CHEP 2007

An Objective Comparison Test of Workload Management Systems

by
Igor Sfiligoi (FNAL)

in collaboration with
Burt Holzman (FNAL)

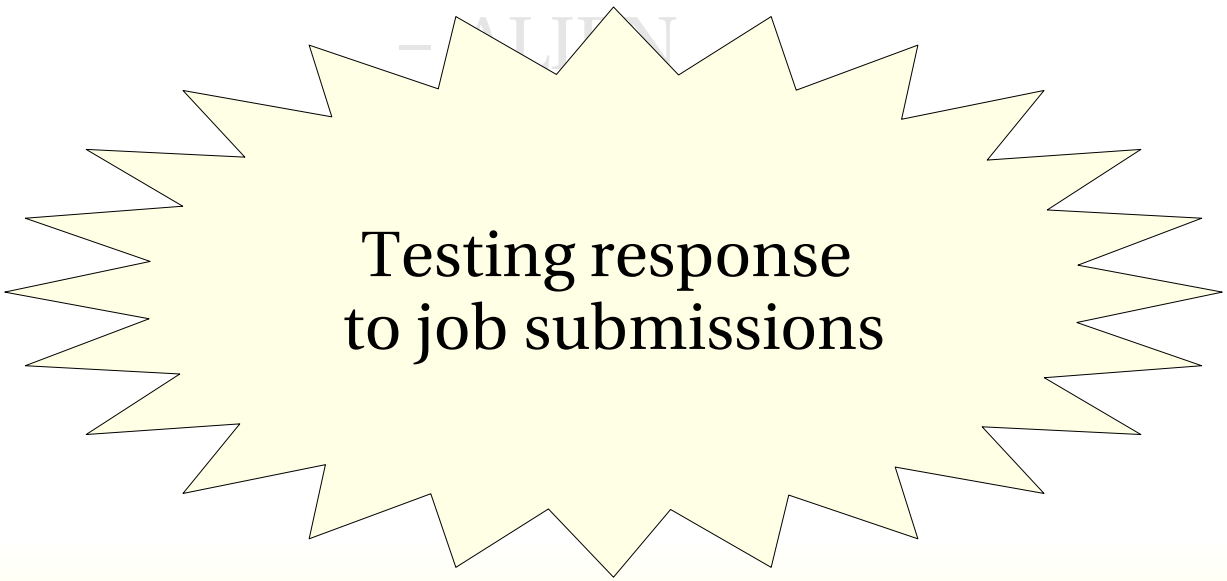
with help from
Terrence Martin(UCSD), Michael Thomas(Nebraska) and Daniele Cesini (CNAF)

Available WMSes

- General purpose
  - Condor-G
  - ReSS
  - gLite WMS
  - glideinWMS
- Experiment/group specific
  - Panda
  - CRONUS
  - ALIEN
  - DIRAC
  - GlideCAF
  - etc.

Tested WMSes

- General purpose
  - Condor-G
  - ReSS
  - gLite WMS
  - glideinWMS
- Experiment/group specific
  - Panda
  - CRONUS
  - ALIEN



Condor-G

- Part of the Condor distribution
  - Although only loosely coupled with the rest of the Condor system
- Supports multiple submission portals
  - Pre-WS and WS Globus
  - Nordugrid
  - unicore, pbs and lsf
- Most other WMSes use it as the underlying submission mechanism to the Grid

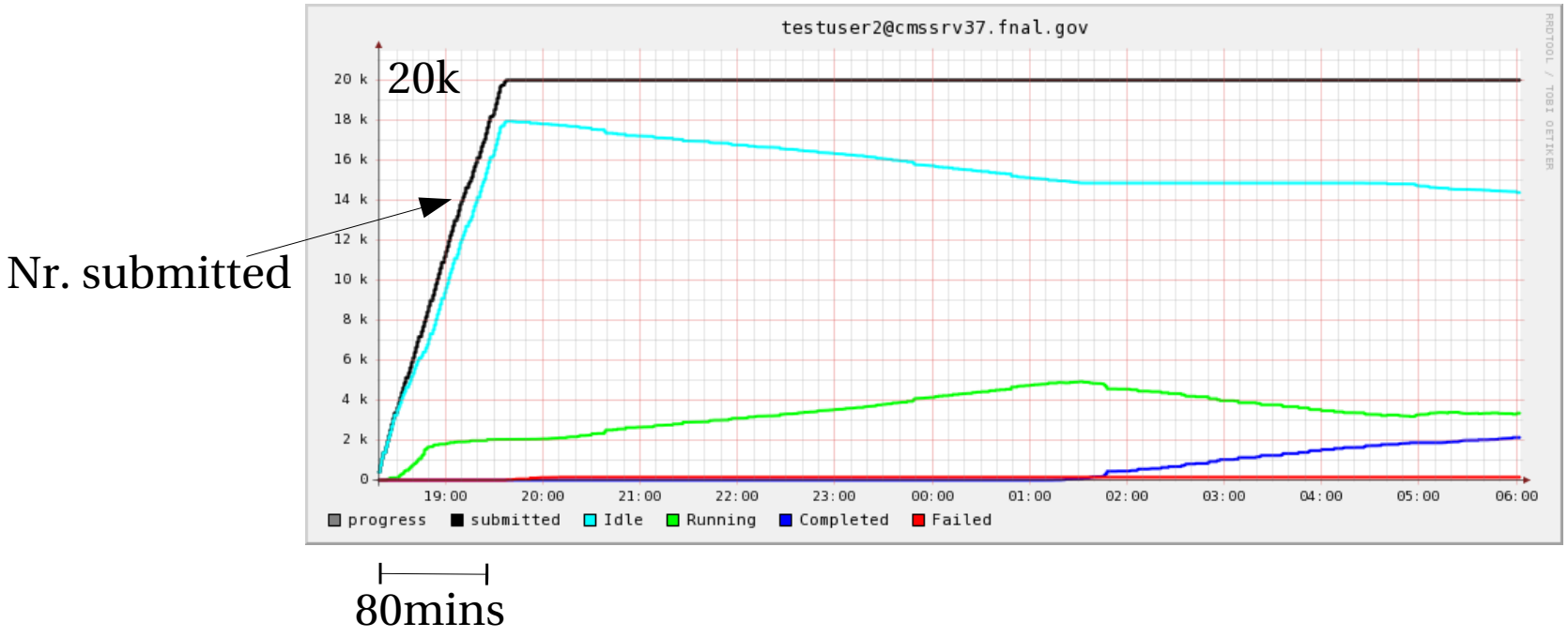
Condor-G

- Part of the Condor distribution
  - Although only loosely coupled with the rest of the Condor system
- Supports multiple submission portals
  - Pre-WS an
  - Nordu
  - unicore, p
- Most other WMSes use it as the underlying submission mechanism to the Grid

Only Pre-WS Globus (also known as GT2) submissions tests

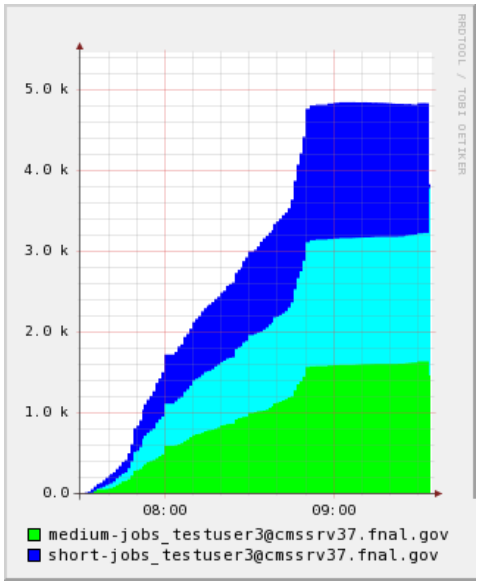
Condor-G submission speed

- At least up to 20k jobs seems to be linear
  - Approx 250 submissions per minute

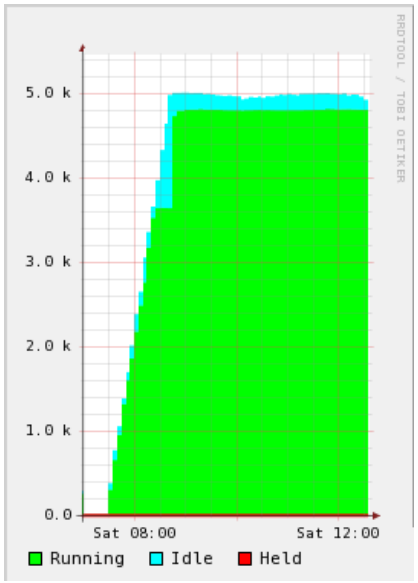


Condor-G startup speed(1)

