OSG Consortium Meeting

Evaluation of Workload Management Systems for OSG

by Igor Sfiligoi & Burt Holzman – Fermilab

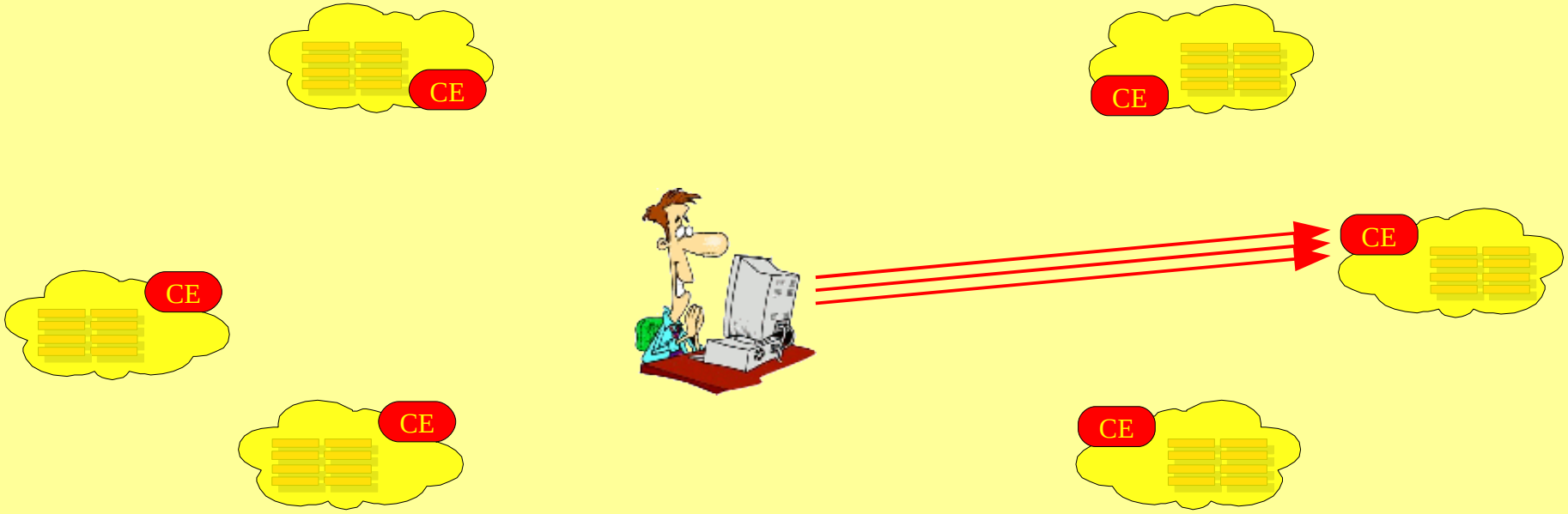
with contributions by Balamurali Ananthan

What did we test

- Scalability and reliability
  - in a single user environment
  - using several Grid sites, setup on top of production ones (Caltech, Fermilab, Madison,UCSD)
  - running simple sleep jobs, using small I/O files
- Tested WMSes
  - Plain Condor-G (http://www.cs.wisc.edu/condor/manual/v6.9/5_3Grid_Universe.html)
  - ReSS (https://twiki.grid.iu.edu/bin/view/ResourceSelection/)
  - gLite WMS (http://glite.web.cern.ch/glite/documentation/)
  - glideinWMS (http://home.fnal.gov/~sfiligoi/glideinWMS/)

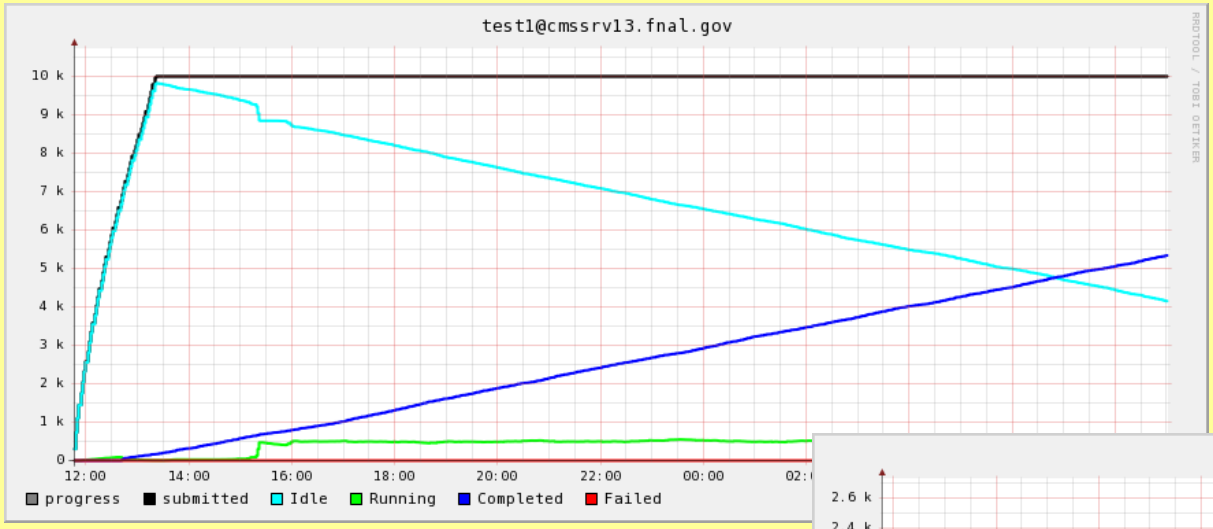
Plain Condor-G

- Manual selection of the site
  - Base test to verify CE scalability and reliability

