor-7.6.5-x86_rhap_5-stripped.tar.gz
Will update Condor in /home/condor/glidecondor
..
Creating backup dir
Putting new binaries in place
Finished successfully

Old binaries can be found in /home/condor/glidecondor/old.120102_13
```

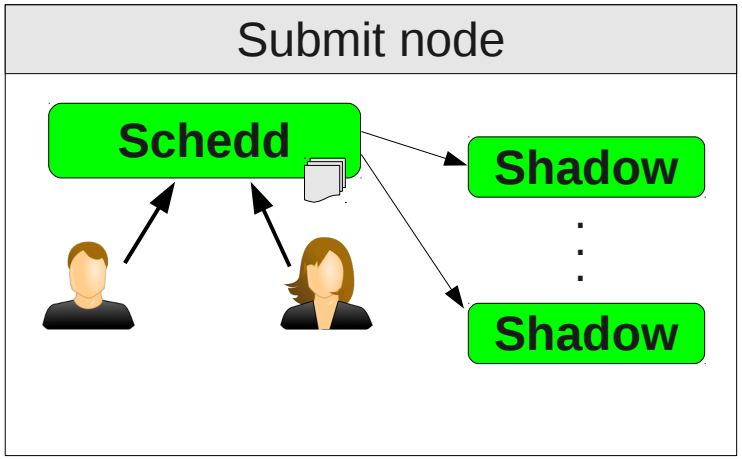
Starting Condor

- The installer will start Condor for you, but you still should know how to stop and start it by hand
- To start condor, run:
  `~/glidecondor/start_condor.sh`
- To stop Condor, use
  `condor_off -daemon master`
- Finally, to force Condor to re-read the config:
  `~/glidecondor/sbin/condor_reconfig`

Condor
Submit node(s)

Refresher - Submit node(s)

- Submit node defined by the schedd
  - Which holds user jobs
- Shadows will be started as the jobs are matched to glideins
  - One per running job
- At least one submit node is needed
  - But there may be many



Network use

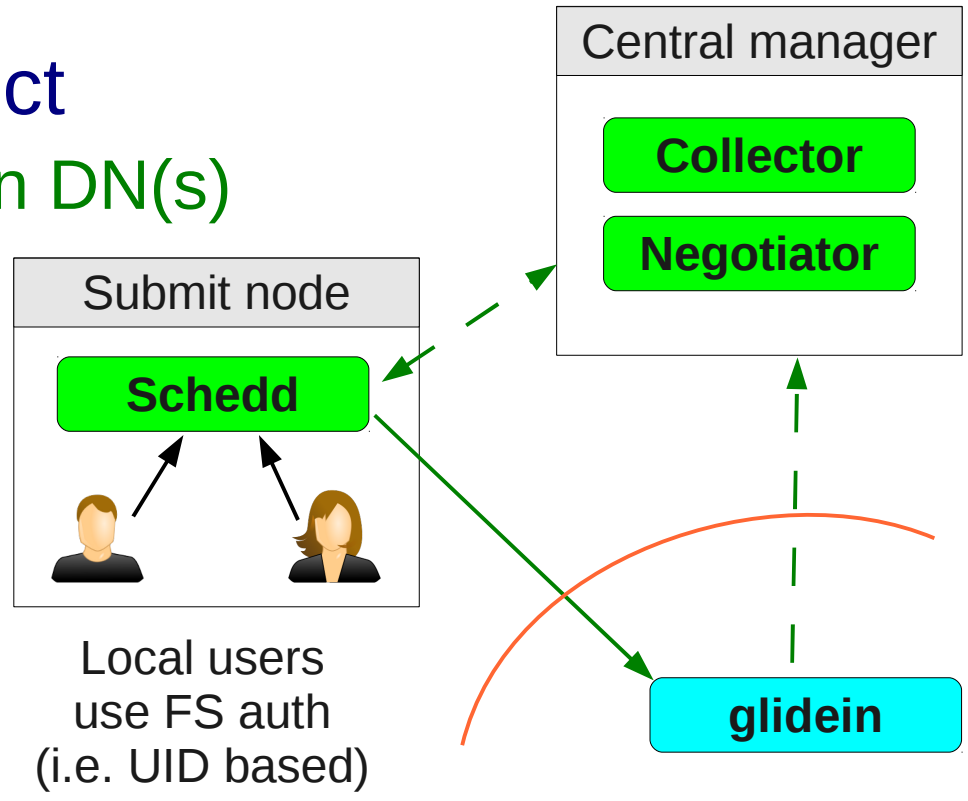
- Glideins must contact the submit node in order to run jobs
  - Both with standard protocol and CCB
- Each shadow normally uses 2 random ports
  - Not firewall friendly
  - Can be a problem over O(10k) jobs
- Newer versions of Condor support "**shared port daemon**"
  - Listens on a single port
  - Forwards the sockets to the appropriate local process

Although firewalls can get overwhelmed anyhow (see CM slides)

Does **not** reduce number of sockets

Security considerations

- Like with CM, must use endpoint security
- Schedd and CM must whitelist each other
  - Certificate DN based
- AuthZ with glideins indirect
  - No need to whitelist glidein DN(s)
  - Collector trusts glidein, Schedd trusts Collector
- Schedd also must whitelist any clients (e.g. VO Frontend)
  - Only startds can use indirect AuthZ



http://research.cs.wisc.edu/condor/manual/v7.6/3_3Configuration.html#param:SecEnableMatchPasswordAuthentication

Hardware needs

