GlideinWMS for ATLAS

- Rod Walker, LMU Munich 26th Jan 2012

- Motivation
- GlideinWMS overview
- ATLAS test bed and experience
- Alternative
- Next steps

Motivation

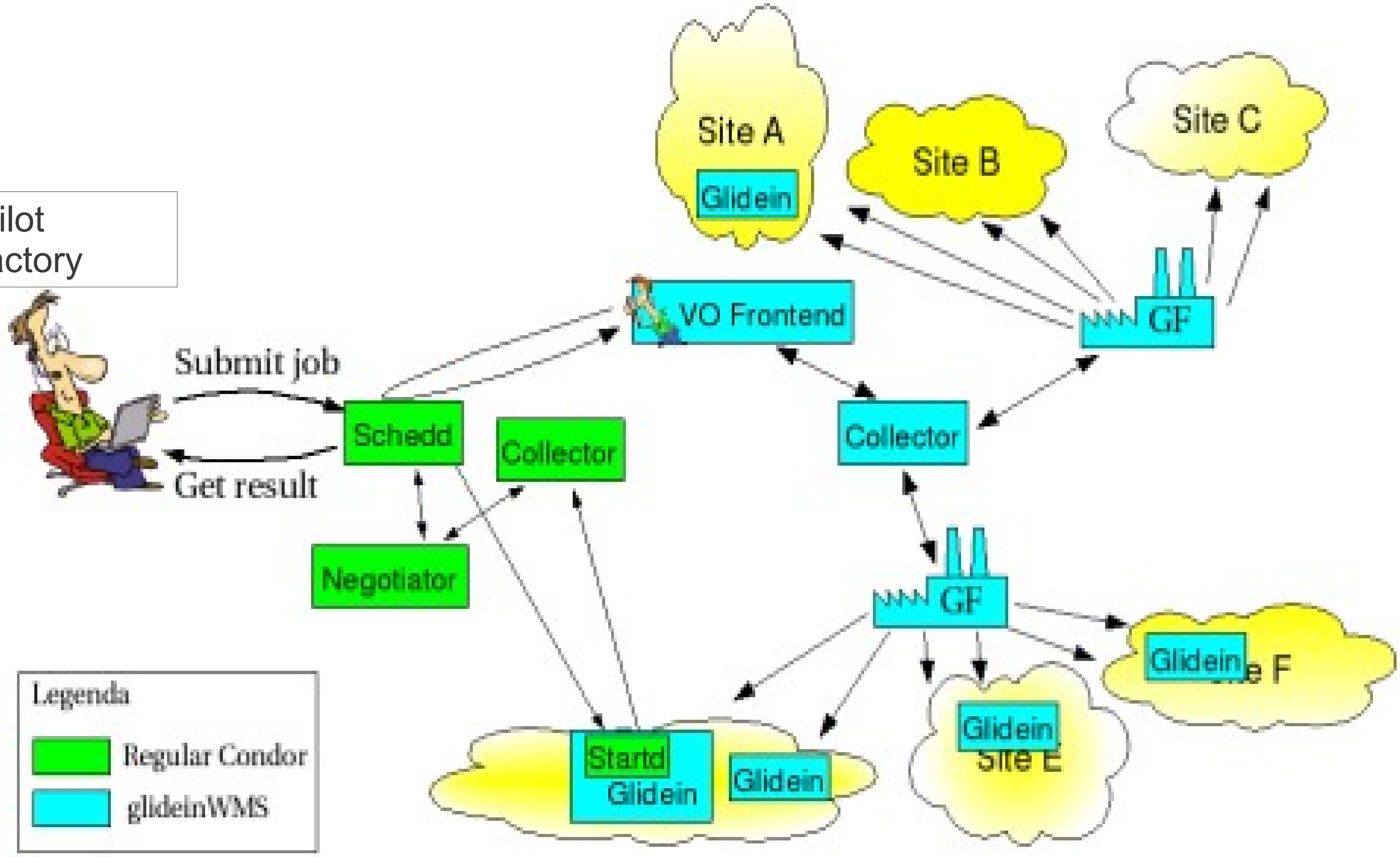
- Historical “Cronus” project did not die a natural death
  - Some Condor instability(GCB) and scalability issues
    - since addressed thanks to glideinWMS connections and persistence
- Glexec usage almost free
  - NIKHEF white-hats will not let it go
- Other Condor freebies
  - Whole-node scheduling, ssh-to-job, fairshare, preemption, prd/analy split