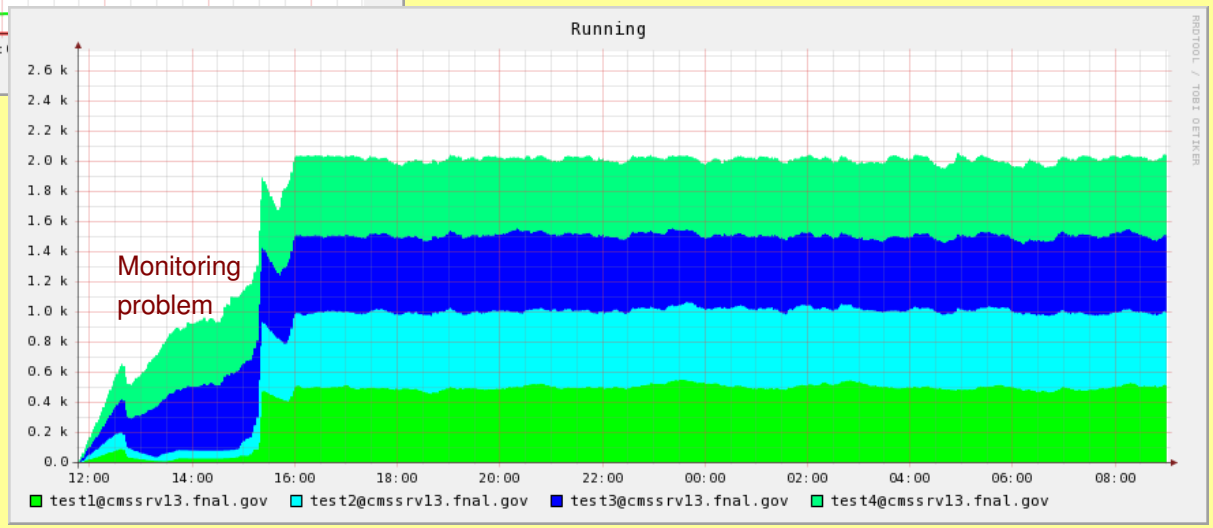


Condor-G scalability

- Scales nicely, no problems found up to 4x10k

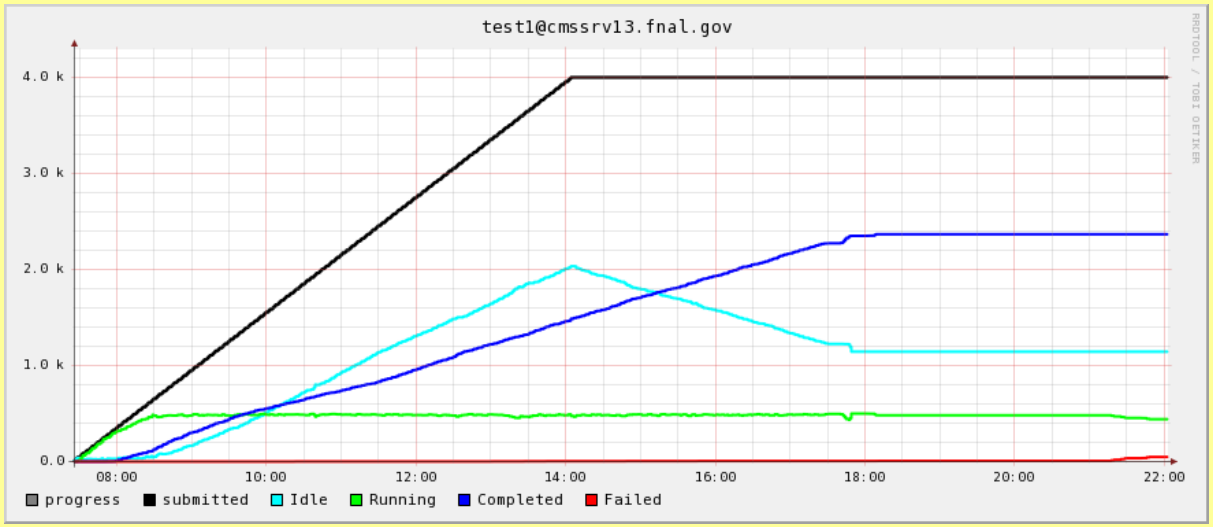


~2k batch slots at Grid site

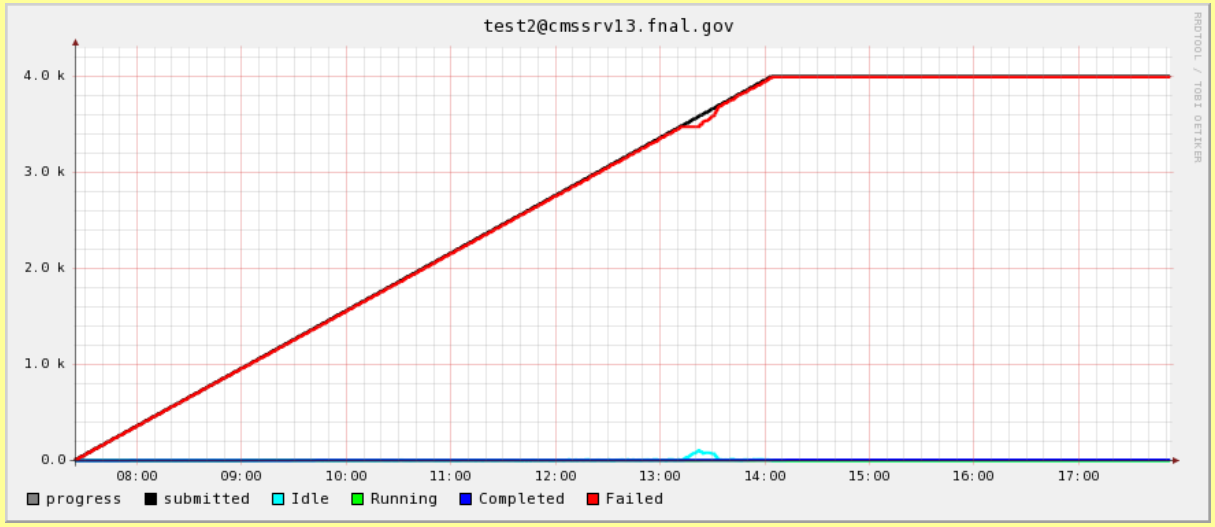


Condor-G reliability

- Works fine when Grid site stable



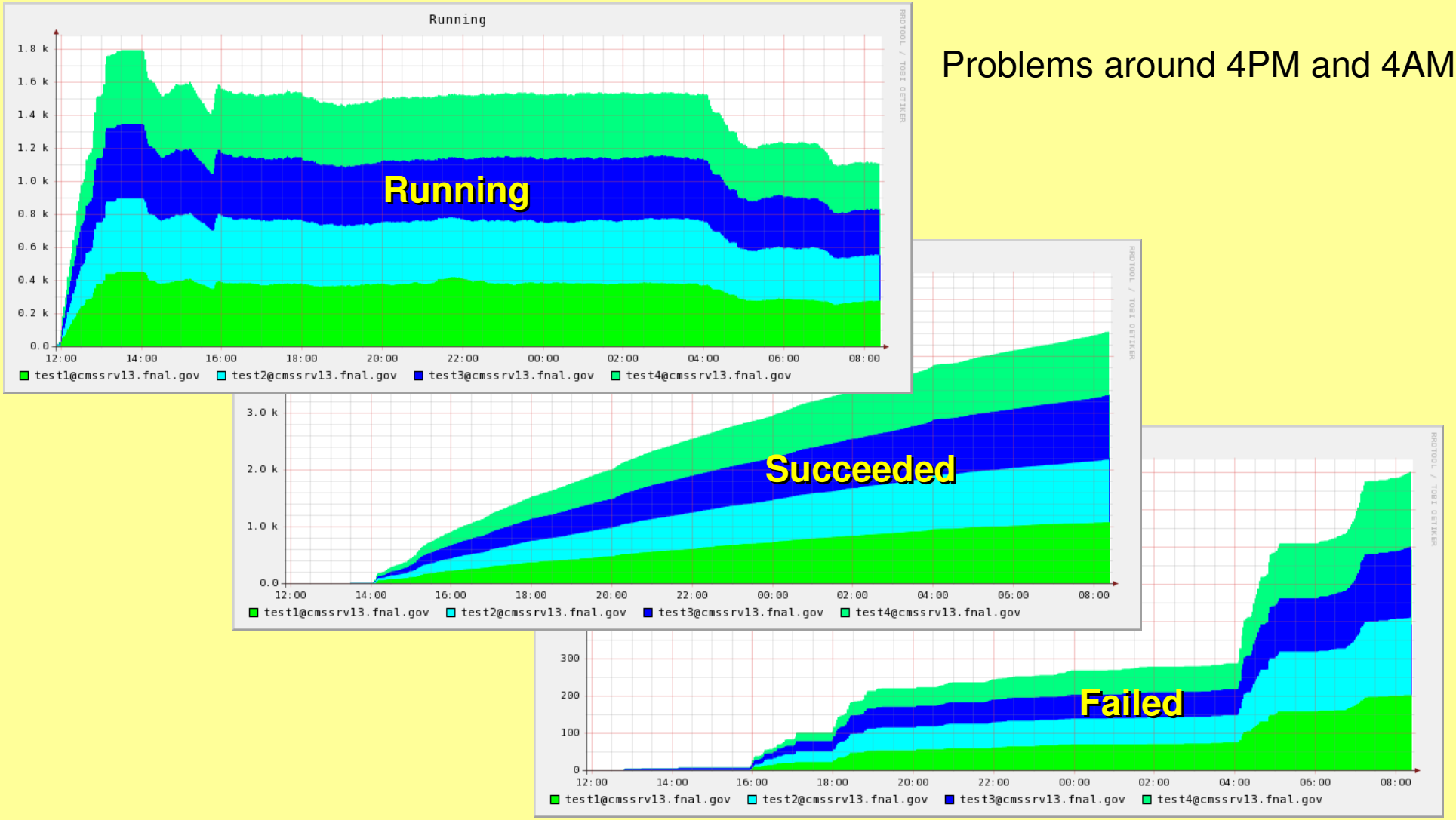
- But lots of jobs fail when Grid site misbehaves
  - Nothing that can be done on the client side



This site worked perfectly 24h ago

Condor-G reliability (2)

