GlideinWMS - How and Why do we use it in CMS

Marian Zvada - KIT

GridKa School 2011
Karlsruhe, September 06, 2011

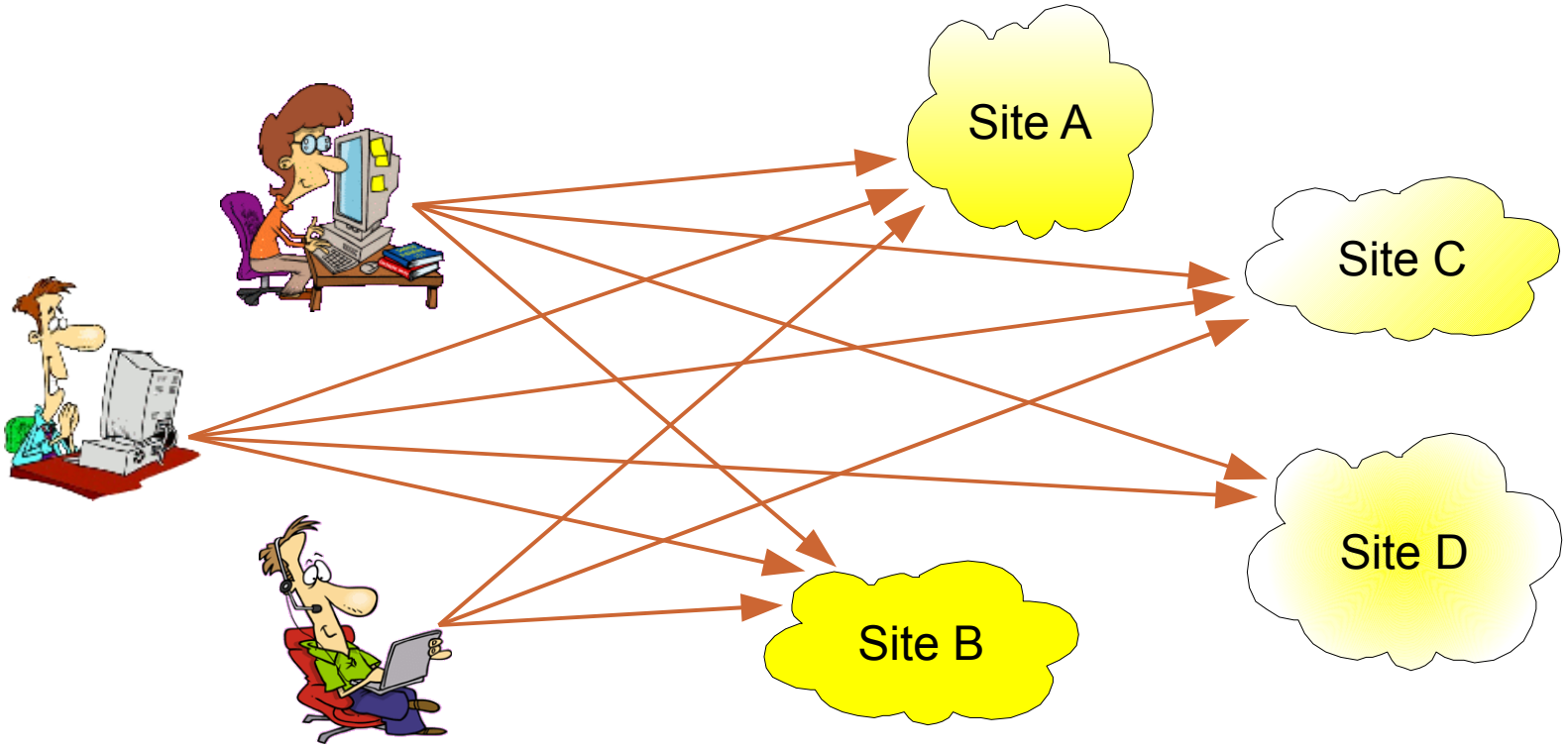
INSTITUT FÜR EXPERIMENTELLE KERNPHYSIK, KIT

What is GlideinWMS

- A WMS or Workload Management System that uses a pilot submission model to run jobs on the grid
- A pilot is a grid job that acquires a batch slot at a site, then calls home and fetches a real user job to start on it

Some background – The Grid

- Based on the principle of administrative autonomy
  - Many compute islands



- Users have to handle errors from O(N) sources

Some background - Condor

- In order to fully understand how GlideinWMS works it helps to have some basic knowledge of Condor first
- Condor is a widely used batch system
- Many sites on OSG deploy it (not only)
- GlideinWMS is heavily built on and dependent on the Condor Architecture