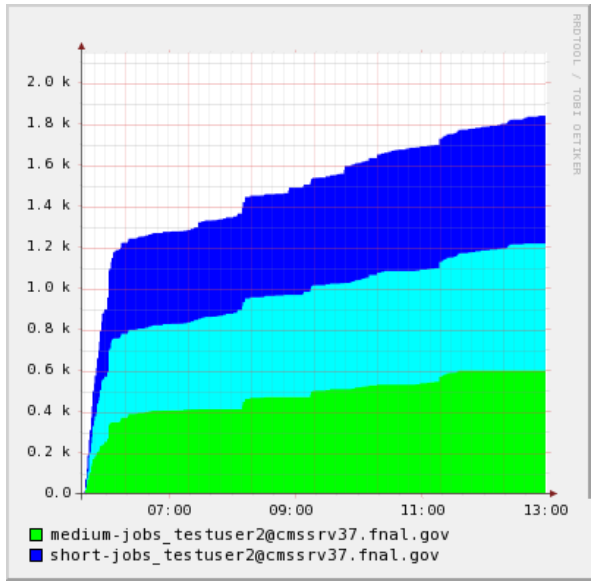
- Not so linear
  - Linear with up to 7k jobs in the queue
  - Chocked with 20k in the queue



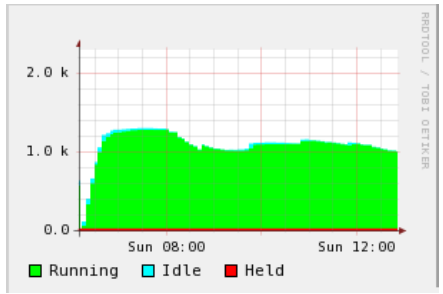
Condor-G running



CE queue



Condor-G running



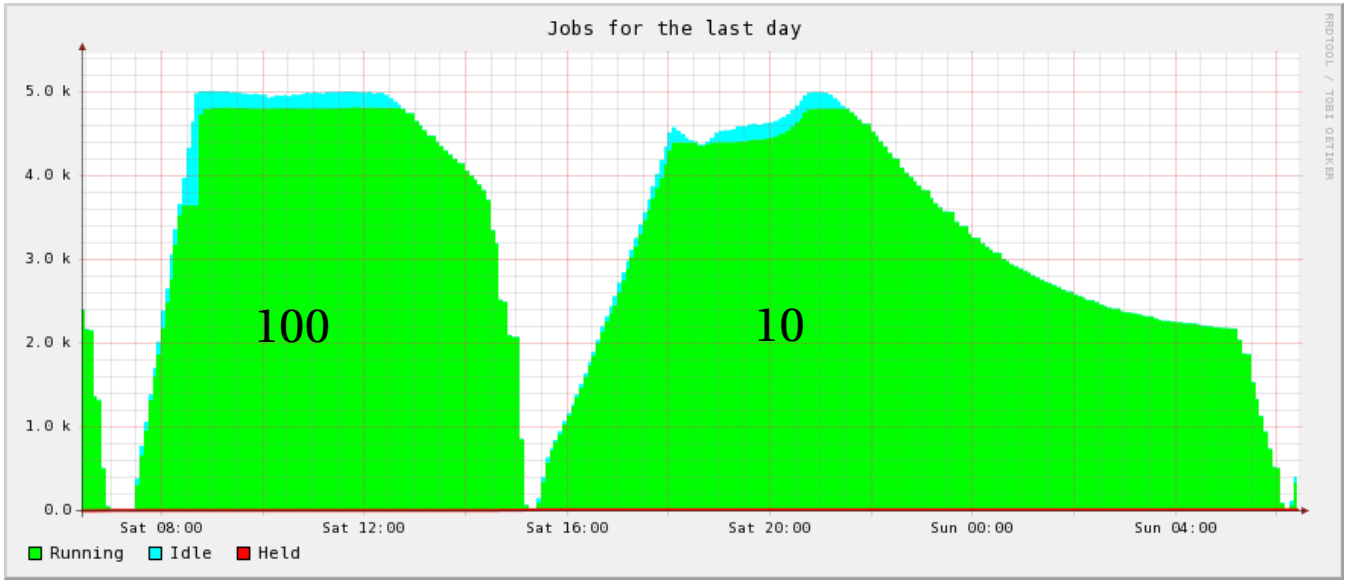
CE queue

All further tests done with at most 7k jobs in the queue

condor_gridmanager was using 100% of the CPU

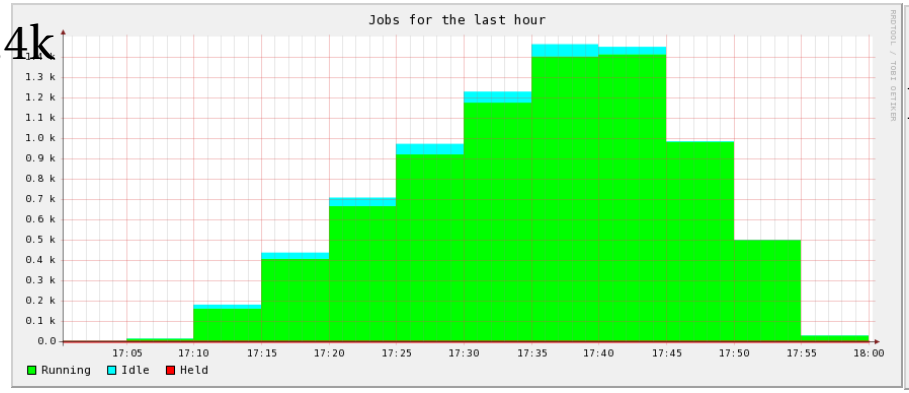
Condor-G startup speed(2)

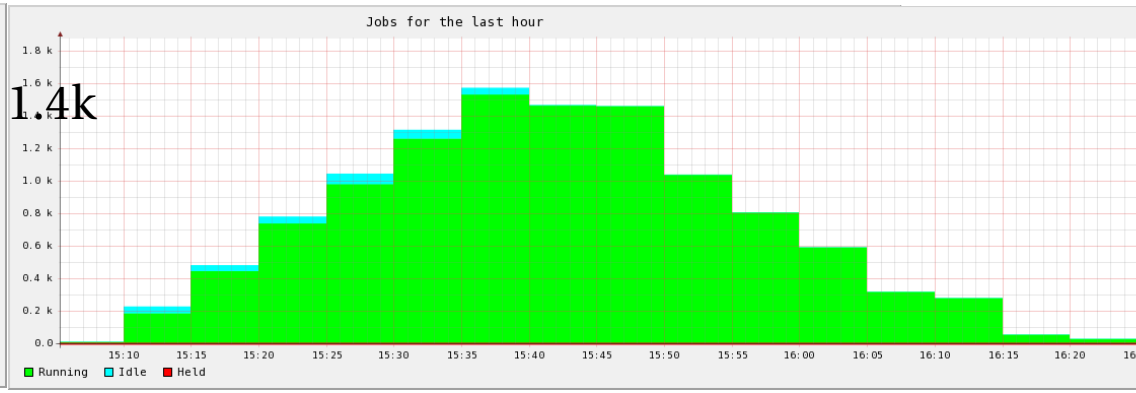
- Startup speed a load tradeoff
  - With `GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE = 100`
    ~60 jobs per minute
  - With default `GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE = 10`
    ~30 jobs per minute

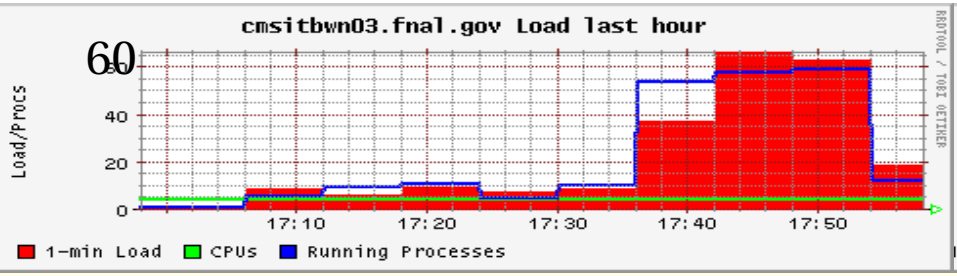


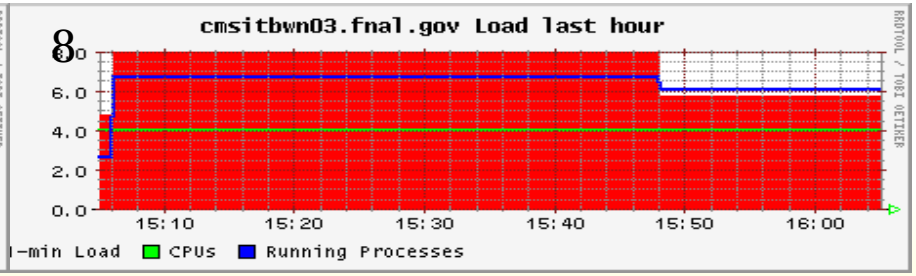
Condor-G removal speed

- Sometimes one does need to delete a large number of jobs
- Similar results as startup, but higher price diff
  - Again ~60 with `GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE = 100`
  - And ~30 with default `GRIDMANAGER_MAX_JOBMANAGERS_PER_RESOURCE = 10`