- Another example



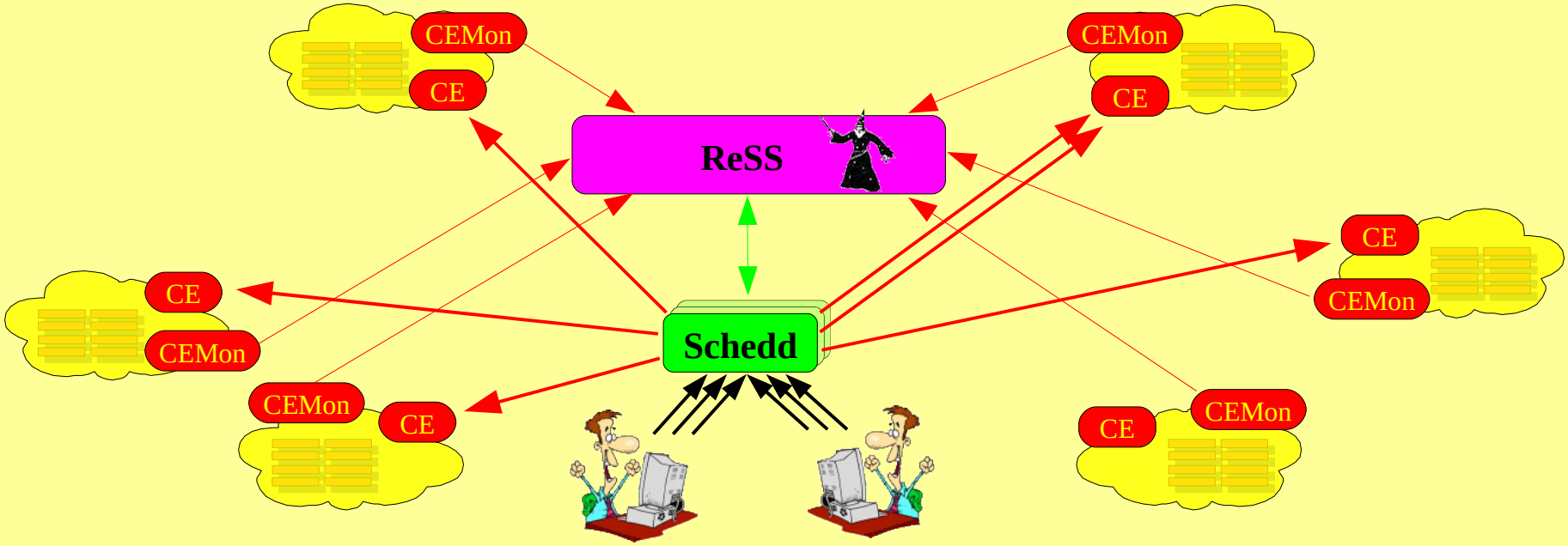
Problems around 4PM and 4AM

Condor-G reliability

- Condor-G does not handle well Grid CE crashes
  - If jobs are removed from the Grid queue before the CE comes back, Condor-G still thinks all the jobs to be there
  - If the GridMonitor process gets killed on the CE, Condor-G loses all control over the jobs that were managed by it
- I have several times observed substantial differences between what Condor-G thinks is queued and what was actually queued

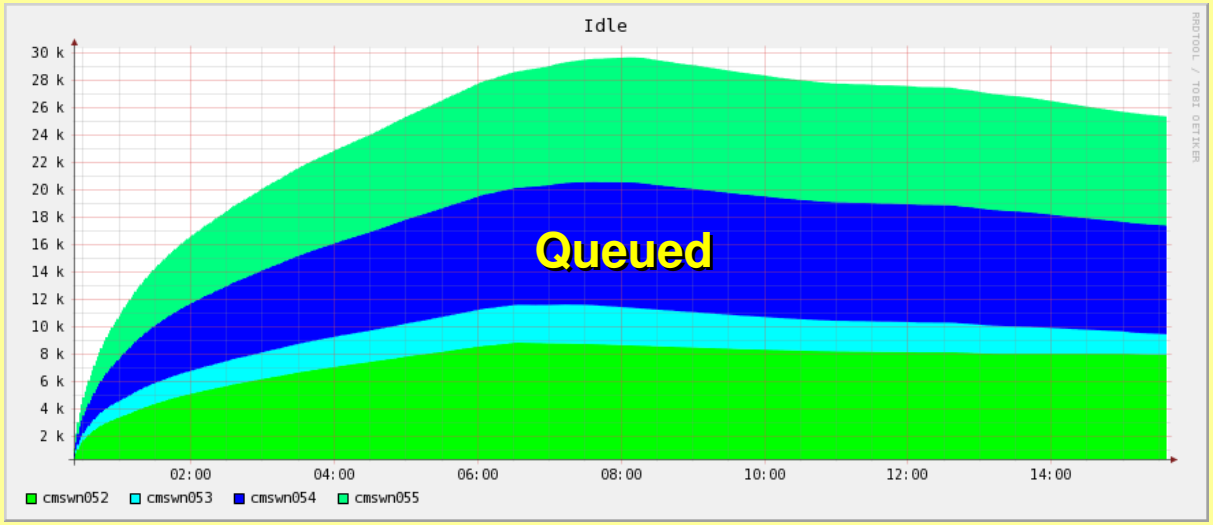
ReSS

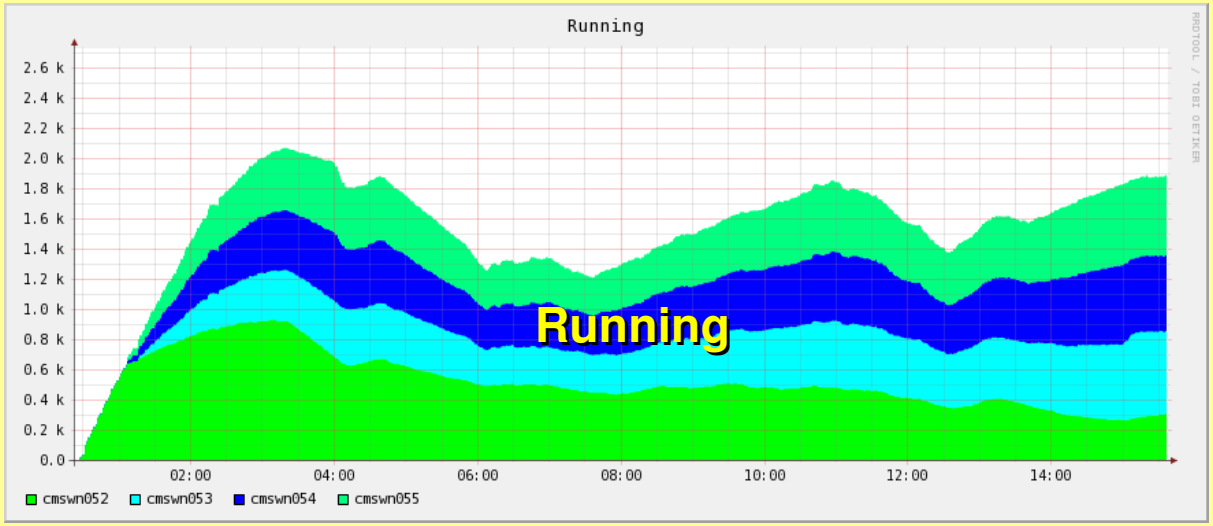
- A Condor-G based system
  - ReSS selects the Grid site for the user
  - Needs information from the Grid sites (CEMon in OSG v0.6)



ReSS scalability

- No problem up to 4x10k queued
  - Had to test on a single Grid pool (the only w/CEMon)

