



Rosetta

OSG Engagement VO

Resource Selection on the Grid



Engagement Mission

- Help new user communities from diverse scientific domains adapt their computational systems to leverage OSG
- Facilitate University Campus CI deployment, and interconnect it with the national organizations



User Environment

- All about usability
 - Make it easy to run simple (but real) jobs
- Engagement VO provides to its users:
 - Submit host
 - Resource selection system
 - Site verification



Why resource selection?

- Engagement users
 - Small labs/experiments
 - Grid?
 - Little or no CS/IT resources
- Example: Kuhlman lab at UNC
 - Protein folding
 - 1 job run (~ 3 days)
 - 4 weeks in wet lab, using results from run
- No knowledge / time / resources / need for a complex job handling system



Engagement VO Jobs

- Mostly very simple jobs
 - No inter job dependencies
 - Simple staging requirements
 - Inputs/outputs staged with job
- Job independence makes it easy to spread a run across many sites
- Great candidates for matchmaking



Condor

- Quick introduction to Condor
 - Representation of resources and jobs
 - Class ads
 - Match making
 - Requirements
 - Ranks



Condor ClassAds

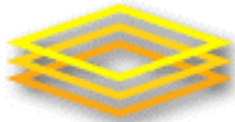
- “Condor's ClassAds are analogous to the classified advertising section of the newspaper. Sellers advertise specifics about what they have to sell, hoping to attract a buyer. Buyers may advertise specifics about what they wish to purchase. Both buyers and sellers list constraints that need to be satisfied. For instance, a buyer has a maximum spending limit, and a seller requires a minimum purchase price. Furthermore, both want to rank requests to their own advantage. Certainly a seller would rank one offer of \$50 dollars higher than a different offer of \$25. In Condor, users submitting jobs can be thought of as buyers of compute resources and machine owners are sellers.”

(source: Condor documentation)



ClassAds (cont.)

- Both resources and jobs uses the same format to describe themselves
- ASCII
- Key/value pairs
- Simplicity of the ClassAds is good news for us, because it means they are easy to create/manipulate



Resource ClassAd

```
MyType = "Machine"  
GlueSubClusterLogicalCPUs = 2  
GlueCEPolicyAssignedJobSlots = 0  
GlueCEInfoHostName = "antaeus.hpcc.ttu.edu"  
GlueHostNetworkAdapterOutboundIP = TRUE  
GlueHostArchitectureSMPSize = 2  
EngageSoftware_Rosetta_v3 = TRUE  
EngageMemPerCPU = 1010460  
GlueSubClusterWNTmpDir = "/state/partition1"  
EngageOSGAPPWriteWorkNode = TRUE  
GlueCEInfoContactString = "antaeus.hpcc.ttu.edu:2119/jobmanager-  
lsf"  
GlueHostOperatingSystemName = "CentOS"
```



Open Science Grid

Job ClassAd

FIXME



Open Science Grid

Job/Resource Requirements



Job/Resource Ranks

- How is the “best” site chosen from a set of sites matching a job’s requirements?
- FIXME
- Engagement is using a custom resource rank (explained later)



ReSS Overview

- CEMon
- Condor ClassAds
- User expresses job requirements in Condor submit file
- Condor matches jobs to resources
- Condor handles resubmits
- Use hold/release expressions to resubmit “stuck” jobs
- Post job scripts to workaround exit codes