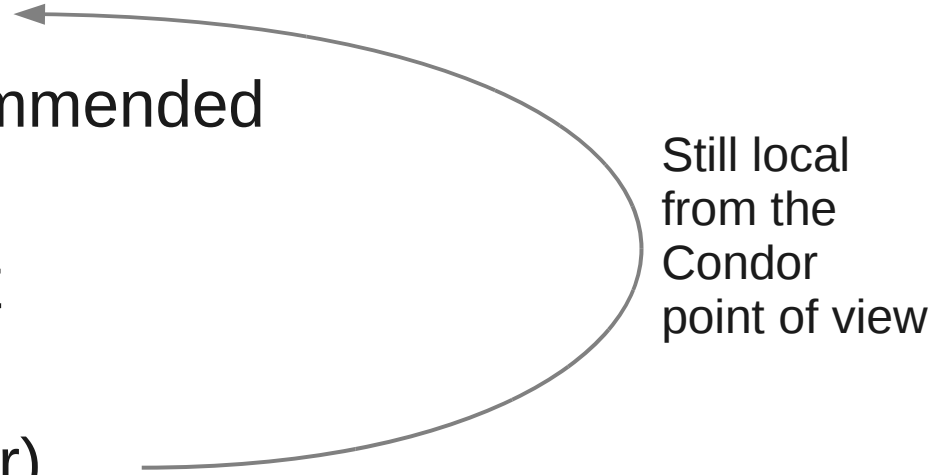
- **Submit node is memory hungry**
  - 1M per running jobs due to shadows
  - O(10k) per job in queue for ClassAds
- Schedd can use a fast CPU (single threaded)
  - Shadows very light CPU users
- Jobs may put substantial IO load on HDD
  - Depends on how much data is being produced
  - Depends how short are the jobs
- And the above is just for Condor
  - VO may have portal software
  - or actual interactive users

Actual need depends on how many additional VO attributes used

Make sure the remaining HW is adequate for these

User account considerations

- Users must be able to launch **condor_submit** locally on the submit node
  - Remote submission not recommended (and disabled by default)
- VO must decide how to do it
  - SSHd (i.e. interactive use)
  - Portal (e.g. CMS CRABServer)
- Will need one UID per user
  - Non-UID based auth possible, but not recommended (but not supported out of the box)



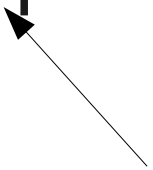
No need to create user accounts before Installing Condor, but do plan for it

Schedd is a superuser

- Schedd must run as root
  (euid==0, even as it drops ruid to “condor”)
  - So it can switch UID as needed
  - To access user files
  - Same for shadows (but ruid set to job user)
- Host cert thus must be owned by root

Installing the submit node

- Two major burdens (for basic install)
  - Shared port daemon
  - Security setup
- The glideinWMS installer helps with both
  - Starting from Condor tarball
  - Should be run as root
  - Highly recommended
- RPM install also an option
  - Easy to keep up-to-date (i.e. yum update)
  - But you will need to configure by hand



Easy-to-use update cmdline tool available, too

Shared port daemon

- Not enabled by default in Condor
- In a nutshell
  - Pick a port for it
  - Enable it
  - Add it to the list of Daemons to start