Some background - Condor

- In order to fully understand how GlideinWMS works it helps to have some basic knowledge of Condor first
- Condor is a widely used batch system
- Many sites on OSG deploy it (not only)
- GlideinWMS is heavily built on and dependent on the Condor Architecture

Some condor definitions

- Resource – a machine that accepts user jobs (Worker Node)
- Job – a batch program submitted to run a cluster of resources
- ClassAd – a list of attributes to describe a resource or a user job
  - For resources a ClassAd might state whether or not it is available – set in machine condor config
  - For a job it is a list of required or desired attributes for the resource it should run on – set by user on job submission

Some background – Condor Daemons

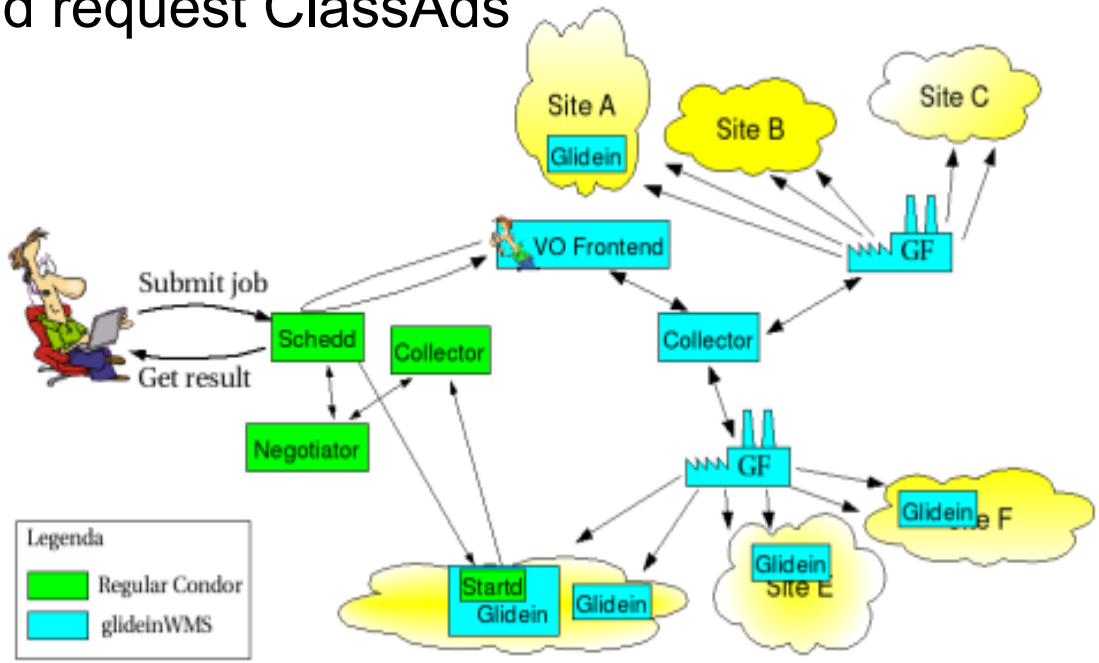
- collector – like a whiteboard that keeps track of job and resource ClassAds
- schedd – Manages user job queue, advertises job ClassAds to collector
- startd – Represents a single resource in the condor pool, advertises resource ClassAds to collector, responsible for starting user job when schedd claims it
- negotiator – Responsible for matchmaking. Traverses collector and reports a match to schedd and startd when found

Some background – glideinWMS definitions

- Glidein – a pilot job that starts a Condor startd on the grid
- Frontend – component that watches over pending user jobs and makes sure glideins are available
- Factory – component that submits glideins to the grid when it receives Frontend requests
- WMS Collector – a condor collector that keeps Factory entry ClassAds and Frontend request ClassAds