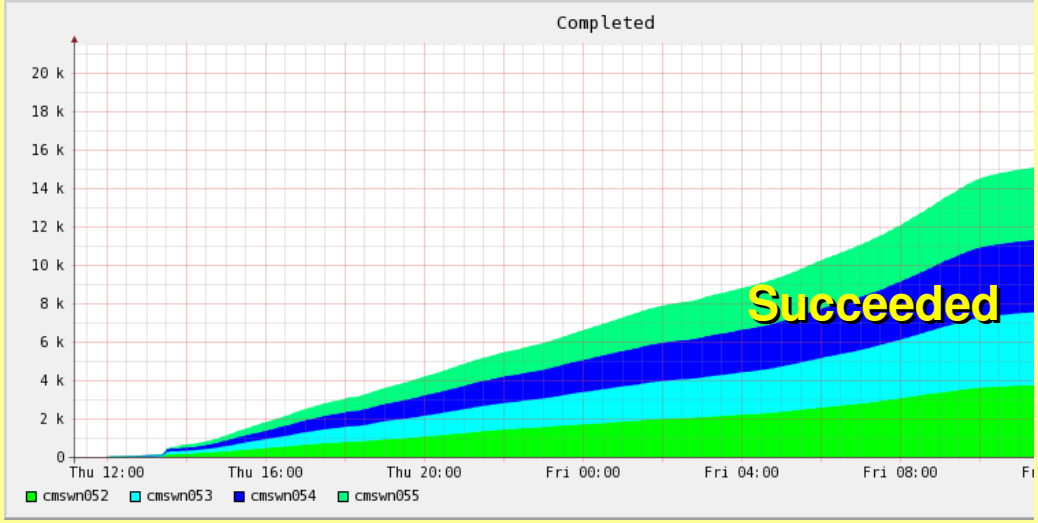


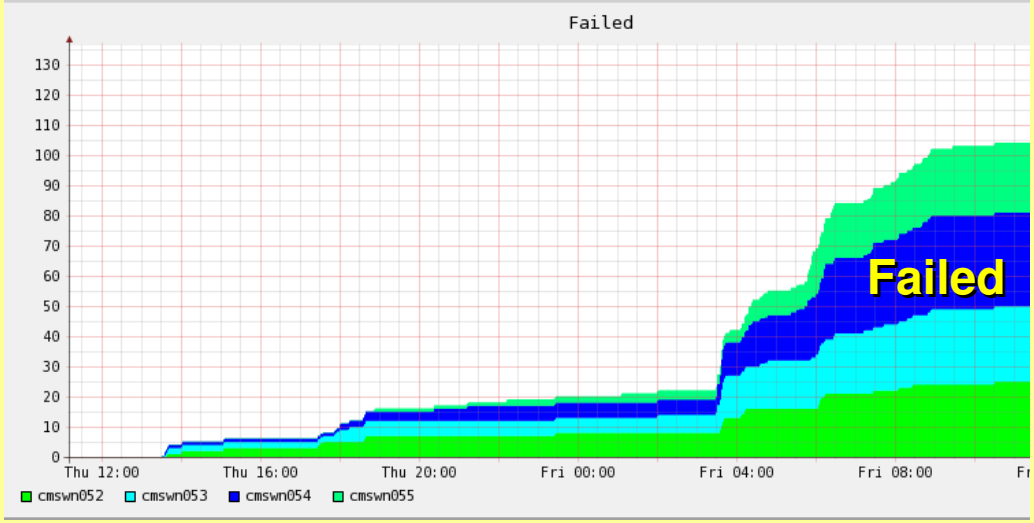


2k slots on Grid site

ReSS reliability

- Similar to Condor-G

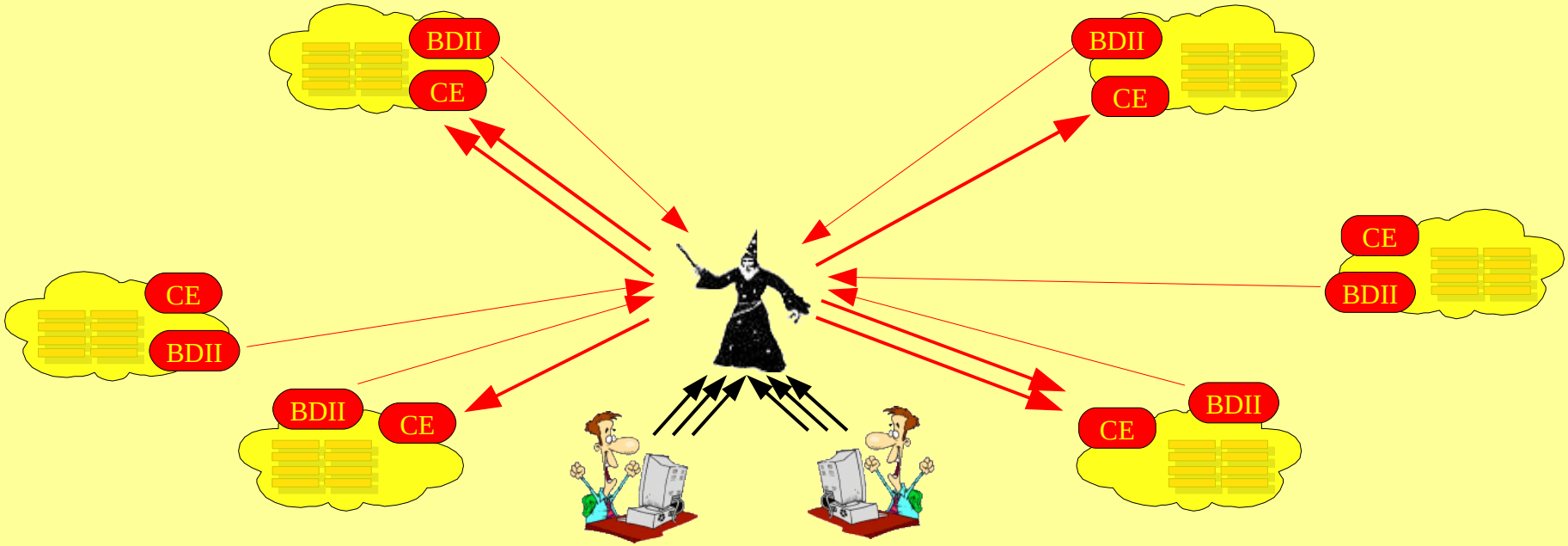




- Potentially, missconfigured CEMon can send jobs to the wrong Grid site
  - At least on paper... unfortunately, tested with just one site
- Failed jobs could potentially be automatically restarted
  - Not out the box, not tested

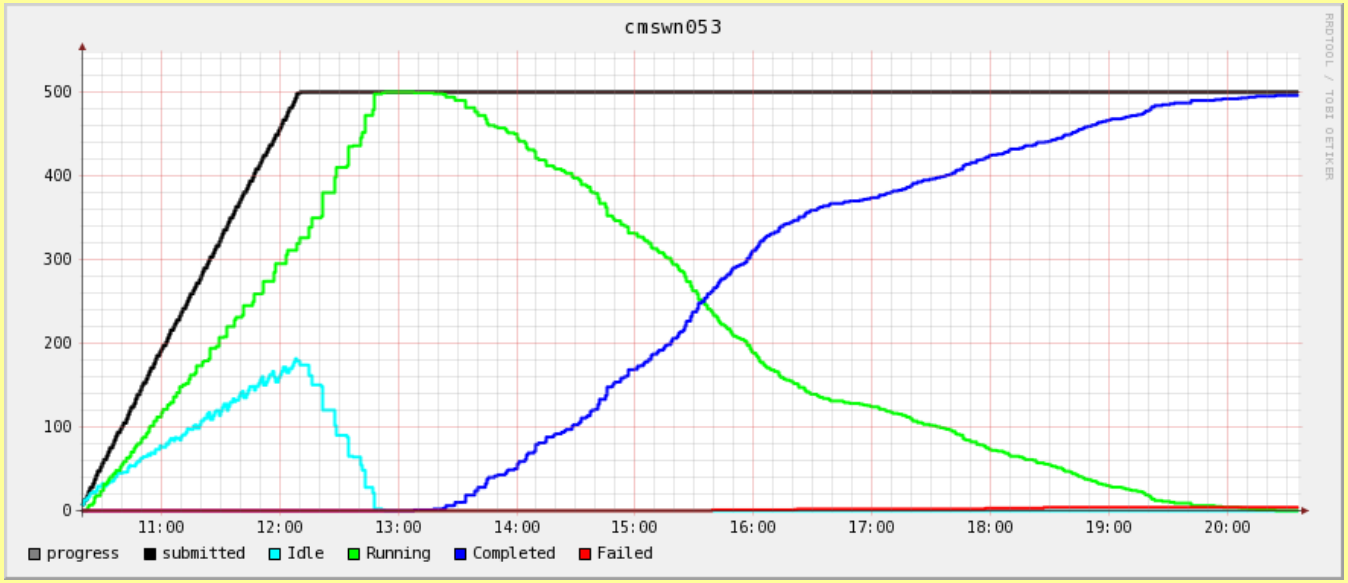
gLite WMS

- A black box solution, needs dedicated client
- Needs support from Grid sites
  - BDII for site information (available on OSG)
  - gLite tools for job execution (not available on std. OSG)



gLite WMS scalability (1)