GlideinWMS

- Result is a distributed Condor pool
  - Looks and feels like local Batch system
- Underneath are Condorg jobs to start glideins
  - Startd deamons connect to Condor pool
- Tricky bit is auto-starting glideins, based on queued vanilla jobs
  - Negotiation, proxy transfer, ...
- Pilot is vanilla job with JDL to specify site
- Condor negotiation decides which pilot to start

The big picture

Pilot factory



Fit into Panda

- Schedd is on the current factory machines
  - also run 1 Collector and Negotiator pair
- Submit panda pilot per site-user pair
  - based on activated jobs on panda queue
    - pull user proxy from MyProxy (and cache/renew)
    - vanilla condor job is pilot to run only one users jobs.
      - has user proxy by delegation.
- VO-FrontEnd can see factory schedd queues
  - UCSD submits glideins to service jobs in queues
    - only cares about site, not user
    - delay to start pilots bit more than now – polling schedd queue
    - no delay if existing glide-in becomes free

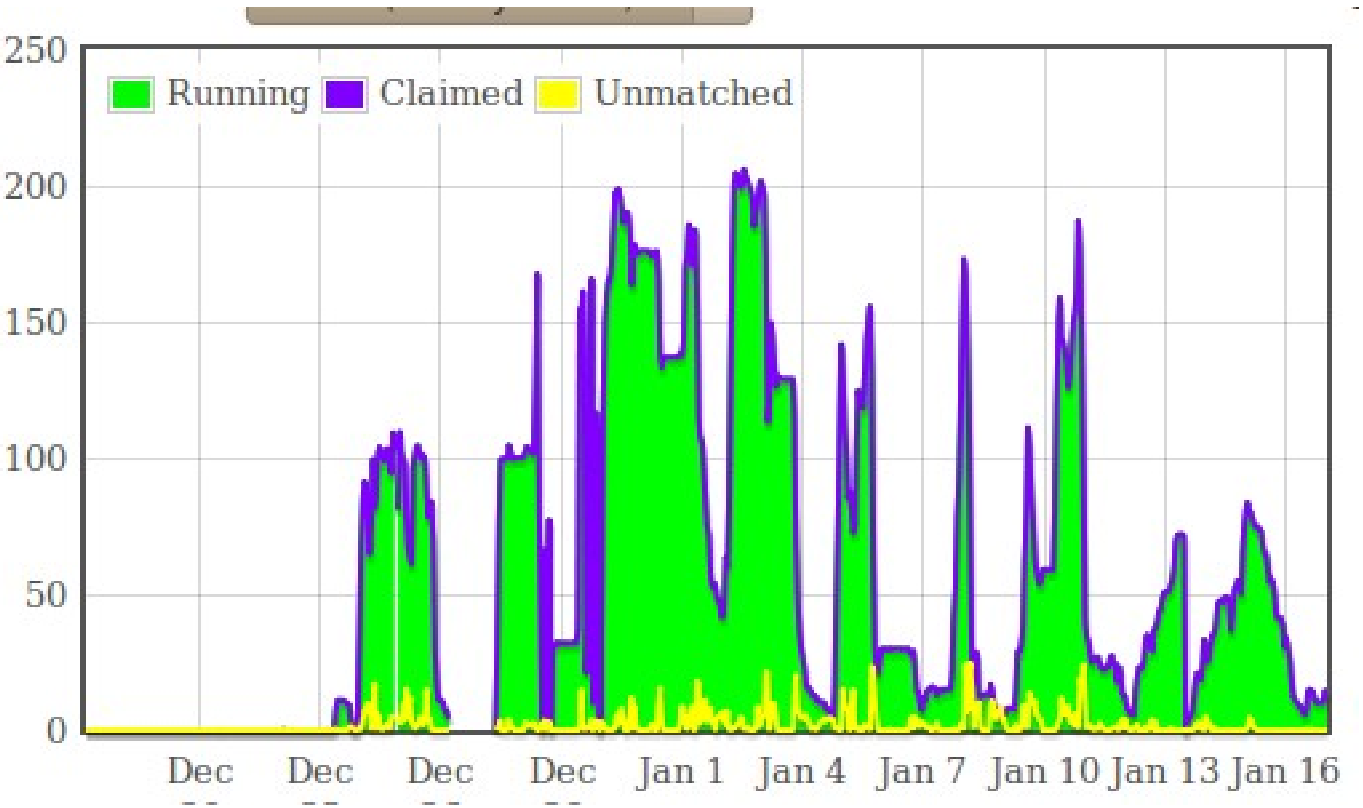
Testbed

- VO-Frontend at LMU linked to gFactory at UCSD
    - NIKHEF, Taiwan, KIT configured (UCSD)
    - Submit ANALY_NIKHEF pilots via cron factory
        - my pilot proxy used for gFactory (mapped to atlasplt)
        - Robot pilot proxy used for vanilla jobs
            - not users own proxy, or even my plain proxy
                - due to storage acls
        - Glexec identity switching working
- GFactory on voatlas92 but not fully working
    - not recommended by UCSD – they like multi-VO gfactories
        - plan to have CERN based ones though