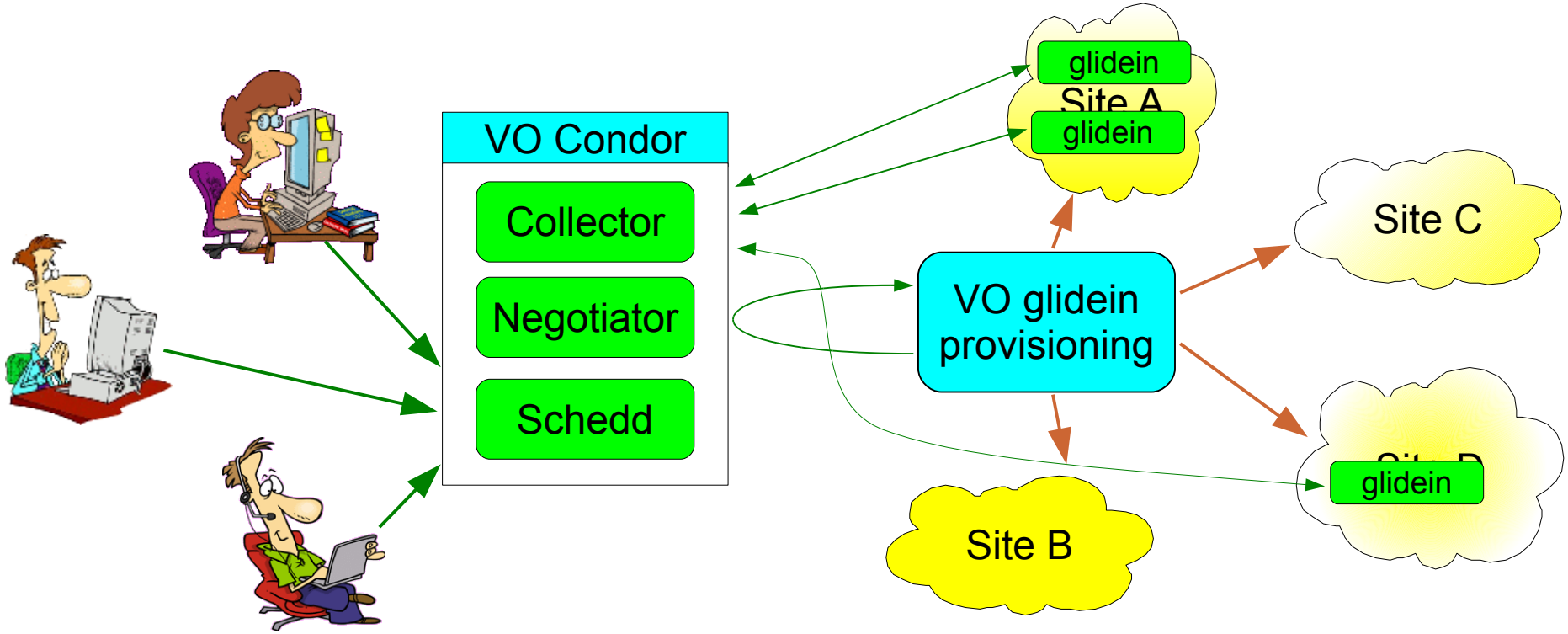
Some background – glideinWMS definitions

- Glidein – a pilot job that starts a Condor startd on the grid
- Frontend – component that watches over pending user jobs and makes sure glideins are available
- Factory – component that submits glideins to the grid when it receives Frontend requests
- WMS Collector – a condor collector that keeps Factory entry ClassAds and Frontend request ClassAds



Glideins make things better (for the users)