```
# condor_config.local

# Enable shared_port_daemon
SHARED_PORT_ARGS = -p 9615
USE_SHARED_PORT = True
DAEMON_LIST = $(DAEMON_LIST) SHARED_PORT
```

Security setup (1)

- In a nutshell
  - Configure basic GSI (i.e. point to CAs and host cert)
  - Enable match authentication
  - Set up authorization (i.e. switch to whitelist)
  - Whitelist all DNs
  - Enable GSI
- DN whitelisting a bit annoying
  - Must be done in two places
    - in condor_config, and
    - in condor_mapfile
  - glideinWMS provides a cmdline tool

And is a regexp here!

Security setup (2)

```
# condor_config.local
# Configure GSI
CERTIFICATE_MAPFILE=/opt/glidecondor/certs/condor_mapfile
GSI_DAEMON_TRUSTED_CA_DIR=/etc/grid-security/certificates
GSI_DAEMON_CERT = /etc/grid-security/hostcert.pem
GSI_DAEMON_KEY  = /etc/grid-security/hostkey.pem

# Enable match authentication
SEC_ENABLE_MATCH_PASSWORD_AUTHENTICATION = TRUE

# Force whitelisting
DENY_WRITE = anonymous@*
… # see CM slides for details

# list all DNs
...
GSI_DAEMON_NAME=$(GSI_DAEMON_NAME),DNxxx
...

# enable GSI
SEC_DEFAULT_AUTHENTICATION_METHODS = FS,GSI
SEC_DEFAULT_AUTHENTICATION = REQUIRED
SEC_DEFAULT_ENCRYPTION = OPTIONAL
SEC_DEFAULT_INTEGRITY = REQUIRED
# optionally, relax client and read settings
```

```
# condor_mapfile
...
GSI "^DNxxx$" UIDxxx
...
GSI (.*) anonymous
FS (.*) \1
```

x N

Also enable local auth

Network optimization settings

- Since glideins often behind firewalls
  - The glidein Startd setup optimized to avoid incoming connections and UDP
- The Schedd must also play along

```
# condor_config.local

# Reverse protocol direction
STARTD_SENDS_ALIVES = True
# Avoid UDP
SCHEDD_SEND_VACATE_VIA_TCP = True
```

Installing with Q&A installer

```
~/glideinWMS/install$ ./glideinWMS_install
...
Please select: 5
[5] User Schedd
…
Which user should Condor run under?: [condor] condor
Where do you have the Condor tarball? /root/condor-7.6.4-x86_rhap_5-stripped.tar.gz
Where do you want to install it?: [/home/condor/glidecondor] /opt/glidecondor
If something goes wrong with Condor, who should get email about it?: me@myemail
Do you want to split the config files between condor_config and condor_config.local?: (y/n) [y] y
...
Do you want to get it from VDT?: (y/n) y
Do you have already a VDT installation?: (y/n) y
Where is the VDT installed?: /etc/osg/wn-client
Will you be using a proxy or a cert? (proxy/cert) cert
Where is your certificate located?: /etc/grid-security/hostcert.pem
Where is your certificate key located?: /etc/grid-security/hostkey.pem
My DN = 'DN1'
...
DN: DNxxx
nickname: [condor001] uidxxx
Is this a trusted Condor daemon?: (y/n) y
...
DN:
What node is the collector running (i.e. CONDOR_HOST)?: collectornode.mydomain
Do you want to enable the shared_port_daemon?: (y/n) y
What port should it use?: [9615] 9615
How many secondary schedds do you want?: [9] 0
```

x N

You can also add the DNs as an independent step

Maintenance

- If you need to add more DNs, use
  - cmdline tool `glidecondor_addDN`

```
~/glideinWMS/install$ ./glidecondor_addDN -daemon "DN of Schedd A" "DNA" UIDA
Configuration files changed.
Remember to reconfig the affected Condor daemons.
```

Do not use -daemon for client's DN

- To upgrade the Condor binaries, use
  - cmdline tool `glidecondor_upgrade`

```
~/glideinWMS/install$ ./glidecondor_upgrade ~/Downloads/condor-7.6.5-x86_rhap_5-stripped.tar.gz
Will update Condor in /home/condor/glidecondor
..
Creating backup dir
Putting new binaries in place
Finished successfully