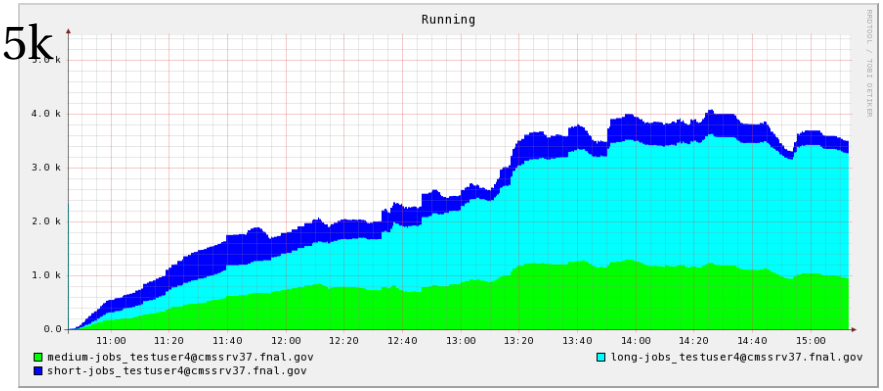




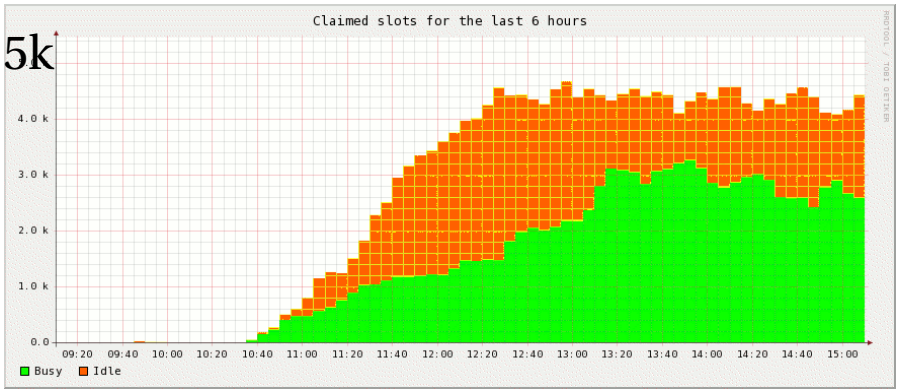


Condor-G and job length

- Given enough jobmanagers, Condor-G can saturate a big site
  - With 100 jobmanagers, 5k available slots and a job mix of 10min, 50min and 2h jobs, the limiting factor was the CE batch system



Condor-G running

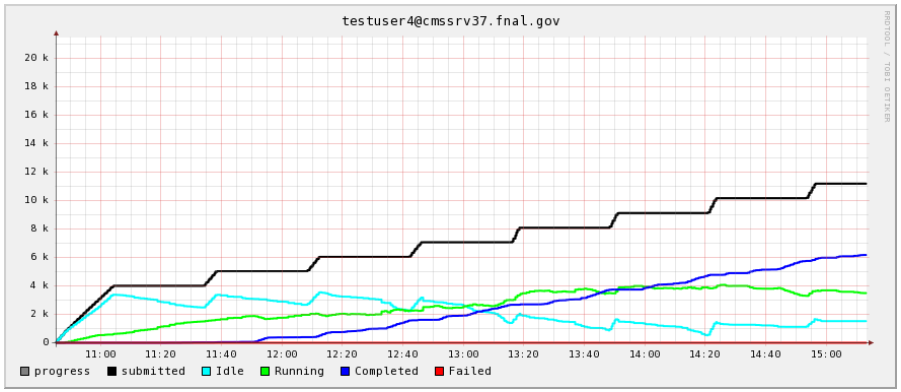


CE status

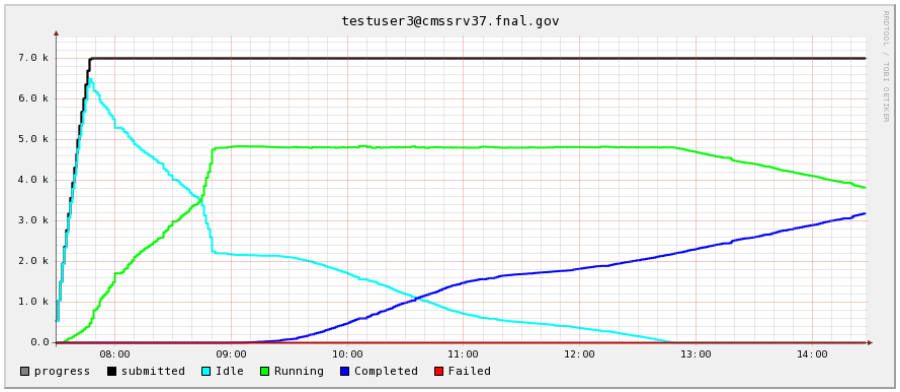
Idle slots
Busy slots

Condor-G reliability (1)

- Submitting to dedicated, well behaved site
  - No problems encountered if less than 7k jobs in queue
  - Given enough jobmanagers, both short and long jobs ran successfully



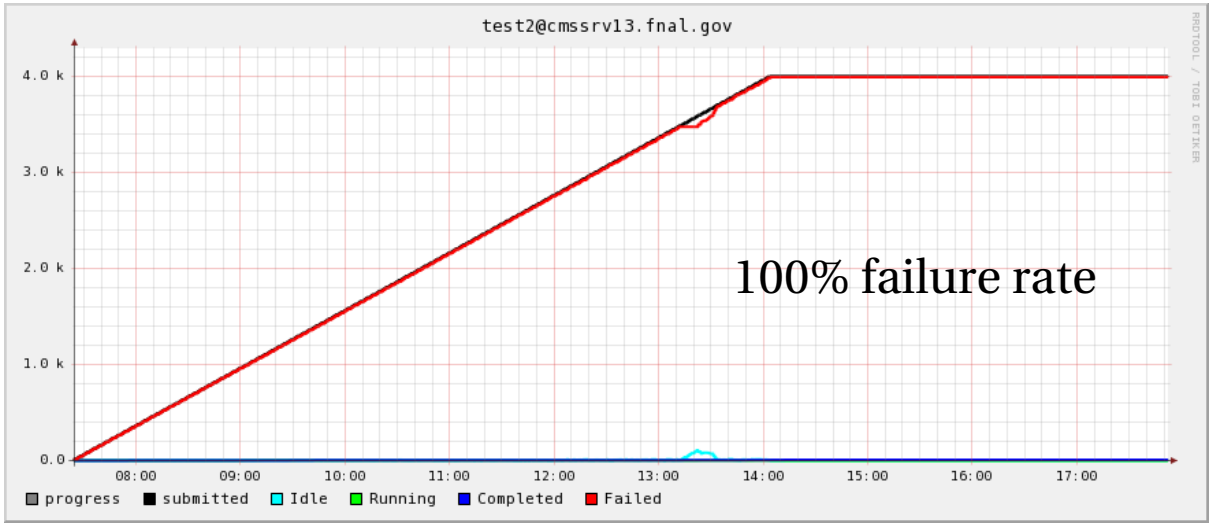
Mix of 10min, 50min and 2h jobs



Mix of 2h, 12h and 18h jobs

Condor-G reliability (2)

- But misconfigured sites can destroy most of your jobs



This same site worked perfectly just the day before!