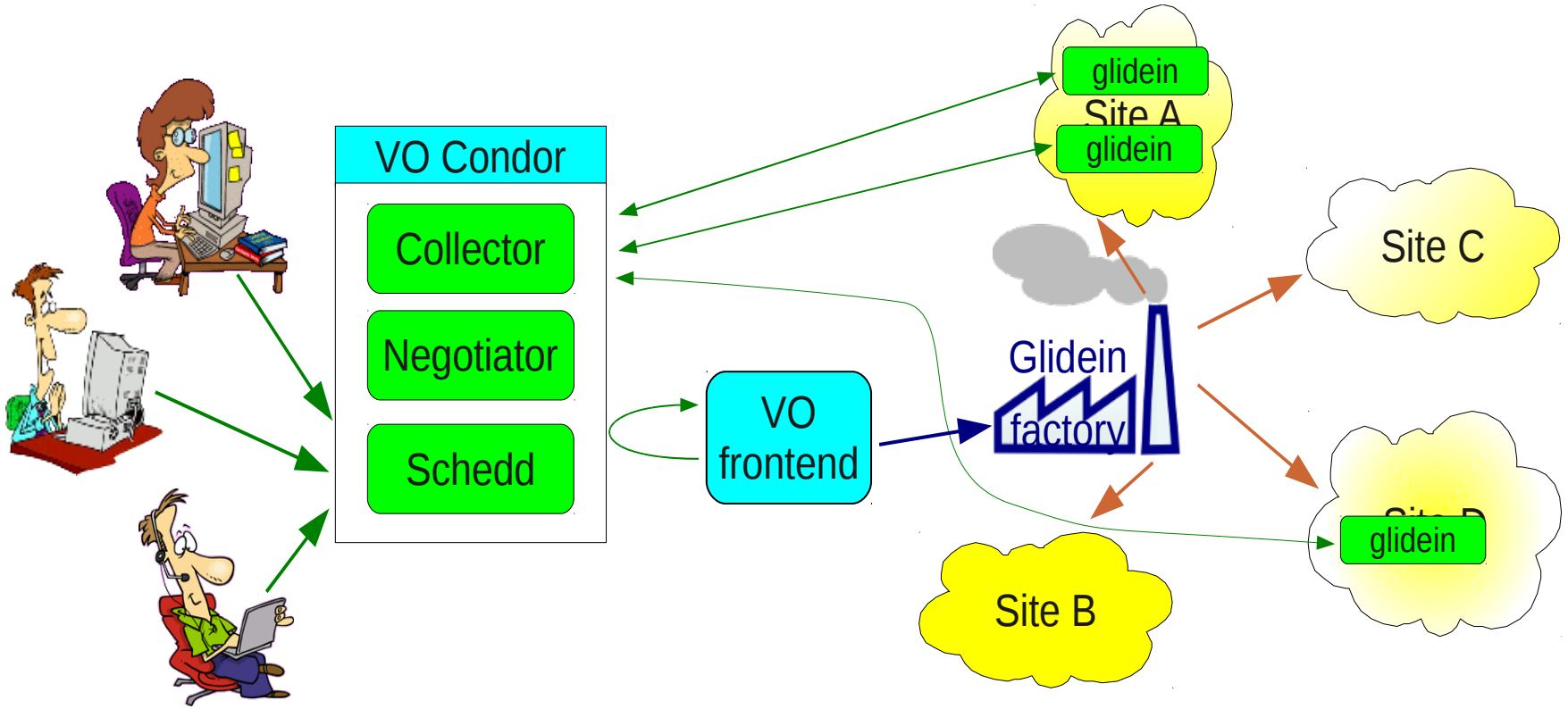
- Looks like a single Condor pool to users



- But more work for the VO admins
  - VO = Virtual Organization (e.g. group)