Old binaries can be found in /home/condor/glidecondor/old.120102_13
```

Starting Condor

- The installer will start Condor for you, but you still should know how to stop and start it by hand
- The installer has created an init.d script for you `/etc/init.d/condor start|stop`
- To force Condor to reload its config, still use `/opt/glidecondor/sbin/condor_reconfig`

Fine tunning

Fine tunning

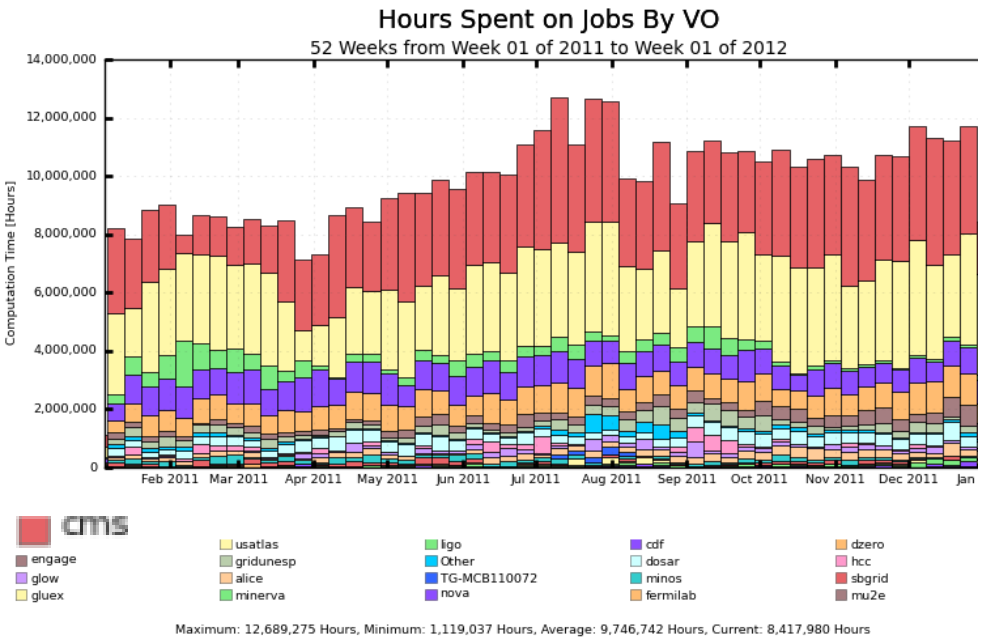
- The previous slides provide only basic setup
  - Although the glideinWMS does some basic tunning
- You will likely want to tune the system further
  - Proper limits in the submit node
  - Default job attributes
  - Sanity checks
  - Priority tunning
- Not part of this talk
  - Will go into details tomorrow

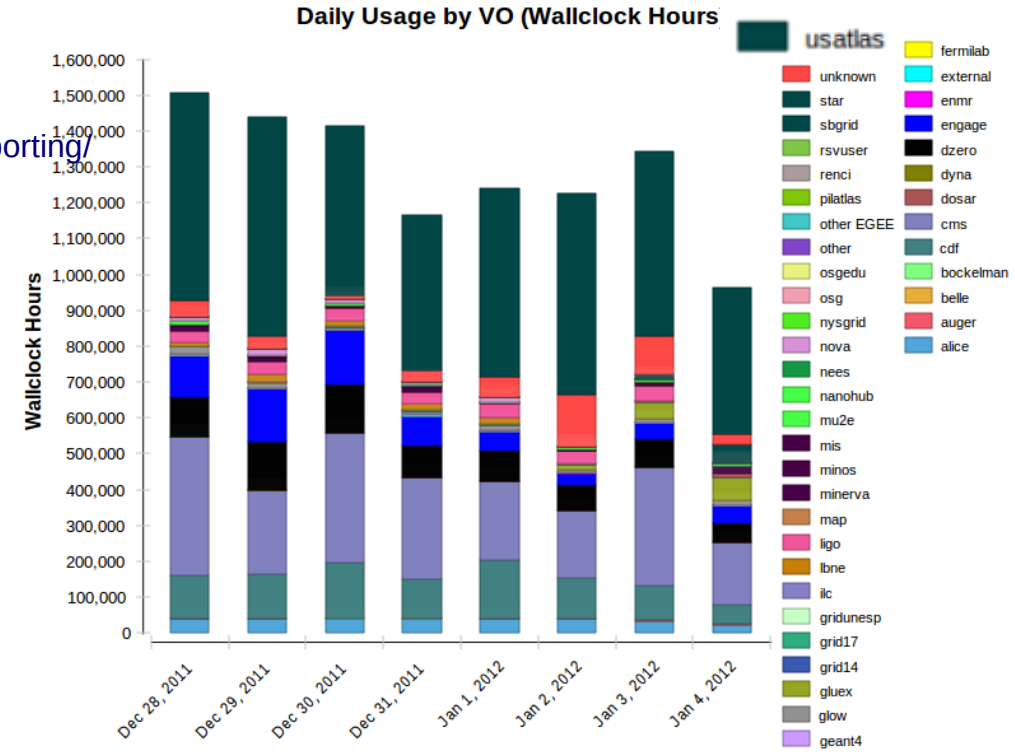
Integration with OSG Accounting

OSG Accounting

- OSG tries to keep accurate accounting information of who used what resources
  - Using GRATIA

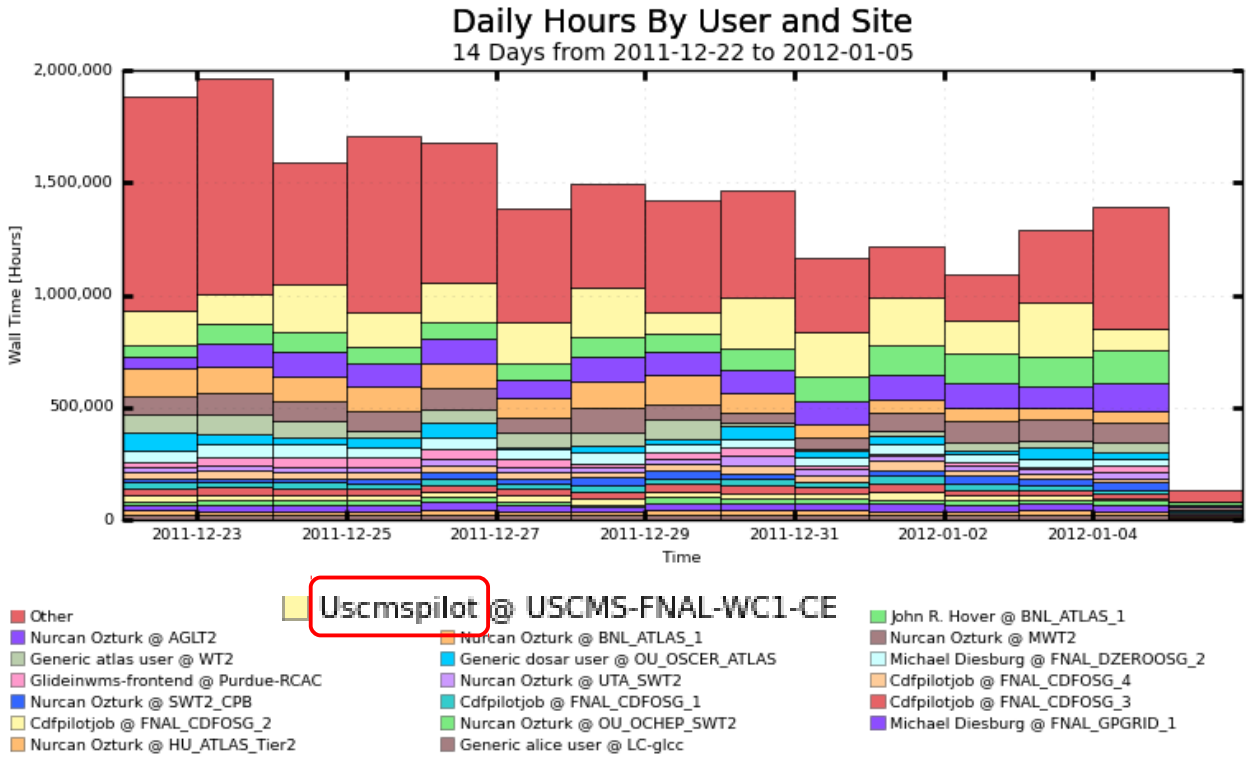
https://twiki.grid.iu.edu/twiki/bin/view/Accounting/WebHome