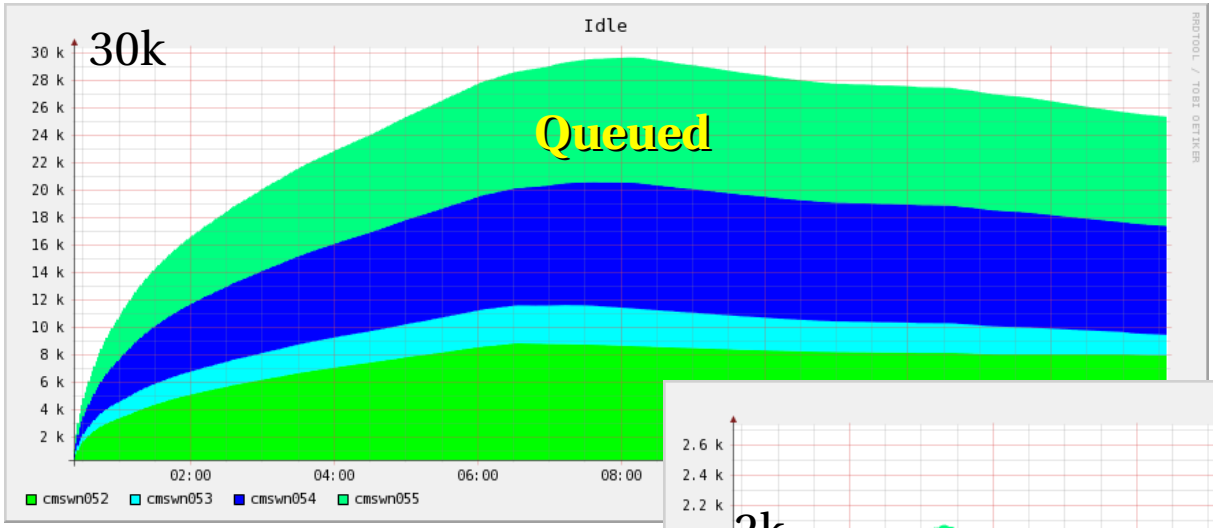
- Condor-G also does not handle well CE crashes
  - Jobs may stay in the queue forever

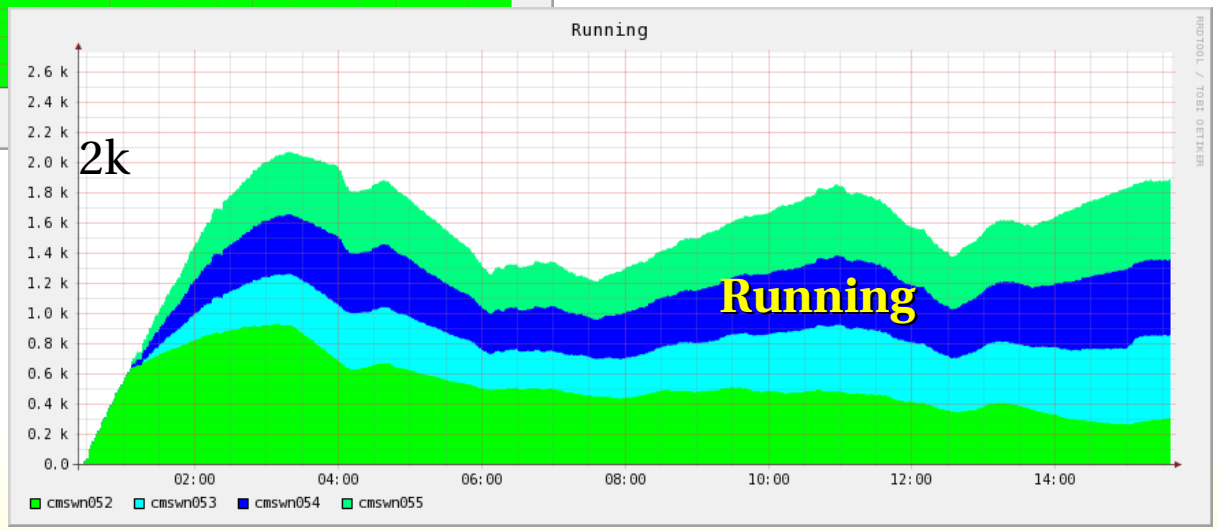
ReSS

- OSG **Re**source **S**election **S**ystem is a matchmaking system for Condor-G
  - Uses information gathered from via CEMon from Grid sites to make decisions
- The submission is still via the local Condor-G queue
  - ReSS choses the site to which to submit
  - Local Condor-G then handles the submission process

ReSS benchmarks

- Essentially the same as plain Condor-G
- Tested with 4x10k queued





2k slots on Grid site
default Condor-G parameters

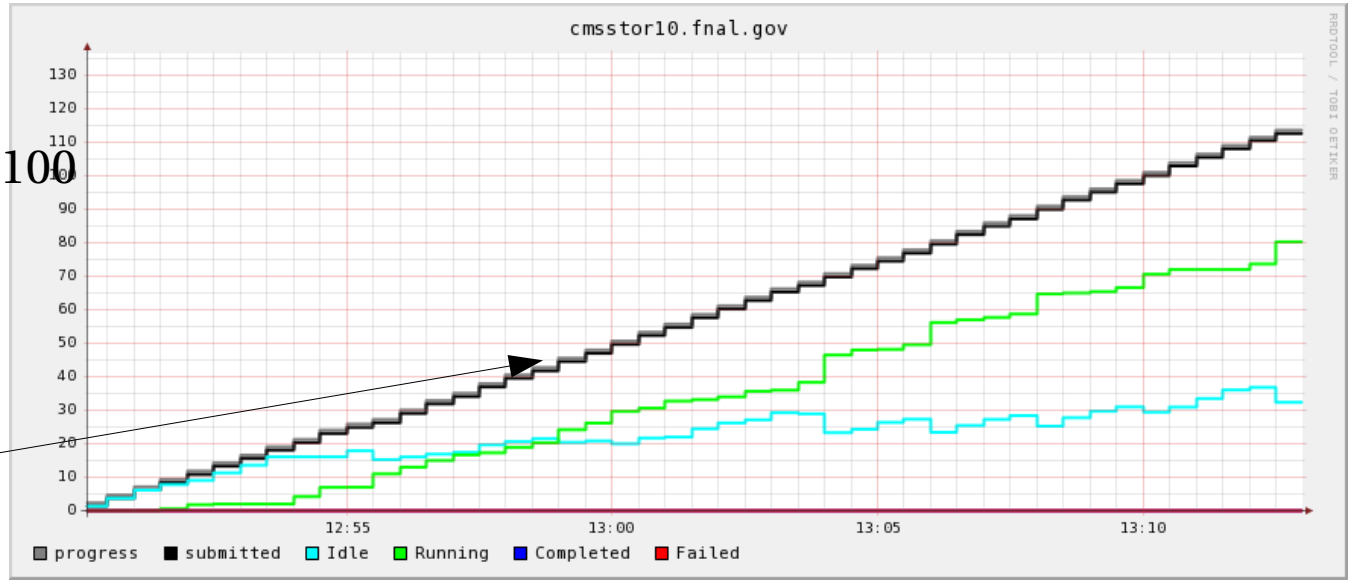
gLite WMS

- A portal solution
  - Users submit to the WMS and the WMS takes care of the submission
  - A dedicated client used for interaction
- Proprietary resource selection service inside
  - Uses BDII to gather information about the Grid sites
- Uses Condor-G internally for job submission

gLite WMS submission speed

- Way too slow
  - Only about 5 submission per minute

Even using advance delegation



Nr. submitted

20mins

Did not test anything else in single submission mode

gLite WMS collection mode(1)

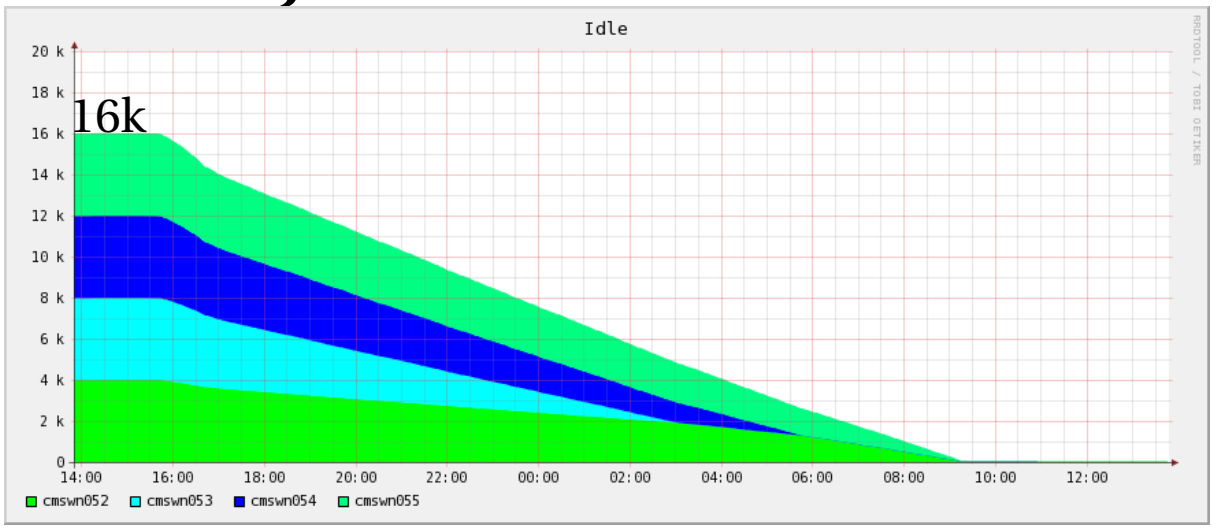
- In collection mode, the submission is very fast
  - ~1000 jobs submitted per minute
  - Collection made of 5k jobs submitted in 5 mins, a 20k collection took 23 mins
- Frequent temporary overloads
  - Only 7 out of 8 consecutive 5k submissions succeeded
    - Last one complained the WMS was overloaded (load>10)
  - Only 1 out of 8 parallel 5k submissions succeeded
    - The one that succeeded took 28mins
    - The others claimed the WMS was overloaded (load>10)
  - Going slowly, able to submit 4x20k collections