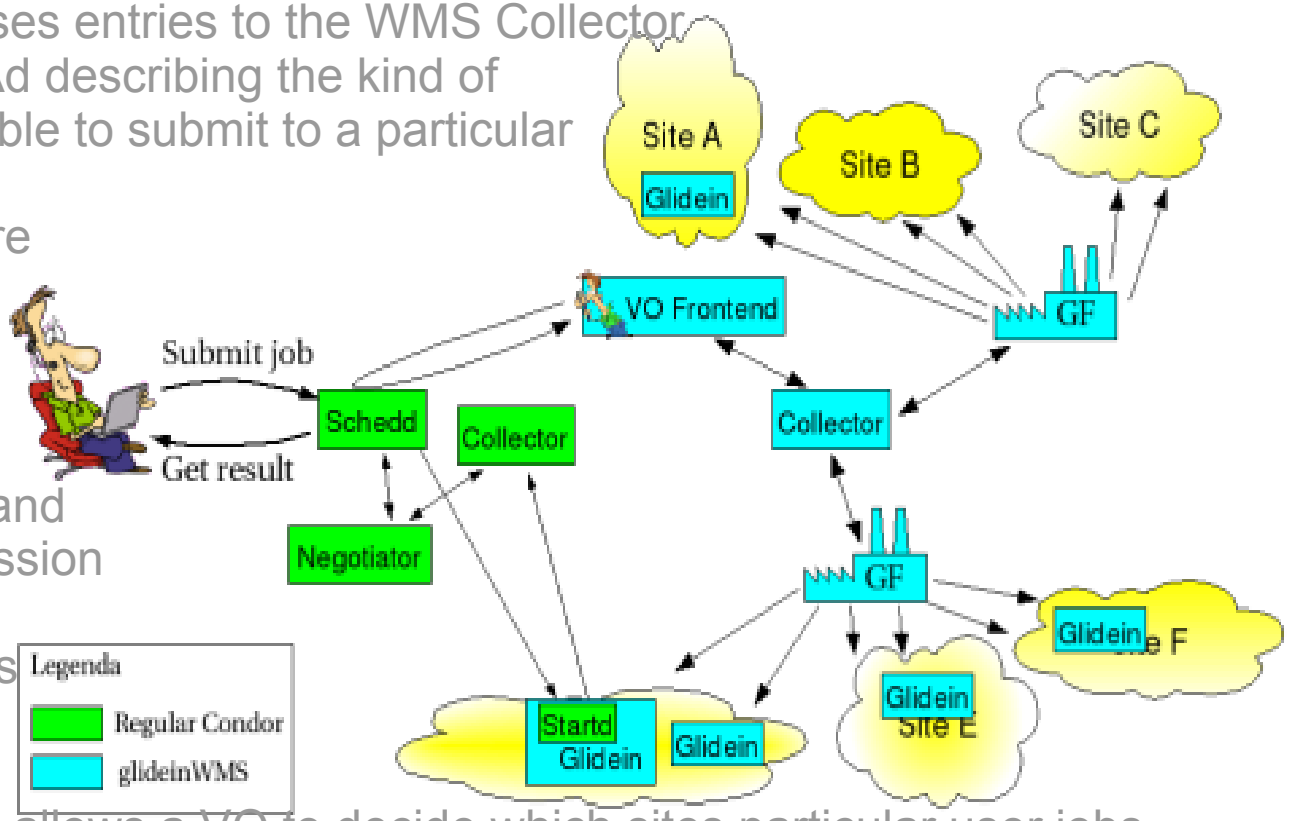
glideinWMS gets a step further

- Separates glidein submission logic from actual Grid submission of glideins
  - Only the factory sees the Grid



glideinWMS matchmaking

- The Factory advertises entries to the WMS Collector
- An entry is a ClassAd describing the kind of glidein a factory is able to submit to a particular grid site resource
- Custom attributes are defined in the entry in the factory config file (xml)
- The frontend reads the WMS Collector and uses a match expression against the factory entries and user jobs
- The match expr is defined in the Frontend config and allows a VO to decide which sites particular user jobs should run on
- If all existing glideins are already in use, the frontend posts a request to submit more glideins



CMS Frontend Match Expression

- CMS user submits jobs:
  - CRAB (AnaOps)
  - ProdAgent/WMAgent (DataOps)
    - both submits to glidein pool schedd
    - from CRAB or ProdAgent/WMAgent service point of view its no different than submitting a job to a condor cluster
- Factory defines entry attributes:
  - GLIDEIN_Site
  - GLIDEIN_Gatekeeper
  - GLIDEIN_SEs
- CRAB/ProdAgent/WMAgent submission accepts lists:
  - DESIRED_Sites
  - DESIRED_Gatekeepers
  - DESIRED_SEs
- If any of the entry attributes are in any of the submission lists it is a match

What does the glidein do when it starts?

- First a glidein runs validation scripts to ensure it can run on the Worker Node environment
  - User jobs only start on pilots that pass validation, if a validation script returns with a non-zero value the glidein terminates and reports validation error
  - This prevents glideins from starting user jobs if validation isn't passed first, user jobs never see a broken node
- The glidein then reserves the slot and starts fetching user jobs to run (one at a time)

What does the glidein do when it starts?

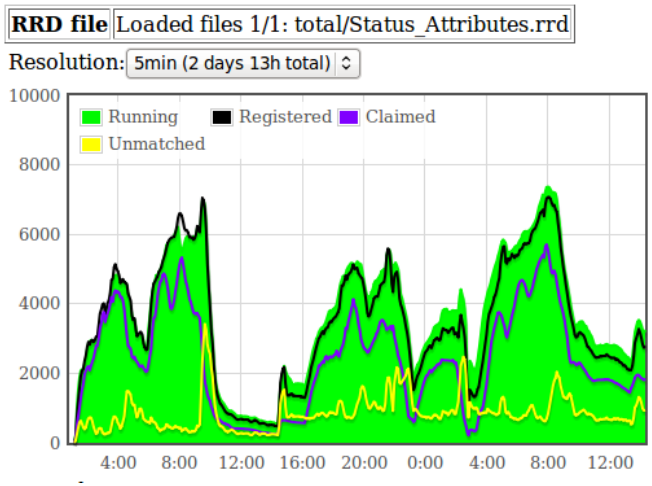
- First a glidein runs validation scripts to ensure it can run on the Worker Node environment
  - User jobs only start on pilots that pass validation, if a validation script returns with a non-zero value the glidein terminates and reports validation error
  - This prevents glideins from starting user jobs if validation isn't passed first, user jobs never see a broken node
- The glidein then reserves the slot and starts fetching user jobs to run (one at a time)
- **Key Concept:** Glideins sequentially run more than one user job and are not limited to running jobs from a single user over its lifetime

What does the glidein do when it starts?