http://gratia-osg-prod-reports.opensciencegrid.org/gratia-reporting/





Per-user accounting

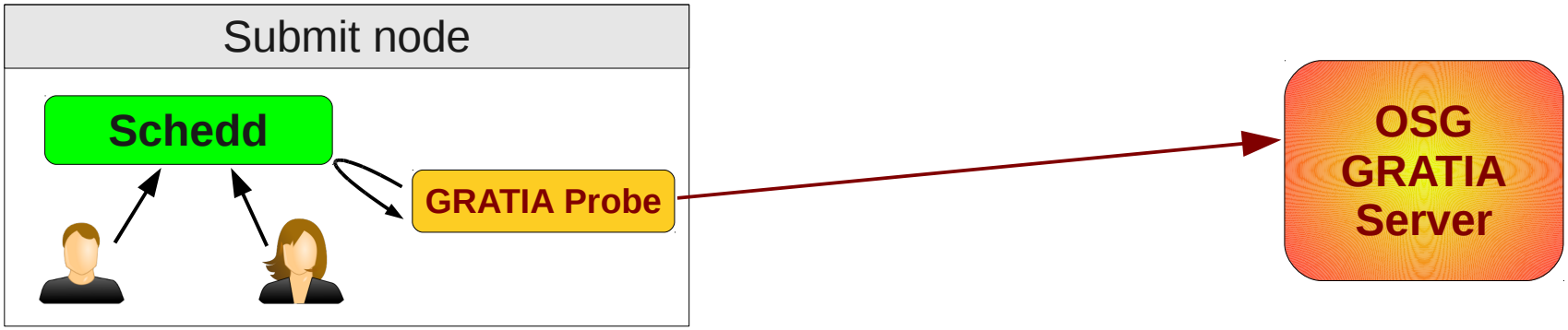
- OSG has per-user accounting, too
  - With glideins, this level of detail lost
  - Only pilot proxy seen by OSG (sites)



The glidein GRATIA probe

- OSG thus asks glidein operators to install a dedicated probe alongside the glidein schedd(s)
  - Which will provide per-user accounting info to the OSG GRATIA server
  - Optimized for use with OSG glidein factory

https://twiki.grid.iu.edu/bin/view/Accounting/ProbeConfigGlideinWMS



Installing the GRATIA probe

- In a nutshell
  - Register submit node with GOC
  - Tweak condor config
  - yum install gratia-probe-condor
  - Configure GRATIA

https://twiki.grid.iu.edu/bin/view/Accounting/ProbeConfigGlideinWMS

Condor changes for GRATIA

- GRATIA gets information from history logs
  - Requires one file per terminated job for efficiency
- GRATIA needs to know where the job ran
  - Additional attribute added to the job ClassAd
    (more general details on this tomorrow)