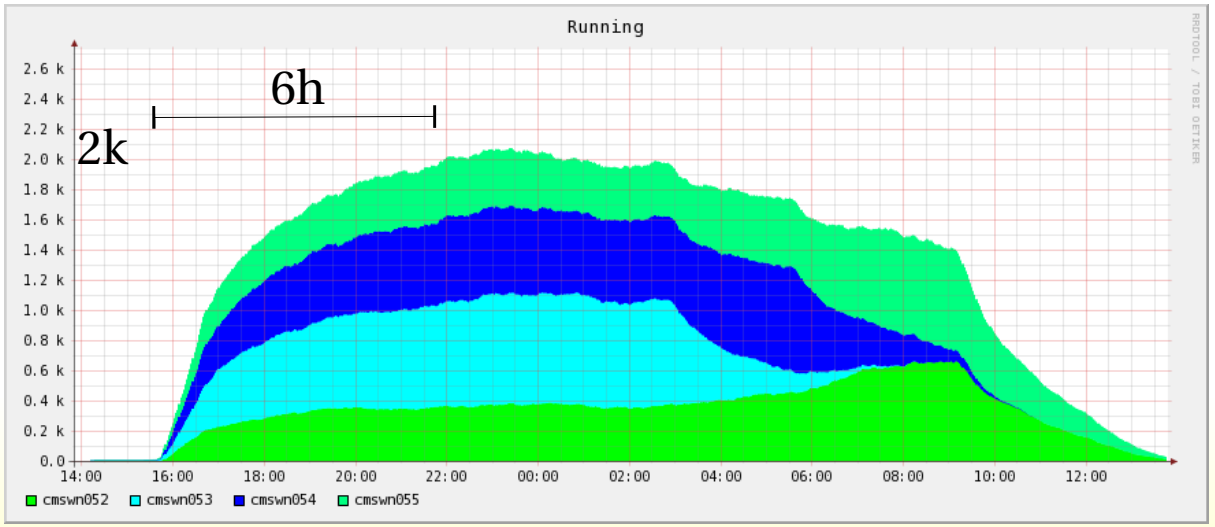
gLite WMS collection mode(2)

- Getting the status of the job
  - A 5k collection takes 40s to query
  - A 20k collection takes approx. 3mins
    - Single job in collection takes a few seconds, on average
    - However, had to retry in several occasions (timeouted)
  - Provided you know the job identification string
    - Could not find an easy way to obtain list of own jobs
- Removing a collection is reasonably fast
  - The 20k collection removal command returned in 30s
    - But don't know how long it took for the internal WMS cleanup
  - With 7x5k in queue, a single 5k removal took ~1min

gLite WMS startup speed

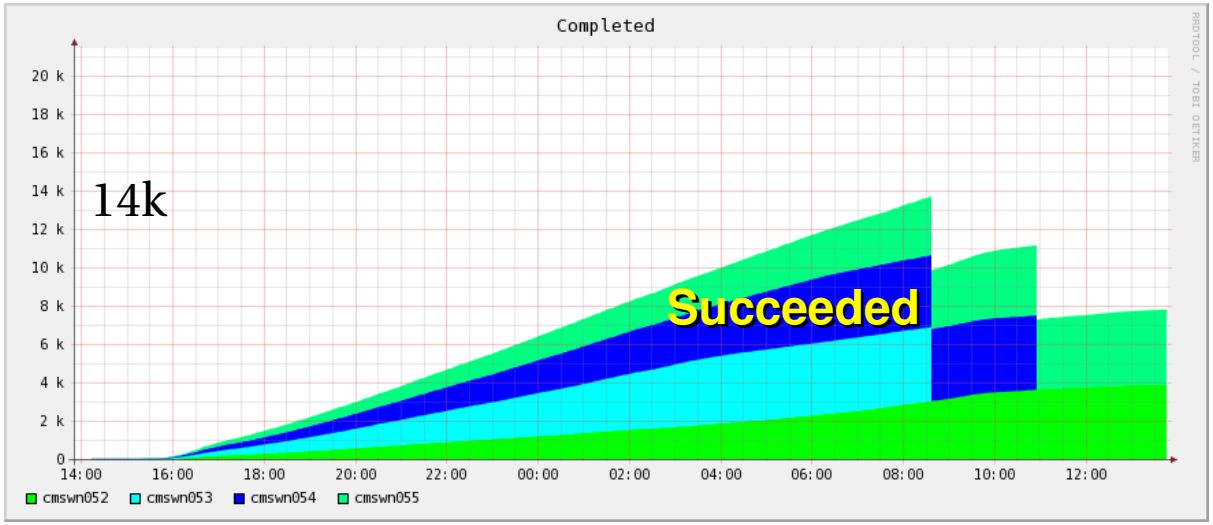
- Have only results from Jan'07
- Tested 4x4k
  - Would not scale past that
- Using 2 sites, started max 20 jobs per min

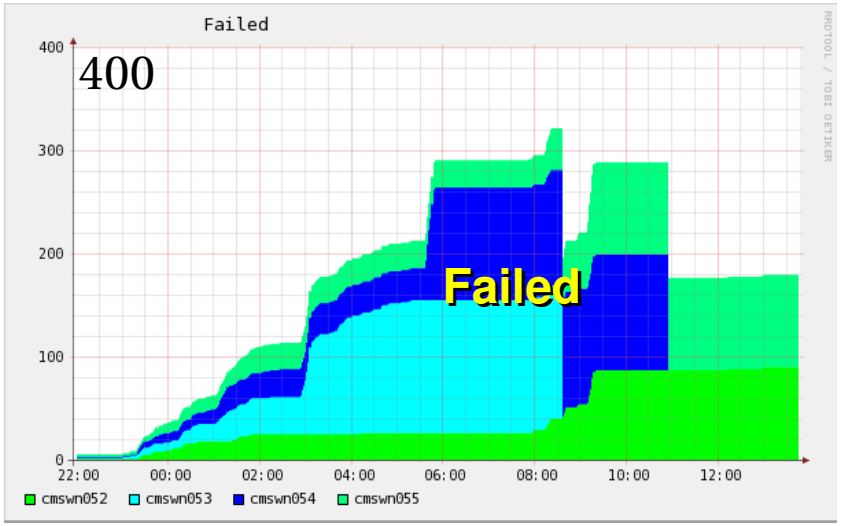




gLite WMS reliability

- Internally uses Condor-G, so most problems the same
  - But it does retry a job several times if first submission fails
    - Still several jobs failed in our tests



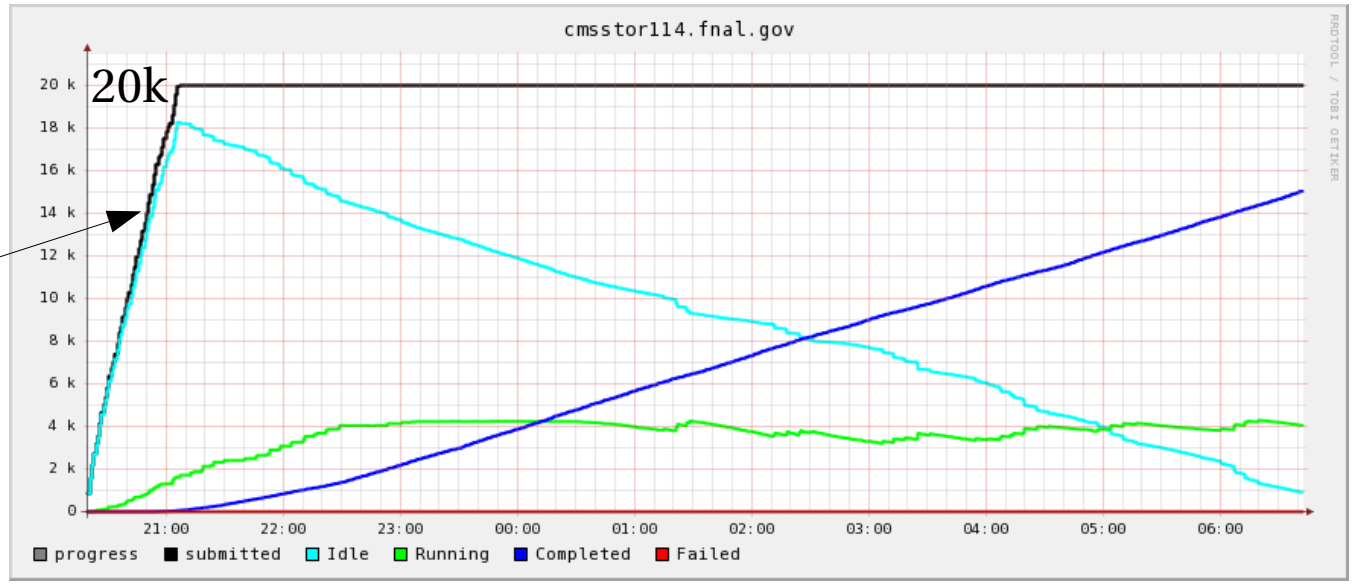


glideinWMS

- Condor glidein based WMS
  - Based on the pilot, or just-in-time philosophy
- User jobs are not directly submitted to the Grid
  - Instead, a Condor daemon (glidein) is submitted to the Grid using Condor-G
  - After the glidein registers back to the WMS, a user job with the highest priority is sent to that resource
- User jobs are usually vanilla Condor jobs
  - Although standard and MPI jobs are possible, too