- voatlas195 frontend configured for voatlas92 gfactory
    - switch to UCSD and run dev pilot service from here
    - Aside: same Condor instance submits pilots to Startd's in Clouds

Monitoring

http://gar-ex-etpgrid01.garching.physik.uni-muenchen.de/vofrontend/monitor/frontend_ATLAS-gar-ex-etpgrid01-v1_0/

ANALY_NIKHEF pilots on glideins
Menus and buttons for glidins,status,jobs,....



ToDo list

- User proxy stored in MyProxy
  - already the case for sites with 'glexec' schedconf
  - Rebrokerage possibility suggests should always fill MyProxy
- Panda client get jobs per user and queue
- Factory submits vanilla jobs using this info
- Ensure storage acls allow user proxy
  - currently need pilot role
  - Scratchdisk meant to be user writable
- Safe to run in mixed mode? Think about it.
  - Some sites glexec, some not
    - Different pilot proxies – only one can access MyProxy

Alternative

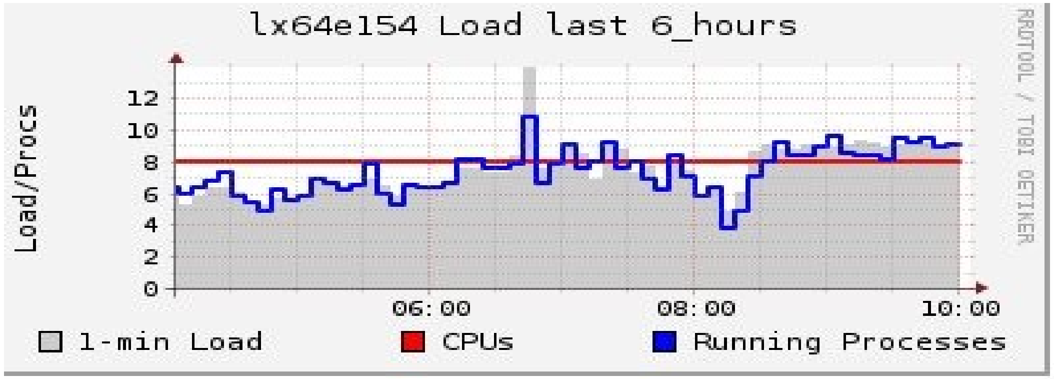
- Glexec in the pilot
  - After job pulled, get user proxy from MyProxy, and run payload under glexec
    - stage-in and out as pilot user
- Pros
  - get next job chosen by Panda fairshare
  - no additional pilot layer
- Cons
  - pilot proxy can get user proxy delegation from anywhere
    - GliteWMS uses MyProxy accessible from limited hosts
    - Some shared-secret plan to patch this up
    - Possible scalability issue 600k jobs per day = 5Hz to MyProxy, with peaks
  - Complications in file ownership from mixing pilot with glexec user.
  - For whatever reason – not in production
- GlideinWMS cleaner, general solution with bonus features