- First a glidein runs validation scripts to ensure it can run on the Worker Node environment
  - User jobs only start on pilots that pass validation, if a validation script returns with a non-zero value the glidein terminates and reports validation error
  - This prevents glideins from starting user jobs if validation isn't passed first, user jobs never see a broken node
- The glidein then reserves the slot and starts fetching user jobs to run (one at a time)
- **Key Concept:** Glideins sequentially run more than one user job and are not limited to running jobs from a single user over its lifetime
- Frontends can include their own validation scripts to further ensure they have everything on the WN they need
- Validation errors are tracked in the monitoring making it easier to find and troubleshoot failing glideins

Site Debugging

- gWMS provides a set of monitoring tools to ensure glideins are running as expected
- If we see something is wrong we first check if it can be fixed from our end
  - Else we collect any useful debugging info from the logs/monitoring and
  - Open service tickets and work closely with the site to debug



Select elements to plot:

- ☑ Running glidein jobs
- ☐ Max requested glideins
- ☑ Glideins at Collector
- ☑ Glideins claimed by user
- ☑ Glideins not matched
- ☐ User jobs running
- ☐ User jobs idle
- ☐ Requested idle glideins
- ☐ Idle glidein jobs
- ☐ Info age

XML last update: Mon Mar 7 06:19:56 2011

		Stat			
Entry Name		**Running**	**Idle**	**Waiting**	**Pending**
CMS_T2_US_UCSD_gw2	↑	**404**	**50**	**0**	**50**
CMS_T2_US_UCSD_gw4	↑	**344**	**66**	**0**	**66**
CMS_T2_US_Nebraska_Husker	↑	**117**	**93**	**0**	**93**
HCCHTPC_T2_US_Purdue_Lepton	↑	**84**	**2**	**0**	**2**
CMS_T2_US_Nebraska_Red	↑	**91**	**73**	**0**	**73**

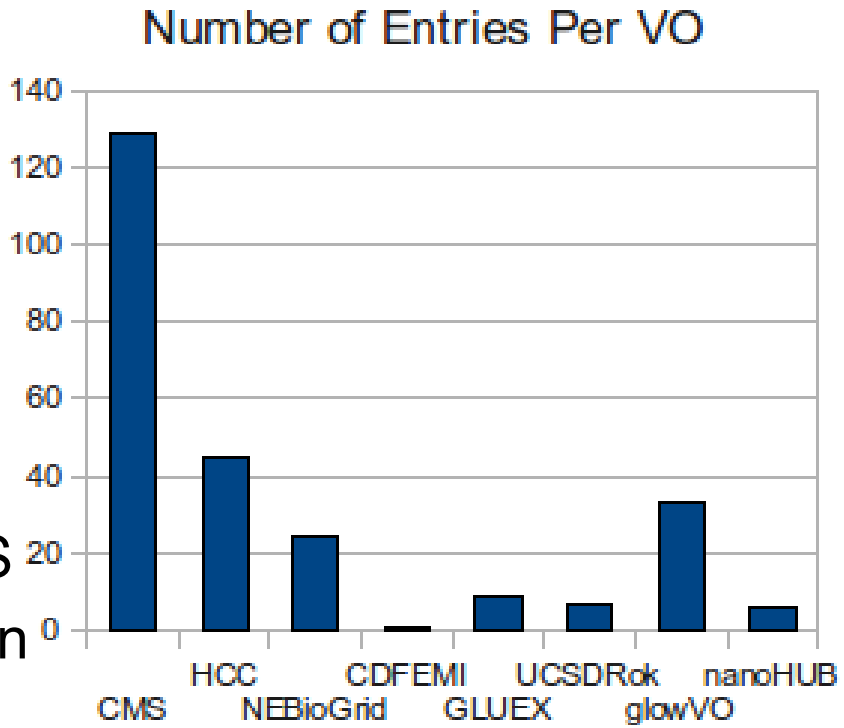
The OSG glidein Factory

- Open Science Grid is a US Grid organization
  - Co-founded by NSF and DOE
- OSG is funding a glidein factory at UCSD
  - Open to all OSG VOs using glideinWMS frontends
  - Submitting glideins to both OSG and overseas sites