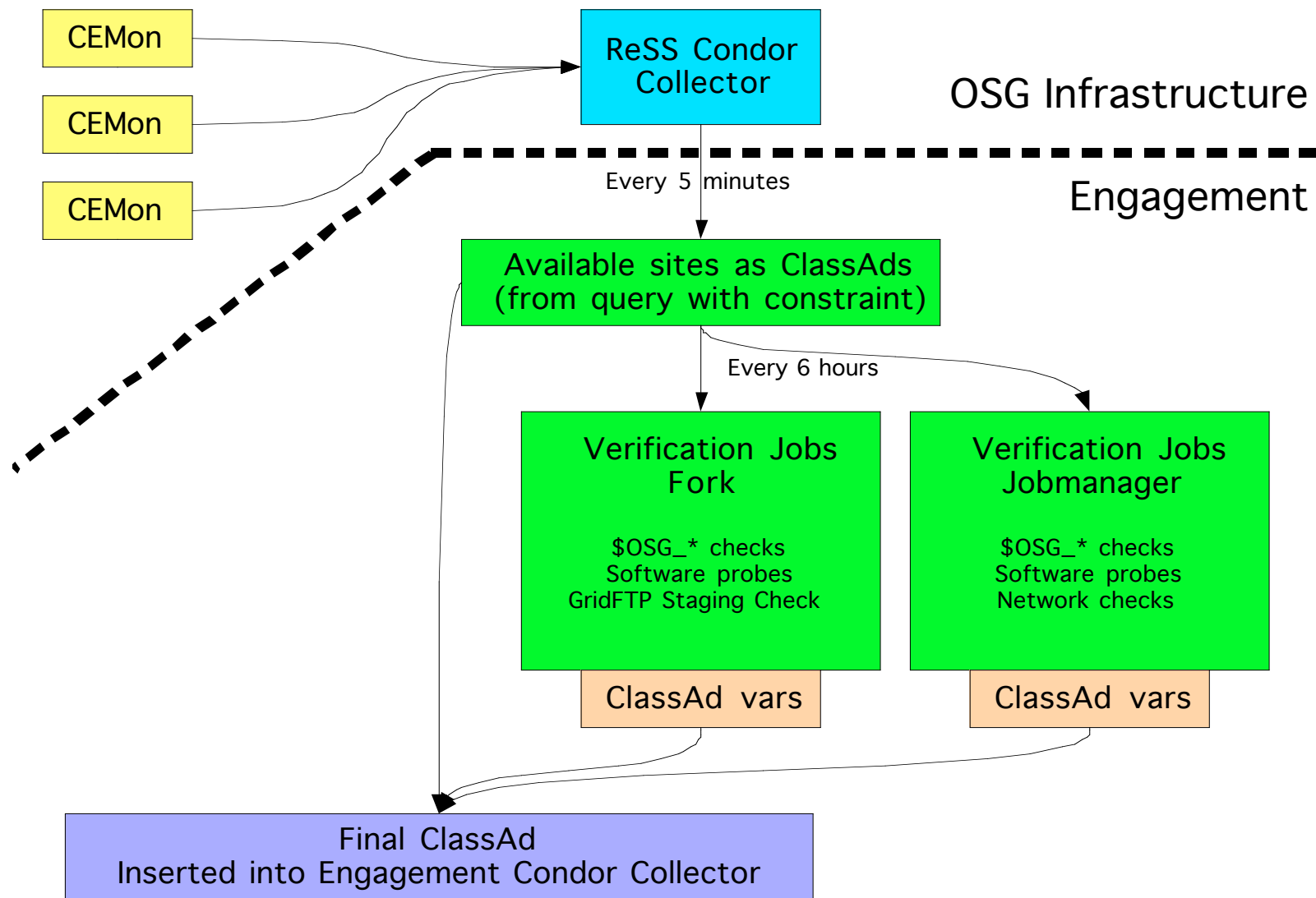


Engagement ReSS Layer

- Retrieve base ClassAds from ReSS
- Validate the sites with probe jobs
- Determine the current state of the system by looking at current job states and success rates
- Merge the information, and insert into local Condor system



Open Science Grid





ReSS + Site Verification

- ReSS information + site verification:
 - The result is a set of resource classads
 - Well formed
 - Verified information
 - Users' can trust the information
 - Increases job success rates



condor_grid_overview

| ID | Owner | Resource | Status | Time | Sta | Sub |
|-------|-------|-------------|---------|---------|-----|-----|
| 46381 | rynge | (DAGMan) | | 1:58:54 | | |
| 46382 | rynge | GLOW | Running | 1:55:43 | 1 | |
| 46384 | rynge | UWMilwaukee | Pending | 1:57:04 | 1 | |
| 46387 | rynge | Nebraska | Running | 1:00:43 | 1 | |

| Site | Jobs | Subm | Pend | Run | Stage | Fail | Rank |
|-------------|------|------|------|-----|-------|------|------|
| ASGC_OSG | 17 | 0 | 0 | 15 | 2 | 0 | 155 |
| FNAL_GPFARM | 14 | 4 | 0 | 10 | 0 | 0 | 720 |
| GLOW | 36 | 6 | 5 | 22 | 3 | 0 | 372 |
| Nebraska | 17 | 0 | 5 | 12 | 0 | 0 | 288 |
| Purdue-Lear | 15 | 4 | 0 | 10 | 1 | 0 | 372 |
| TTU-ANTAEUS | 15 | 2 | 0 | 11 | 2 | 0 | 372 |
| Vanderbilt | 45 | 4 | 4 | 37 | 0 | 0 | 350 |