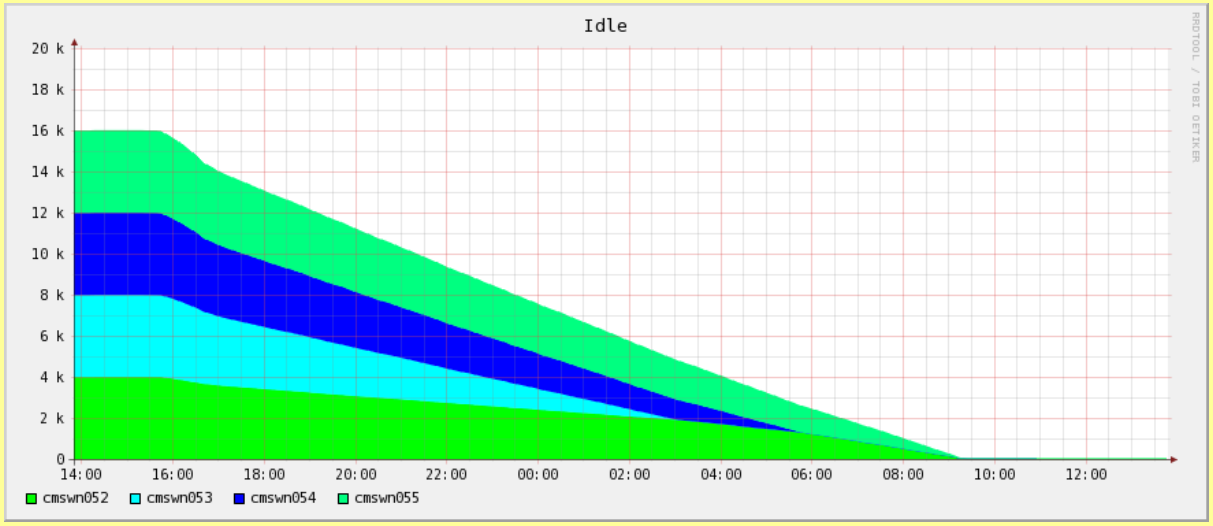
- The normal submission impractical past 4x500
  - Took 2 hours to submit (4x10k would take at least 40h!)

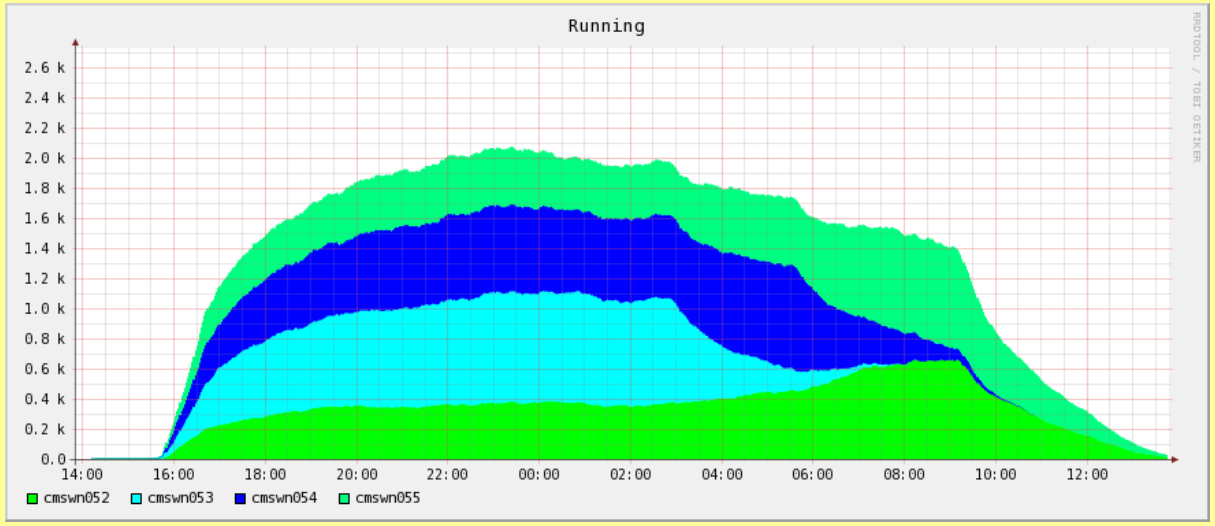


gLite WMS scalability (2)

- Bulk mode much faster: 4x4k submitted in 20mins
  - Already heavily loaded
  - **Breaks when trying 4x10k**
  - Did not test 4x5x2k or similar