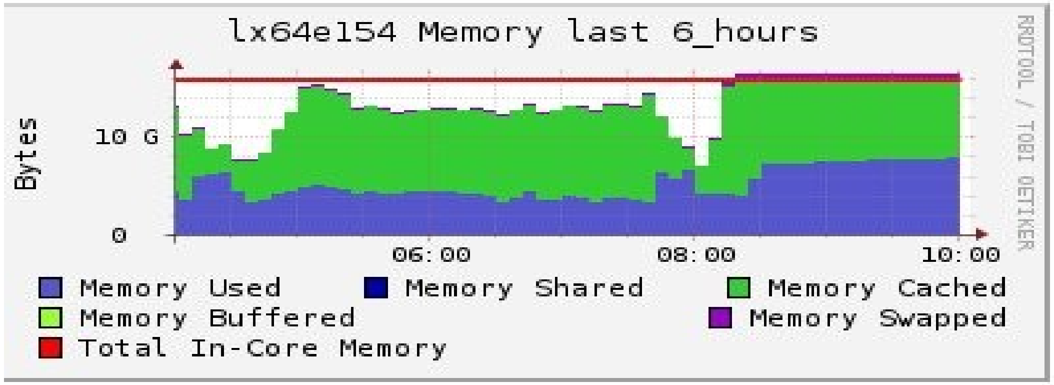
Bonus 1: Fair Share

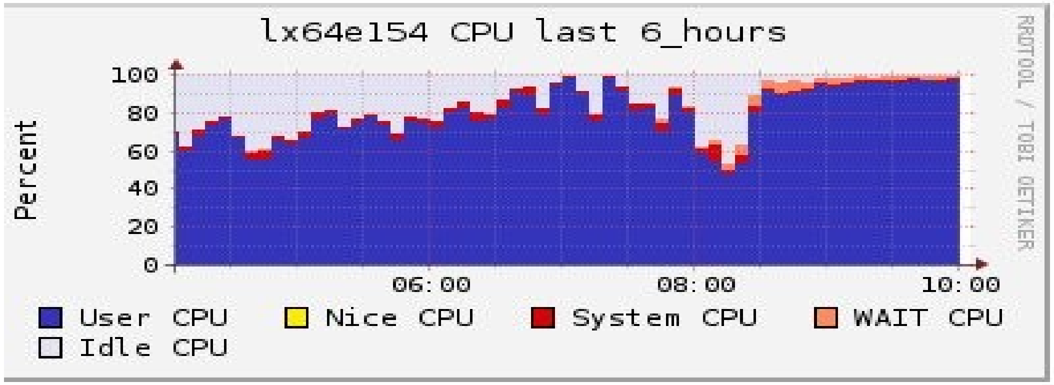
- Condor chooses when user gets next job start
  - Panda priority selects which of that users jobs
- Global fair share built into Condor
  - has group shares, supports VOMS group/roles
  - Panda already has one – Condor possibly fairer, more transparent. Need investigation.
- Run prod on glideins too then
  - share between prd/analy fully dynamic
  - maybe boost analysis in EU daytime