Site Rank

- Simple – but works!
- Integer between 0 and 1000
- Factors:
 - Jobs submitting/staging/pending/running provides the baseline
 - Job success rate for the site over the last 6 hours
 - Ratio between matched jobs, and the max number we want on that site

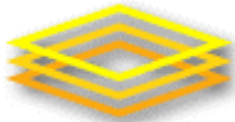


Condor Submit File

```
globusscheduler = $$ (GlueCEInfoContactString)

requirements = (
  (TARGET.GlueCEInfoContactString != UNDEFINED) &&
  (TARGET.Rank > 300) &&
  (EngageSoftwareWget == True) &&
  (TARGET.EngageCENetworkOutbound == True))
```

```
# when retrying, remember the last 4 resources tried
match_list_length = 4
Rank = (TARGET.Rank) -
  ((TARGET.Name == LastMatchName0) * 1000) -
  ((TARGET.Name == LastMatchName1) * 1000) -
  ((TARGET.Name == LastMatchName2) * 1000) -
  ((TARGET.Name == LastMatchName3) * 1000)
```



Condor Submit File (cont.)

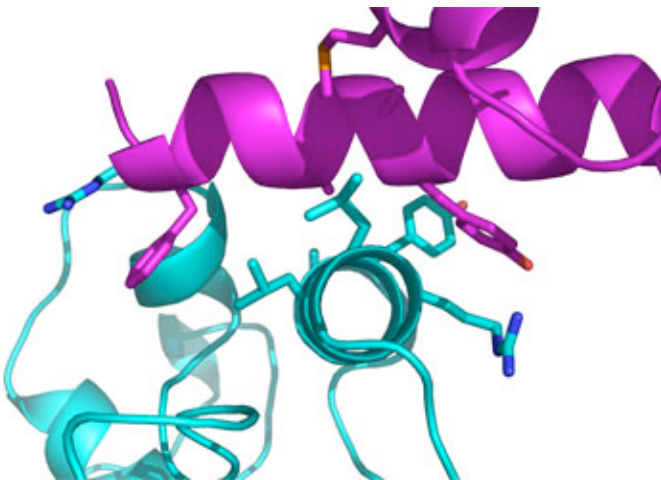
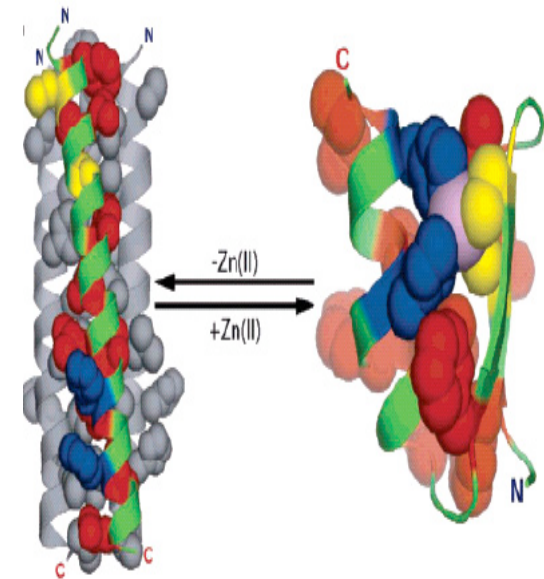
```
# make sure the job is being retried and rematched
periodic_release = (NumGlobusSubmits < 10)
globusresubmit = (NumSystemHolds >= NumJobMatches)
rematch = True
globus_rematch = True
```

```
# only allow for the job to be queued or running for a while
# then try to move it
# JobStatus==1 is pending
# JobStatus==2 is running
periodic_hold = (
  ((JobStatus==1) && ((CurrentTime - EnteredCurrentStatus) > (5*60*60))) ||
  ((JobStatus==2) && ((CurrentTime - EnteredCurrentStatus) > (24*60*60))) )
```



Sample Engagement: **Kuhlman Lab**

Using OSG to design proteins that adopt specific three dimensional structures and bind and regulate target proteins important in cell biology and pathogenesis. These designed proteins are used in experiments with living cells to detect when and where the target proteins are activated in the cells