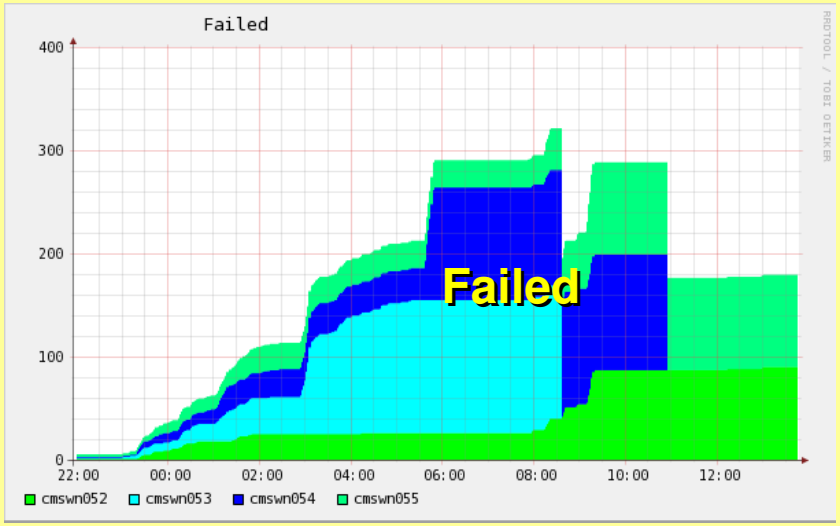
2 Grid sites
~2.5k Grid slot





gLite WMS reliability

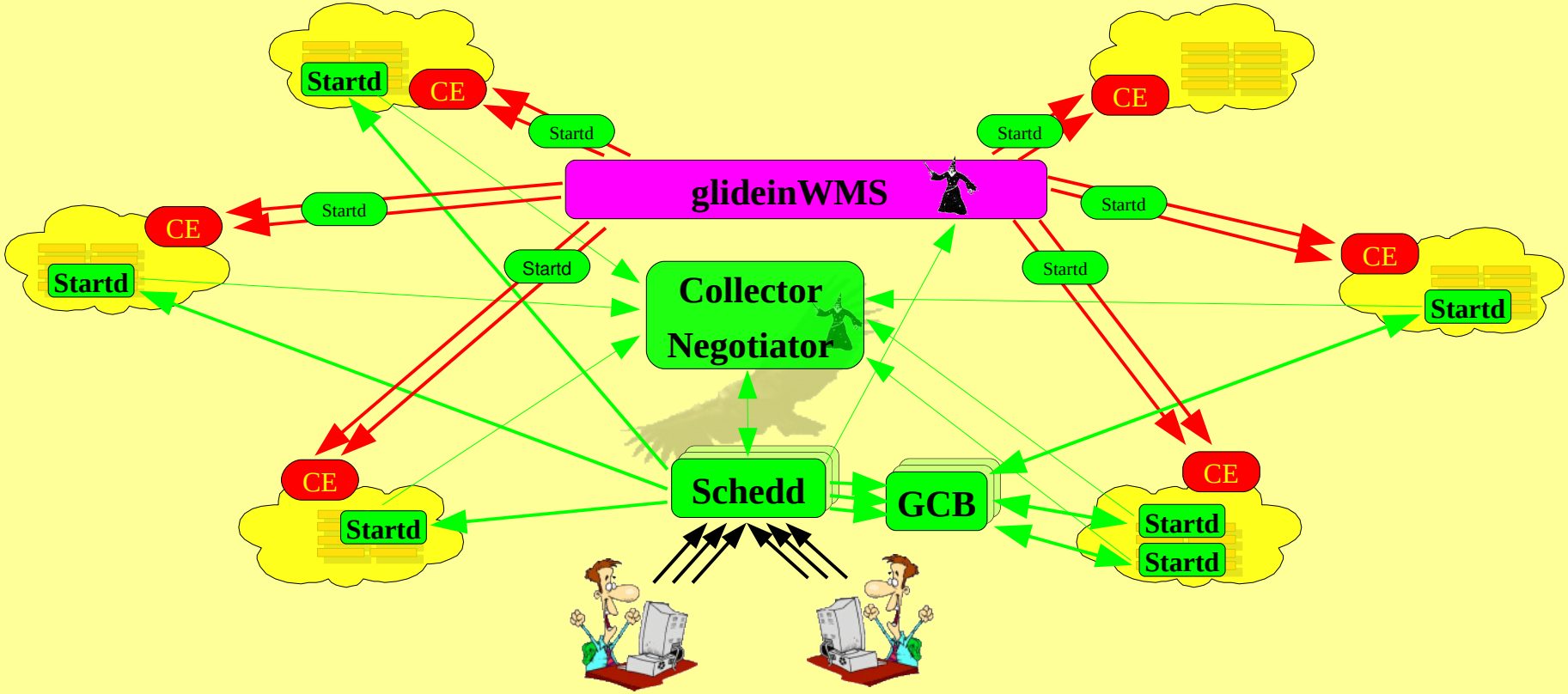
- Internally uses Condor-G, so most problems the same
  - But it does retry a job several times if CG submission fails
    - Still several jobs failed at every try



- Potentially, missconfigured BDII can send jobs to the wrong Grid site
  - At least on paper... did not happen during the test

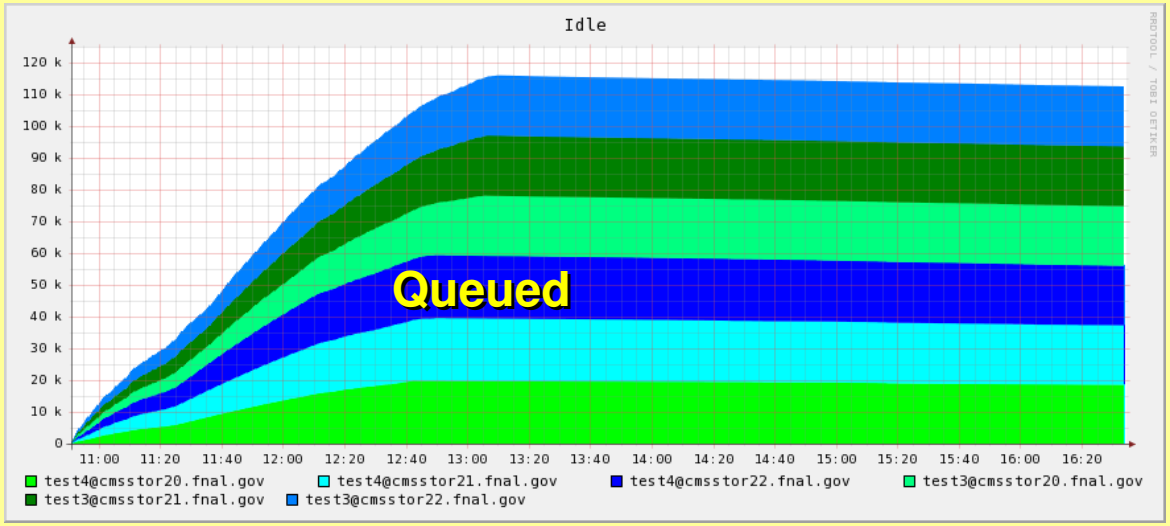
glideinWMS

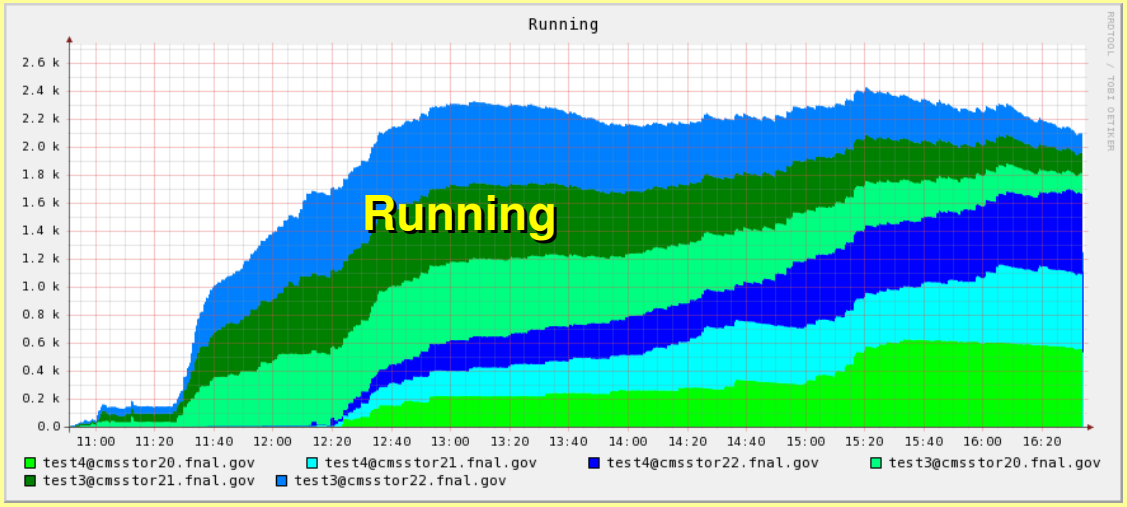
- Essentially a standard Condor pool, with startds started in a dynamic way