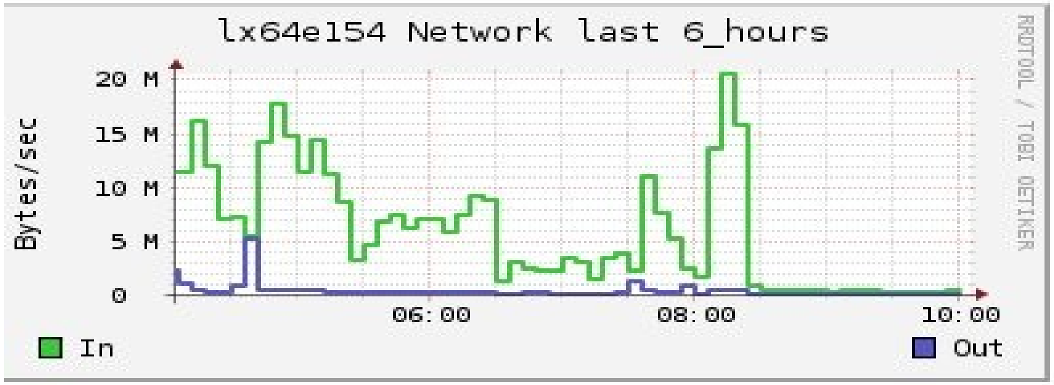
Bonus 2: Whole-node scheduling

- 1 Startd glide-in to manage 16 cores,say
  - very easy to run 16 serial jobs
  - Scheduling is key. When to run more jobs, run a short job, retire node,... panda pilot cannot do this
    - rich condor config logic
- Pack a node with mixed workload
  - 8-core athena MP job plus 8 single core job
    - config for this on condor wiki
  - 6 io limited analysis + 6 G4 on 8 cores
    - no need to know in advance. Use node load/usage info in matchmaking, e.g. room for 1 more.
- No real benefit for 1 job per multi-core slot