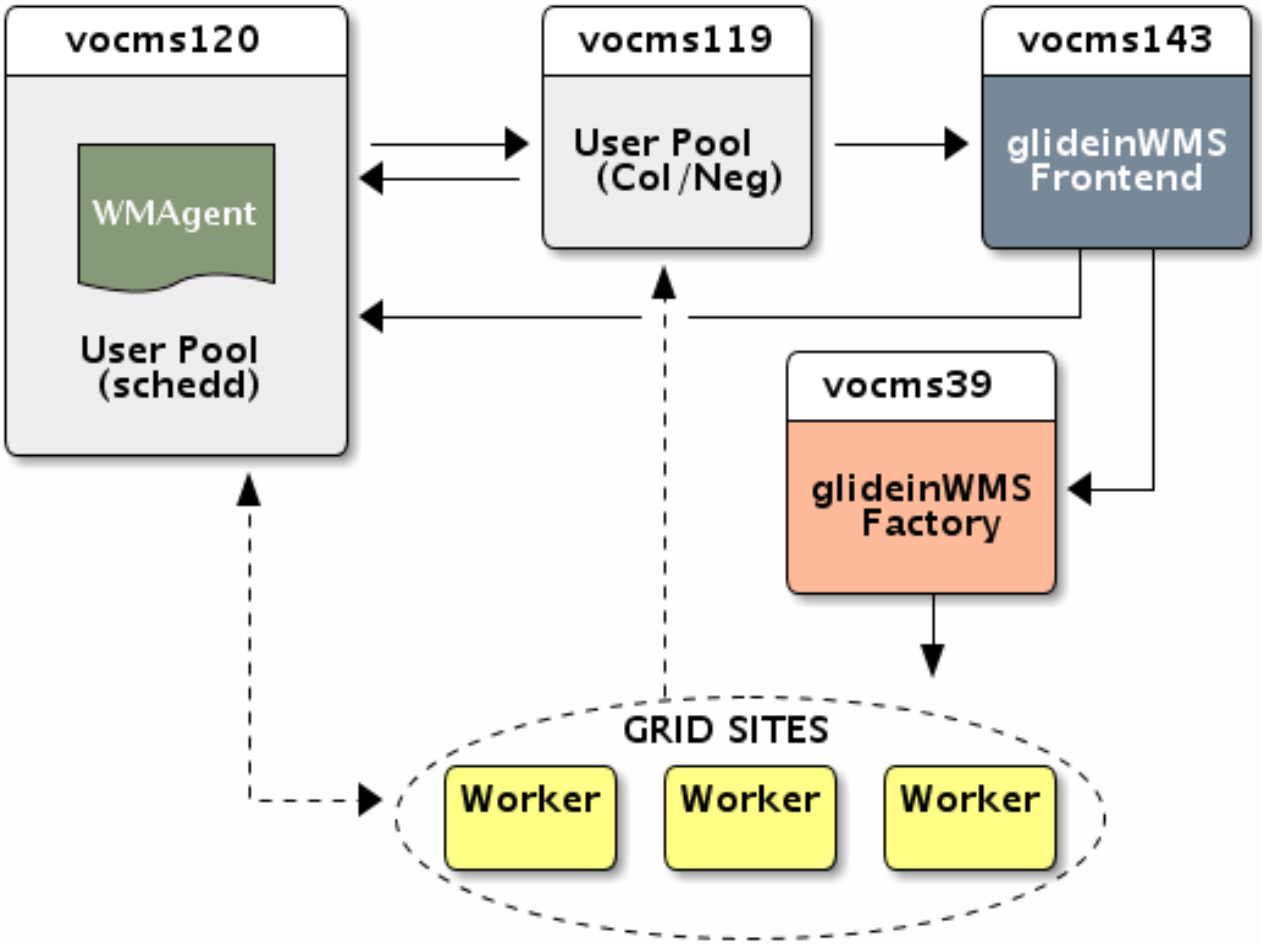
The OSG glidein Factory

- Open Science Grid is a US Grid organization
  - Co-founded by NSF and DOE
- OSG is funding a glidein factory at UCSD
  - Open to all OSG VOs using glideinWMS frontends
  - Submitting glideins to both OSG and overseas sites