glideinWMS scalability

- Tested up to 6x20k jobs without finding a problem

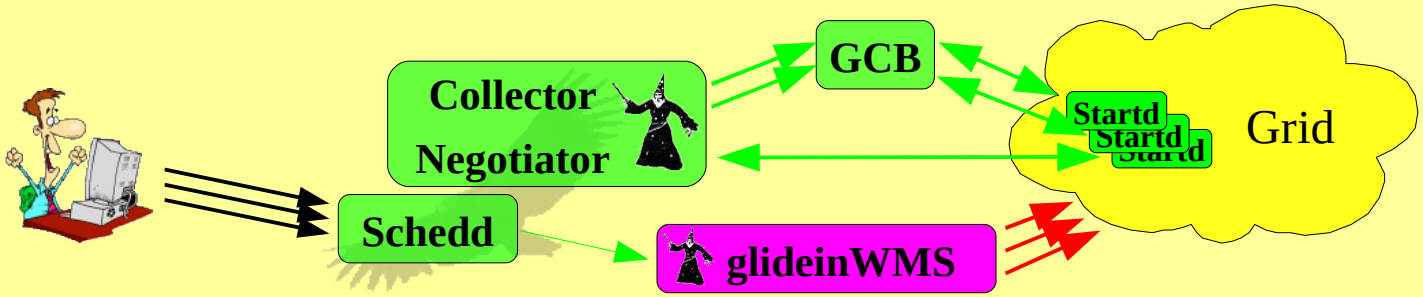




Running over 3 Grid sites

Condor scalability

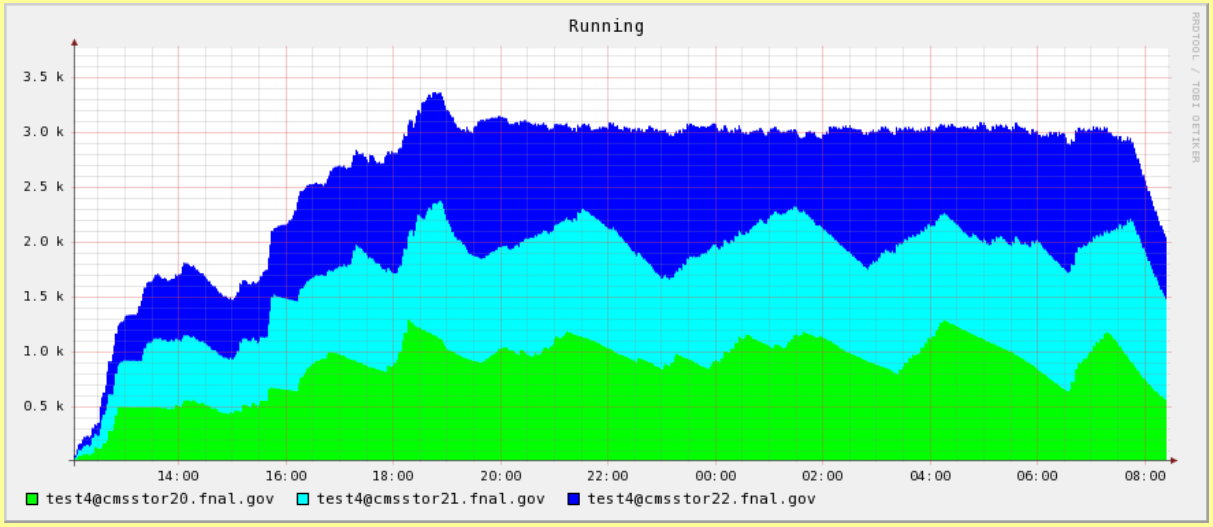
- glideinWMS just a small layer on top of Condor
  - Condor does most of the work



- Tested both Condor v6.8.x and v6.9.x branches
  - Only the latest releases of both branches scale reasonably well in the WAN environment
  - Most tests done with pre-releases, after Condor team fixed (most) observed bugs

Condor Collector scalability

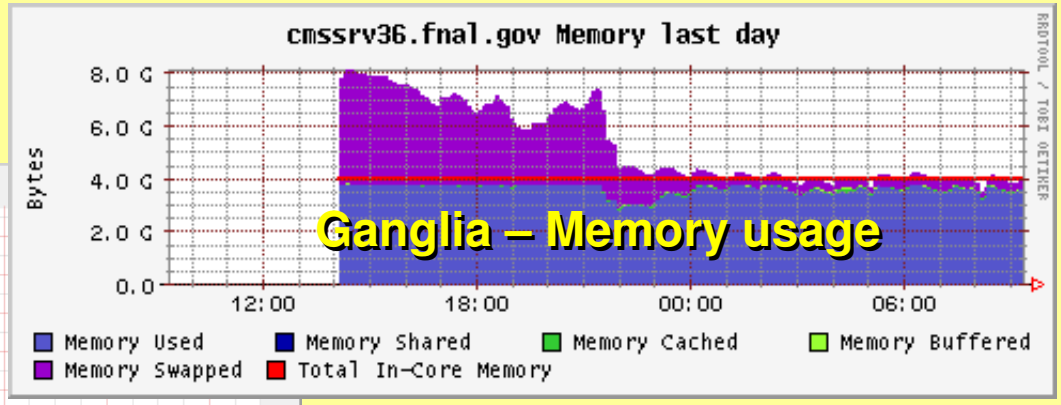
- Collector found scalable to at least 6k VMs
  - Collector was quite loaded, but jobs ran fine
  - Did not test higher, for lack of enough Grid cycles

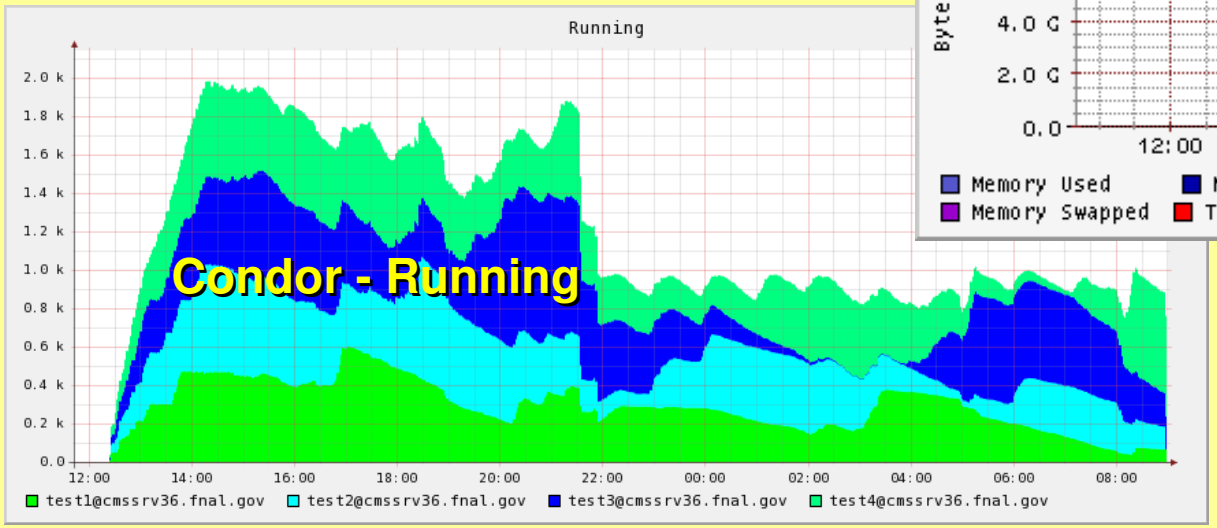


Only half VMs used by jobs in this setup

Condor Schedd scalability

- The main scalability issue found was memory consumption
  - 4M x running job!
  - Need to use multiple nodes





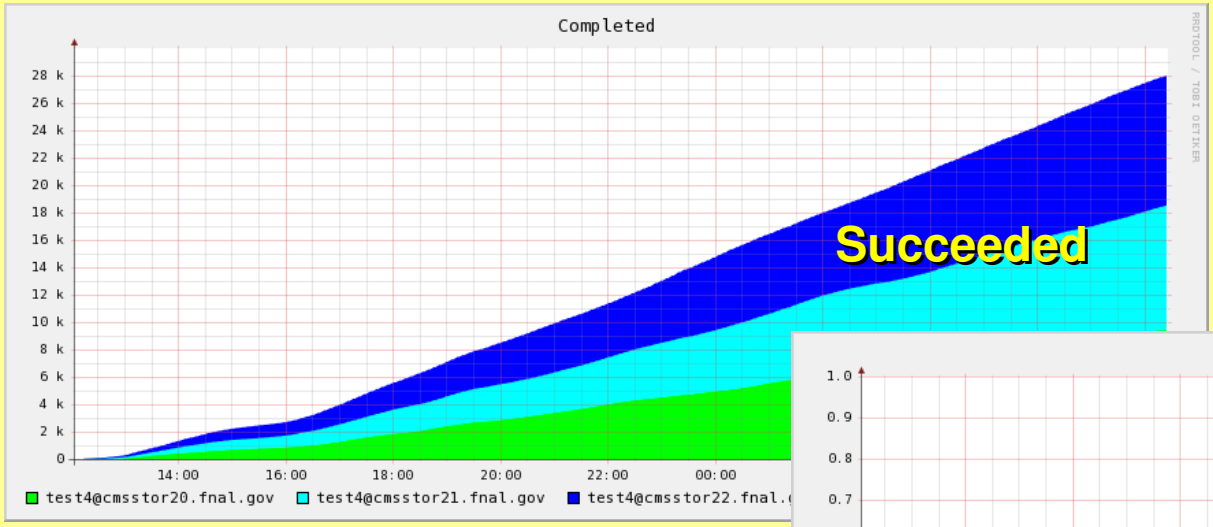
- May be a configuration issue (using strong authentication)
  - Regular Condor pools in OSG use less than 1M x running job