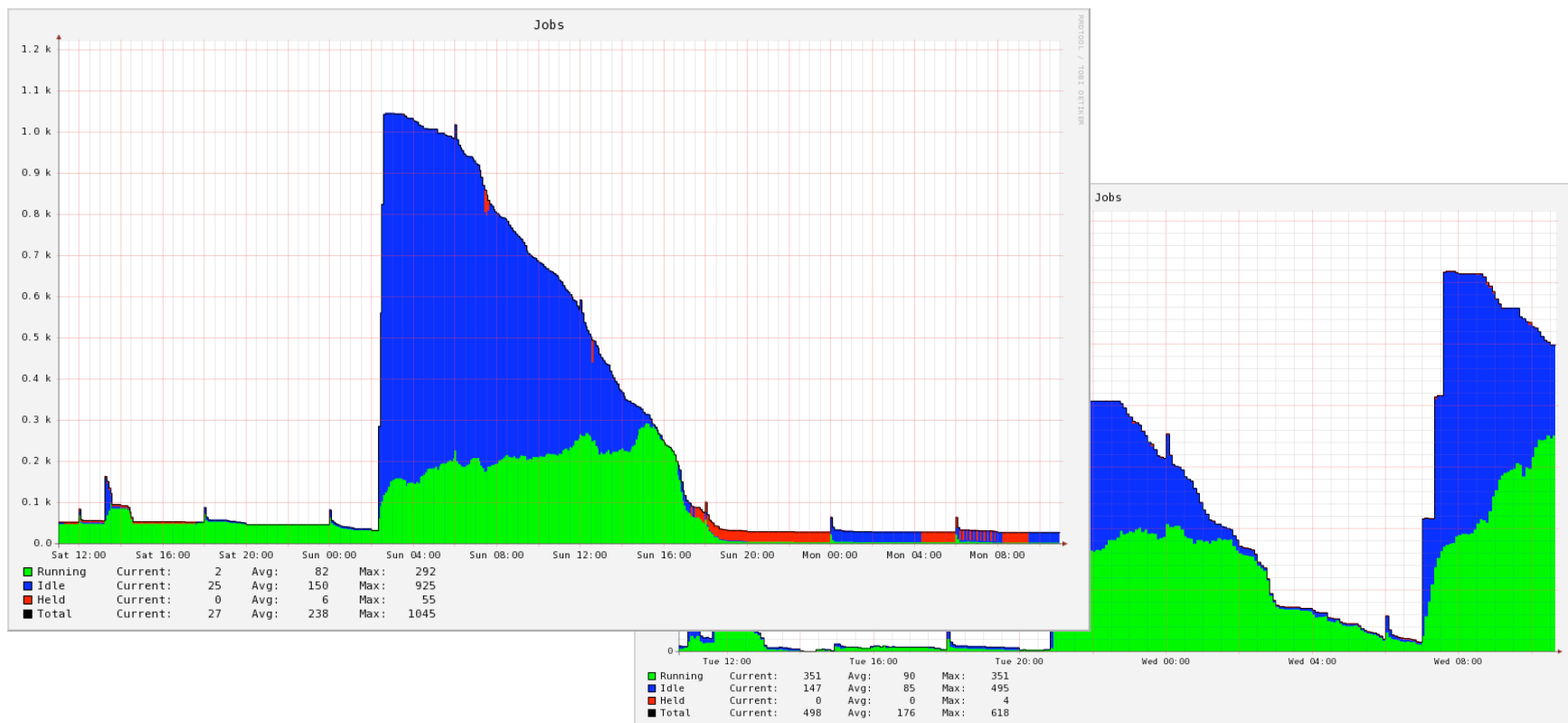


Sr. Rosetta Researcher and his team, little CS/IT expertise, no grid expertise. Quickly up and running with large scale jobs across OSG, **>250k CPU hours**



Rosetta on OSG

- Good example on why resources selection is important
 - Each protein design requires about 5,000 CPU hours, distributed across 1,000 individual compute jobs





Open Science Grid

Resource Selection

- Job submission and management for small labs/experiments
- Based on existing OSG infrastructure
 - ReSS (Resource Selection Service), Condor-G
- Component of a hosted VO infrastructure provided for users
- Easy to use meta-scheduling and fault avoidance
 - Site verification: sites are verified every 4 hours
 - Recent performance of job throughput and job success rate per site enables:
 - Job load re-balancing
 - More intelligent scheduling
 - As a site services Engage jobs well, it matches more, as it slows in servicing Engage jobs, it matches less



DEMO



More Information

- ReSS (Resource Selection Service):
<https://twiki.grid.iu.edu/twiki/bin/view/ResourceSelection/WebHome>
- OSG Engagement VO
<https://twiki.grid.iu.edu/twiki/bin/view/Engagement/WebHome>
- Questions?
 - Email: osg@renci.org