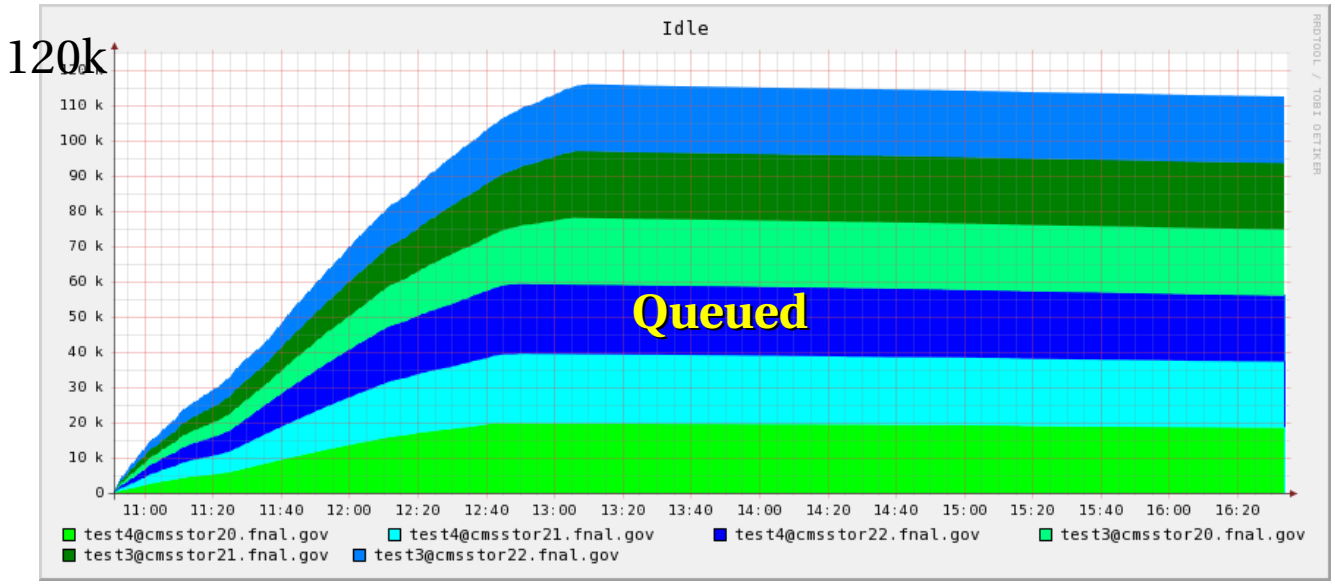
glideinWMS submission speed(1)

- Faster than Condor-G
  - Managed 400 submissions per minute



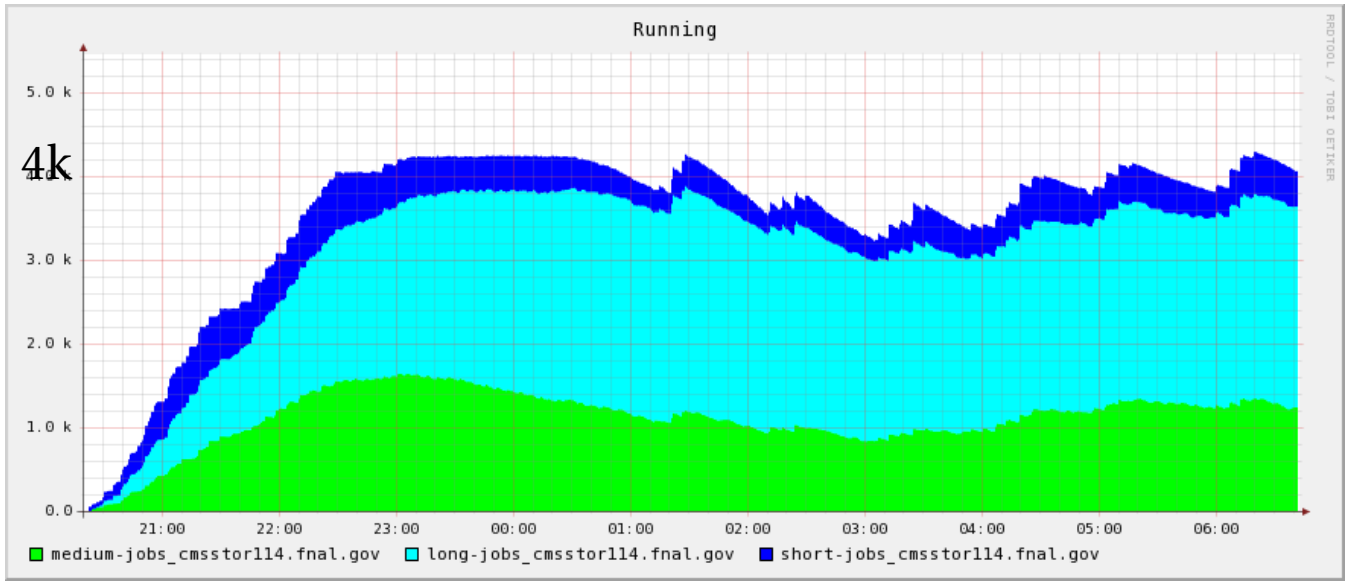
glideinWMS submission speed(2)

- Scales almost linearly with the number of submit schedds



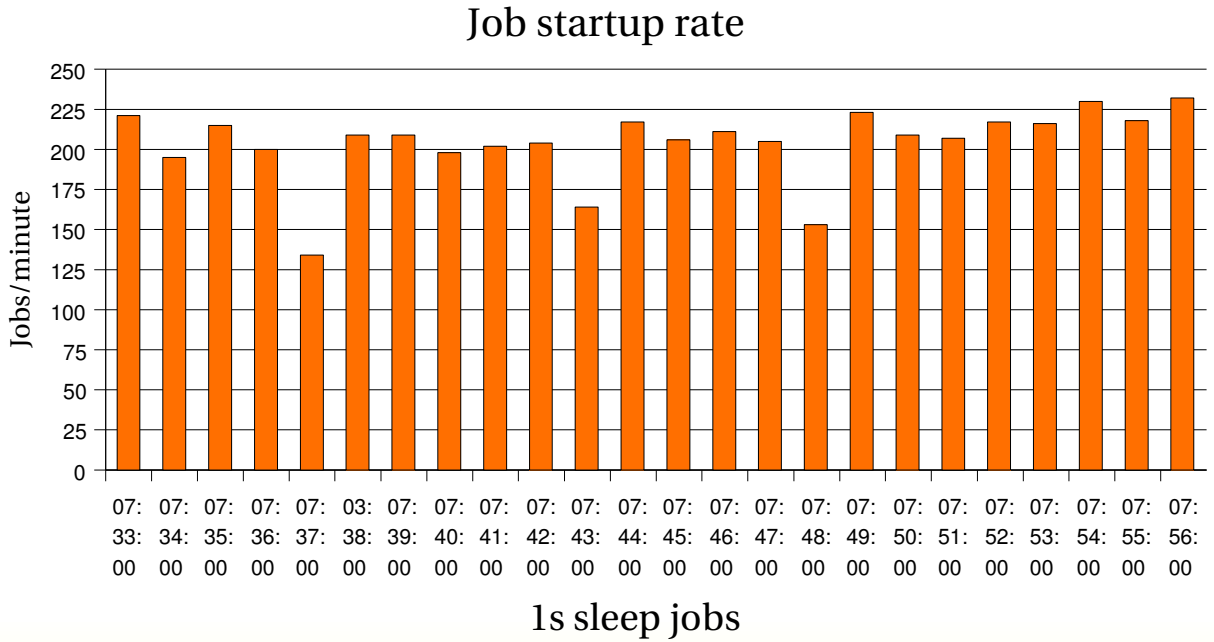
glideinWMS startup speed(1)

- Initially same as Condor-G
  - ~30 jobs per minute
  - Limited by the rate of glidein submission