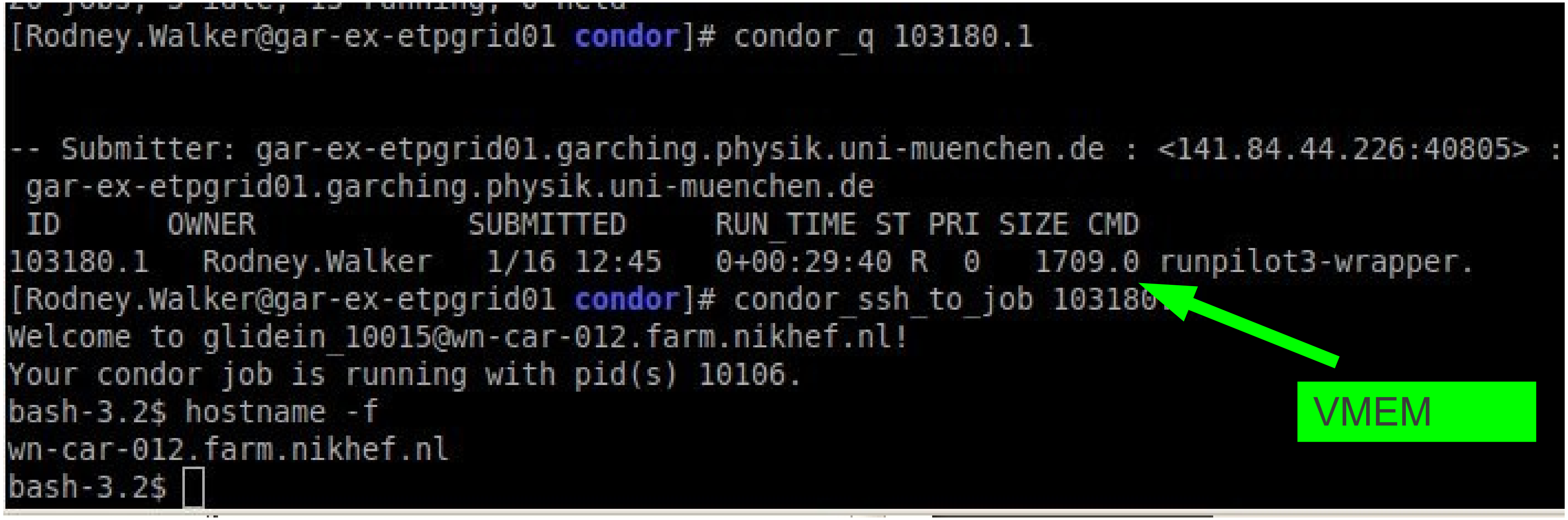






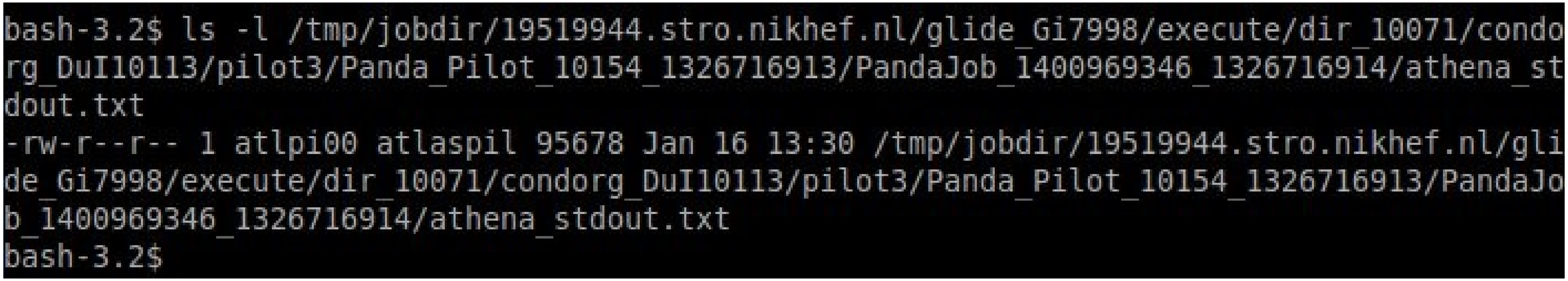
Fine optimization of cpu usage
8 core node with 8 slots
2-3 analysis jobs running up to 08:30
Then 8 G4 jobs
If Condor had whole node, could add G4 job or 2 – when analysis finishes, no new G4 started.

Bonus 3.Cool stuff – ssh to your job



```
[Rodney.Walker@gar-ex-etpgrid01 condor]# condor_q 103180.1


-- Submitter: gar-ex-etpgrid01.garching.physik.uni-muenchen.de : <141.84.44.226:40805> :
 gar-ex-etpgrid01.garching.physik.uni-muenchen.de
 ID       OWNER            SUBMITTED     RUN_TIME ST PRI SIZE CMD
103180.1   Rodney.Walker    1/16 12:45   0+00:29:40 R  0   1709.0 runpilot3-wrapper.
[Rodney.Walker@gar-ex-etpgrid01 condor]# condor_ssh_to_job 103180
Welcome to glidein_10015@wn-car-012.farm.nikhef.nl!
Your condor job is running with pid(s) 10106.
bash-3.2$ hostname -f
wn-car-012.farm.nikhef.nl
bash-3.2$ 
```



```
bash-3.2$ ls -l /tmp/jobdir/19519944.stro.nikhef.nl/glide_Gi7998/execute/dir_10071/condo
rg_DuI10113/pilot3/Panda_Pilot_10154_1326716913/PandaJob_1400969346_1326716914/athena_st
dout.txt
-rw-r--r-- 1 atlpi00 atlaspil 95678 Jan 16 13:30 /tmp/jobdir/19519944.stro.nikhef.nl/gli
de_Gi7998/execute/dir_10071/condorg_DuI10113/pilot3/Panda_Pilot_10154_1326716913/PandaJo
b_1400969346_1326716914/athena_stdout.txt
bash-3.2$
```

Secure using X509 proxy. Traverses firewall using CCB (outbound only). Can be disabled – probably made admin-only)
HLT bpeek++ to stuck jobs. Users can look(and act) before they complain.
NIKHEF people will have a fit, but it is secure and nothing one could not do in a script.

Conclusion

- Would like to scale up tests by making some glexec sites use only GlideinWMS
  - NIKHEF, then maybe KIT, HH, CERN
  - Still multi-user pilots – just exercising glexec
  - include prod pilots to increase scale
- Full deployment pending scaling and stability tests
  - I hear 25k per Schedd and scales with # schedds
    - single Negotiator and Collector instance
- Ask for ToDo list
  - Panda client(per user), autopyfactory(proxy cache), storage acls
- Setup fairshare for users, groups and prod.
  - all prod and analy run via glideinwms on 1 site
- Formalize gFactory support model with CERN/OSG/UCSD