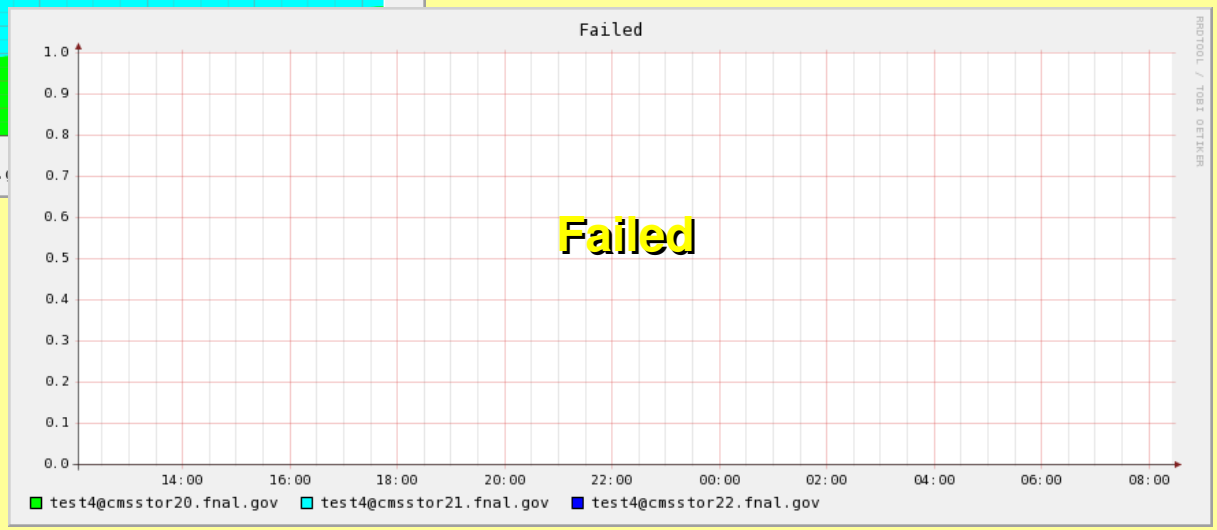
Condor GCB scalability

- Tested up to ~1500 glideins (3k VMs) per GCB
  - up to ~3k glideins with 2 GCBs
- GCB seems to scale reasonably well
  - Test jobs were running fine (with latest version)
  - However, lots of error messages seen in GCB condor logs
    - One critical problem fixed, other still under investigation
- GCB libraries sensitive to malformed packets
  - FNAL security scans occasionally crash some daemons
  - Condor team working on fixes, some in v6.9.2

glideinWMS reliability (1)

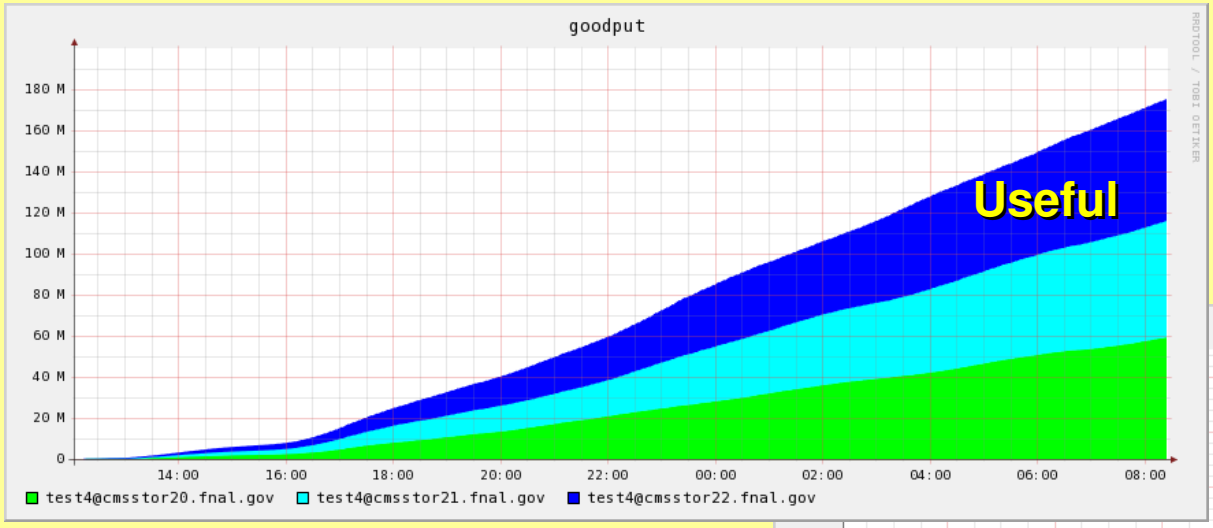
- User jobs almost never fail
  - Problematic Grid sites/nodes kill glideins not user job

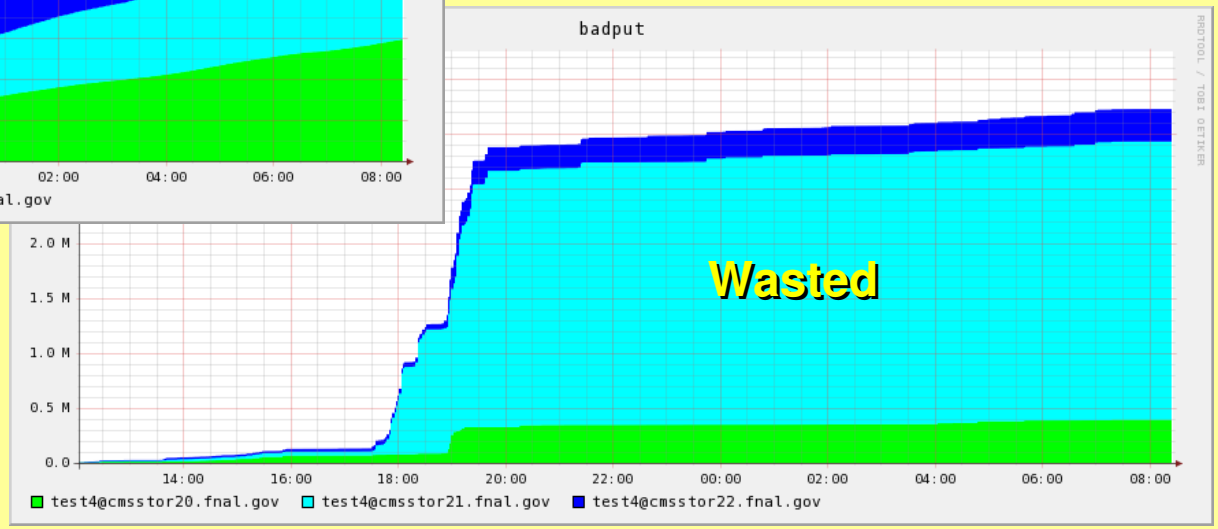




glideinWMS reliability (2)

- If glidein dies after job started, Condor will restart the user job in another glidein
  - Just wasted CPU (Checkpointing could eliminate it)





Conclusions(1)

- ReSS and glideinWMS both performed very well, gLite WMS does not scale
  - ReSS is very lightweight
    - One node can serve large number of jobs and batch slots
  - However:
    - Failures not handled
    - No global fair share
  - glideinWMS the most powerful
    - Virtually no job failures
    - Global fair share across Grid sites (not tested here)
  - However
    - Heavyweight, needs approx. two nodes every 2k batch slots
    - PULL model disliked by some Grid sites
    - Needs gLExec on WN for proper security (not in OSG0.6)

Conclusions(2)

- For automated tasks involving just a few entities, ReSS may be preferable
  - Lightweight, failures can be recovered by the submittor
- For multi-user environments sporting real users, glideinWMS is definitely the way to go if you can afford the needed hardware
  - Virtually no user job failures and real global fair share a must for the average user

Official selection

- This work was sponsored by USCMS to select promising WMS candidates
- Our secondary goal was to also help OSG itself select an official WMS

Next steps

- Additional tests of ReSS and glideinWMS
  - Bigger I/O files
  - More Grid sites
  - Multiple users
- Integrate ReSS and glideinWMS into CMS MC and analysis tools