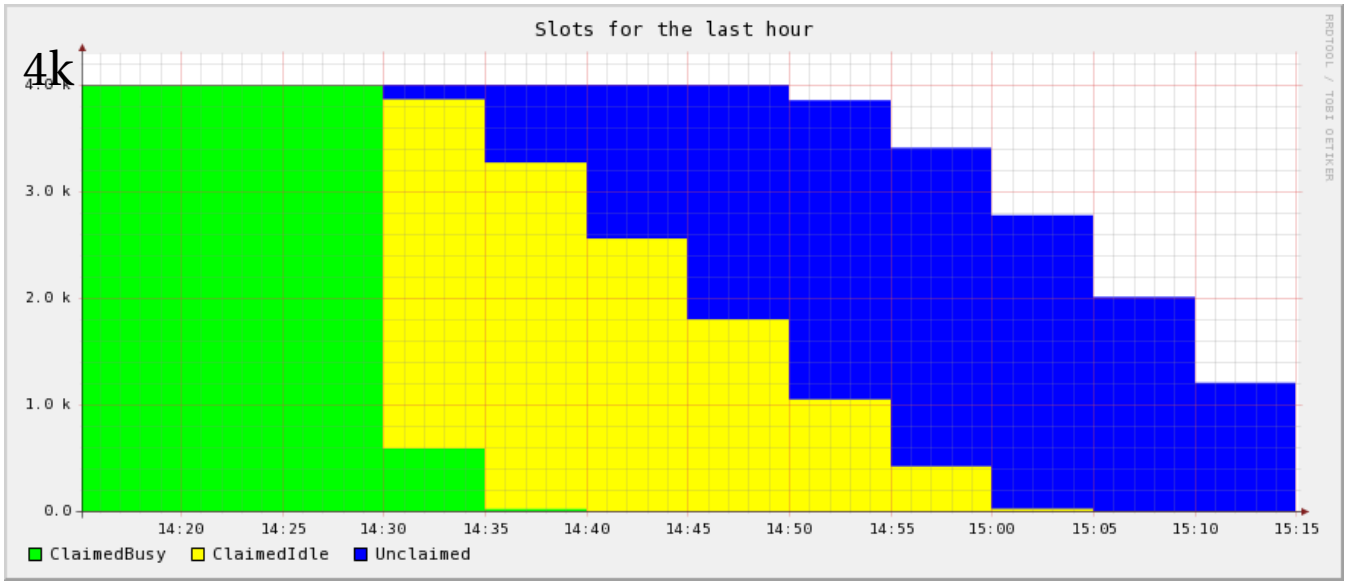
glideinWMS startup speed(2)

- Once the glideins have started, startup rates much higher
  - ~200 jobs per minute



glideinWMS removal speed

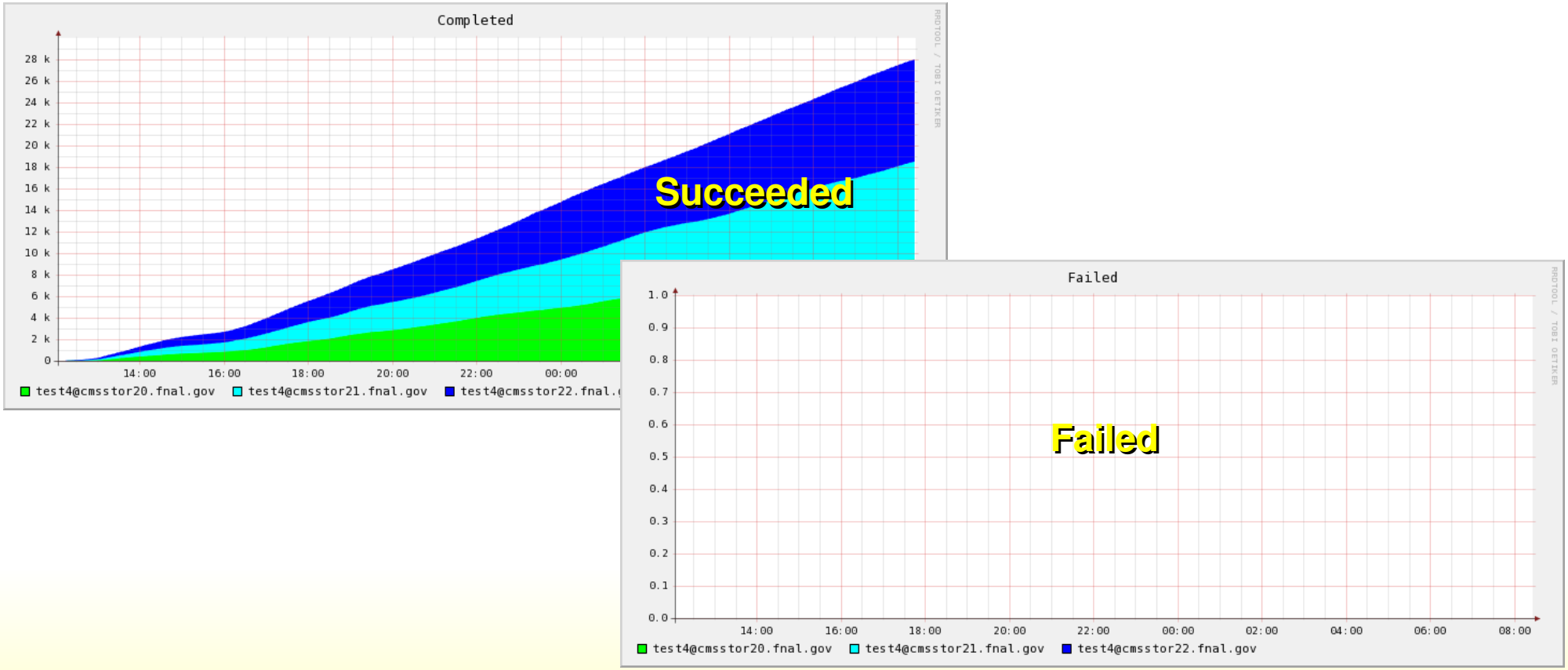
- condor_rm returns within the second, even if removing 20k jobs at once
- Used slots are released at ~120 jobs per minute



- glideins die after 20 minutes if no new jobs

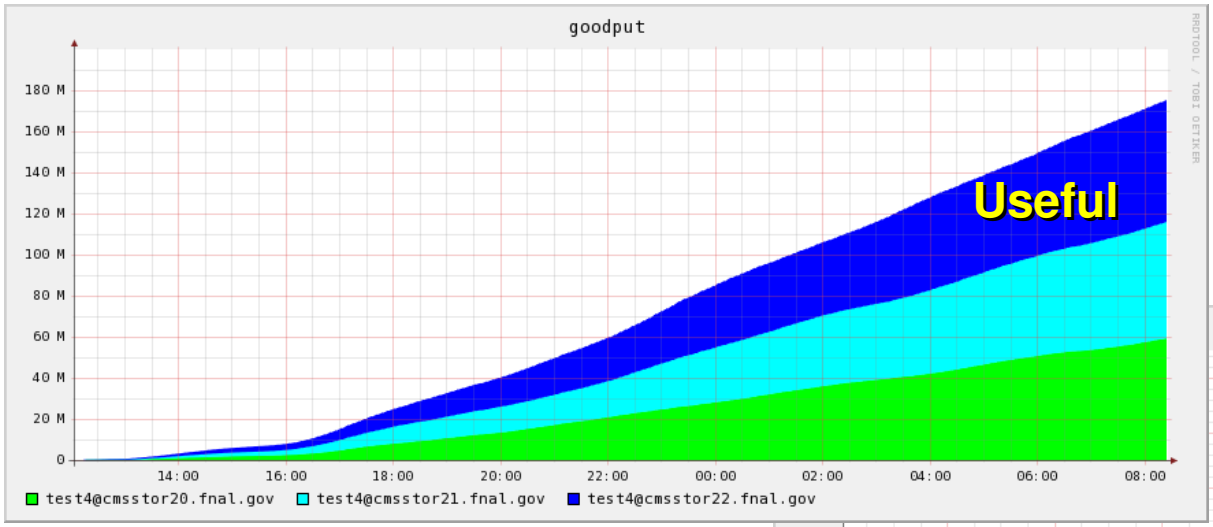
glideinWMS reliability(1)