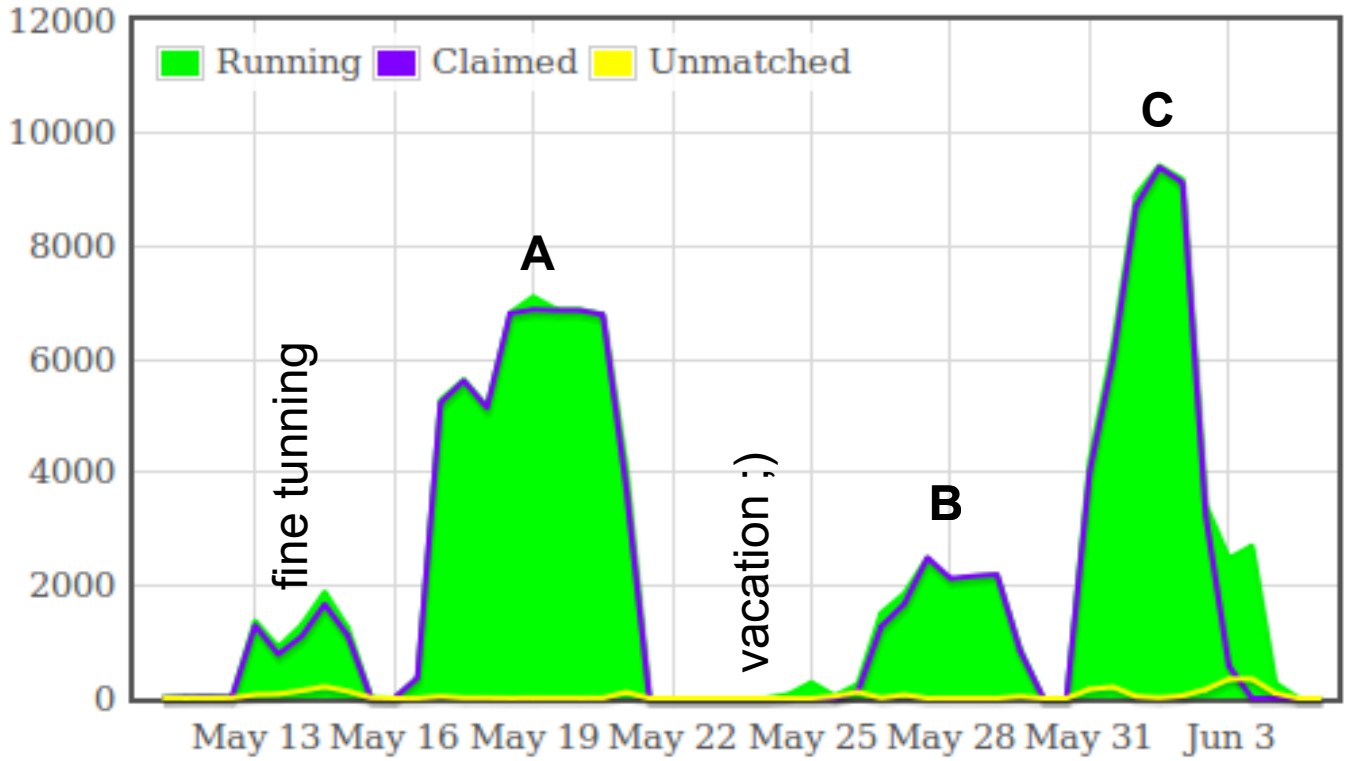
- UCSD Factory Statistics
  - ~10 active VOs served
  - 160 entries total
  - Many entries shared between VOs
  - Biggest share
    - 132 CMS Sites
  - Not just OSG site
    - 94 European CMS sites
- FNAL hosts also Factory for CMS
- CERN is commissioning one soon



The Factory setup at CERN



Performance test in CERN Factory



- **A**: FNAL sleep jobs (~6900 slots)
- **B**: UCSD sleep jobs (~2500 slots)
- **C**: FNAL + UCSD sleep jobs (~9400 slots)
- Note: attempts to use another ~2000 from UWISC, but preemption policy

Summary

- Using GlideinWMS is inherently more efficient than non-pilot grid submission techniques by reducing startup overhead
- GlideinWMS expands Condor by spreading the startds across the grid
- Submitting jobs on the grid becomes as simple as submitting to any other condor pool
- Site admins no longer have to micromanage O(1k) users. This is taken care of in the glidein negotiator by the CMS glidein admins.

Acknowledgements

- Special thanks to the glideinWMS and Condor teams
  - Igor Sfiligoi and Jeff Dost for source slides and pictures to make this presentation happen
- gWMS work is partially sponsored by
  - the US Department of Energy under Grant No. DE-FC02-06ER41436 subcontract No. 647F290 (OSG), and
  - the US National Science Foundation under Grants No. PHY-0612805 (CMS Maintenance & Operations), and OCI-0943725 (STCI).

Copyright notice

- Several images in this presentation are copyright of ToonADay.com and have been licensed by Igor Sfiligoi for use in his presentations
- Any other use strictly prohibited

glideinWMS project page

http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.prd/index.html

- Useful docs for further reference

  http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/doc.prd/documentation.html