```
# condor_config.local
PER_JOB_HISTORY_DIR = /var/lib/gratia/data

JOBGLIDEIN_ResourceName=\
 "$$([IfThenElse(IsUndefined(TARGET.GLIDEIN_ResourceName), \
                    IfThenElse(IsUndefined(TARGET.GLIDEIN_Site), \
                                FileSystemDomain, TARGET.GLIDEIN_Site), \
                    TARGET.GLIDEIN_ResourceName)])"
SUBMIT_EXPRS = $(SUBMIT_EXPRS) JOBGLIDEIN_ResourceName
```

GRATIA configuration

- Essentially just tell GRATIA what name you have registered in with GOC
  - Then enable it

```
# /etc/gratia/condor/ProbeConfig

SiteName="VOx_glidein_node1"
EnableProbe="1"

# add this line to allow user jobs
# without a proxy
MapUnknownToGroup="1"
```

- You also need to tell it where to find Condor

```
# /root/setup.sh

source /etc/profile.d/condor.sh
```

The End

Pointers

- The official glideinWMS project Web page is http://tinyurl.com/glideinWMS
- glideinWMS development team is reachable at glideinwms-support@fnal.gov
- Condor Home Page http://www.cs.wisc.edu/condor/
- Condor support condor-user@cs.wisc.edu condor-admin@cs.wisc.edu

Acknowledgments

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