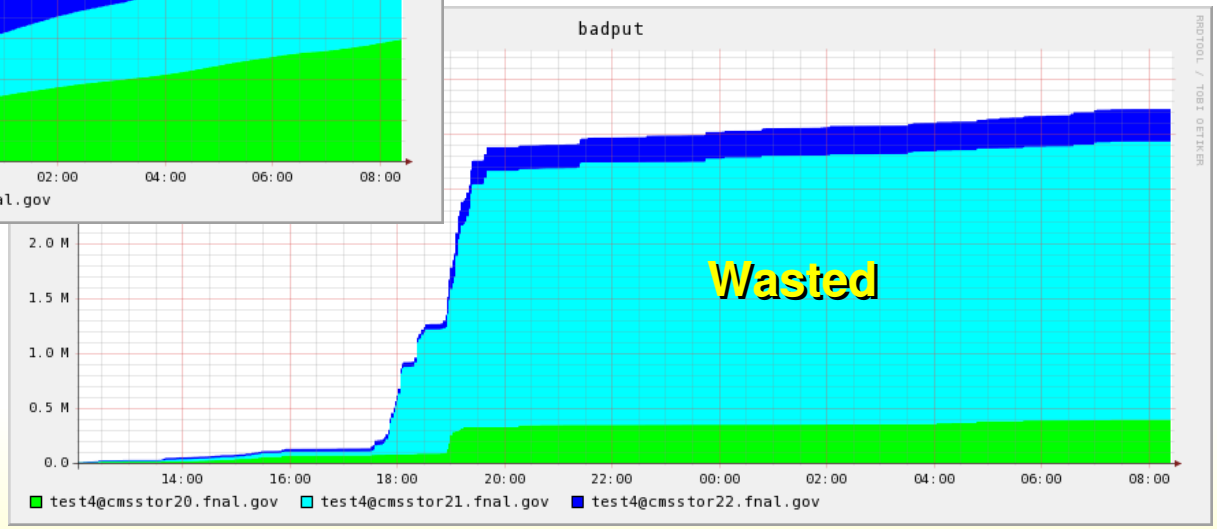
- User jobs almost never fail
  - Problematic Grid sites/nodes kill glideins not user job



glideinWMS reliability (2)

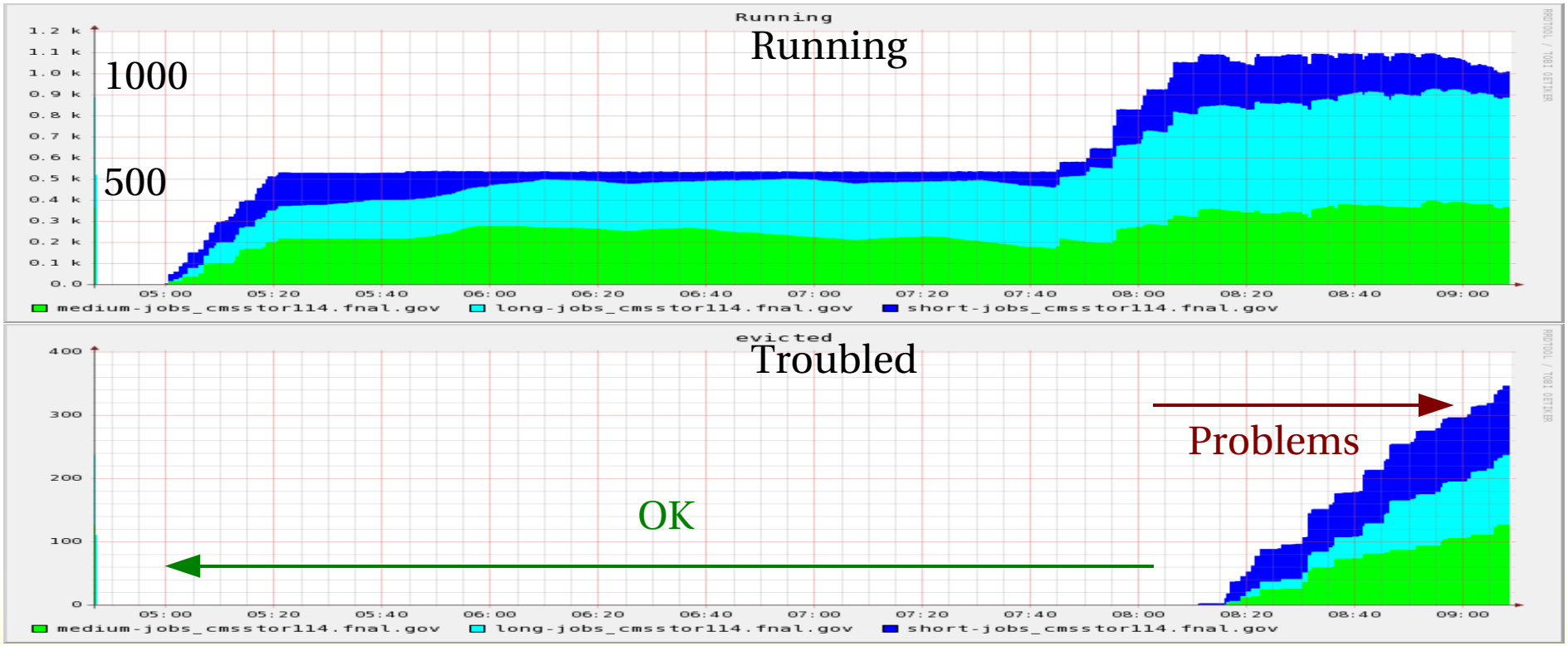
- If glidein dies after job started, Condor will restart the user job in another glidein
  - Just wasted CPU (Checkpointing can eliminate it)





GCB scalability

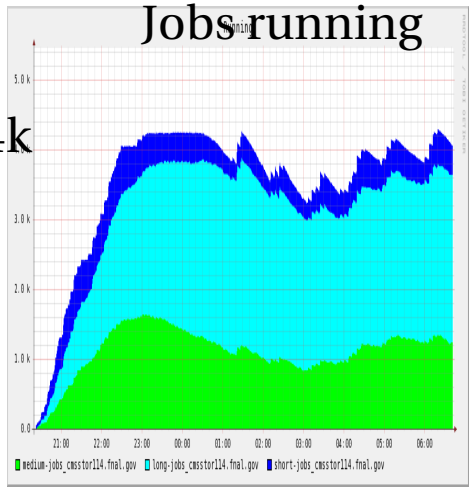
- Single GCB/schedd pair have limited scalability
  - Stable only to ~600 running jobs
  - Even if GCB configured to support 3k+ glideins (i.e. 100x200 connections)

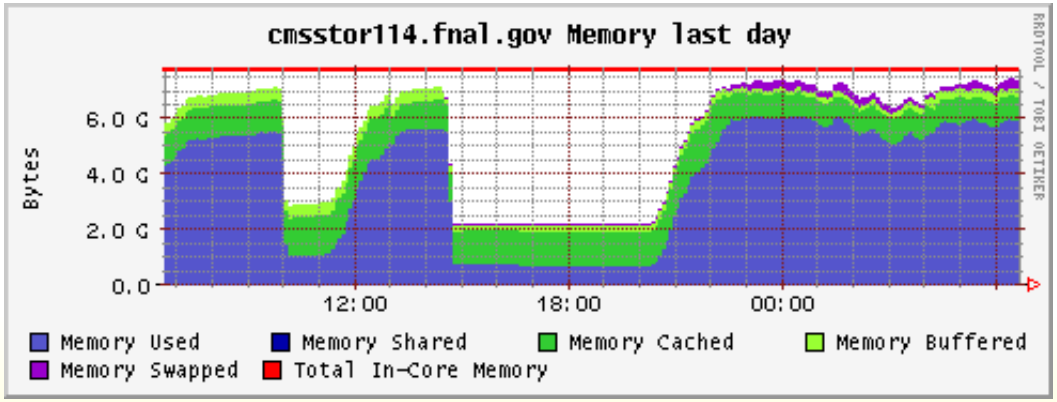


Must install many to scale

Condor memory usage

- Condor uses ~1.3Mb of memory per running job for the condor_shadows





Must install high amount of memory or use several submit nodes

glideinWMS daemons load

- VO frontend uses up to a full CPU and up to 1GB of memory
- Glidein factory uses ~1/5 of a CPU and 20M of memory per served Grid site
  - Most load coming from monitoring
  - May need to deploy several if you want to serve all the Grid sites

WMS comparison(1)

- The amount of service offered varies
  - Condor-G and ReSS provide just basic Grid submission and job retrial
  - gLite WMS is a black box, portal solution
    - Centralized optimizations
  - glideinWMS is a pilot based WMS
    - just-in-time scheduling
    - node validation and environment preparation
    - active job management

WMS comparison(2)

- Approach to resource selection varies:
  - Condor-G relies on the user to select a site
  - ReSS relies on CEMon information from the sites
  - gLite WMS relies on BDII information from the sites
  - glideinWMS schedules user jobs only after the glideins start
    - To submit the glideins the WMS admin can use any of the above for configuration purposes

WMS comparison(3)

- Different amount of investment is needed for each of them:

WMS	Client	Server
Condor-G	Light daemon	None
ReSS	Light daemon	Light daemon
gLite WMS	None	2 high end nodes (SL3 only)
glideinWMS	Heavyweight daemon	Several daemon nodes

WMS comparison(4)

- They have different scalability limits
  - Condor-G scales well, providing you install multiple schedds
  - ReSS seems to have similar scalability potential as Condor-G
  - gLite WMS unusable in single job submission mode
    - Users must group jobs in collections
    - Seems to scale, but experienced temporary overloads
  - glideinWMS seems to scale well, but you need to dedicate a lot of hardware to it

Conclusions

- I hope this overview was useful
- As you can see, each system has its own strengths and weaknesses
- Your needs may vary, so will not give any recommendations