Two Years of HTCondor at the RAL Tier-1

Andrew Lahiff, Alastair Dewhurst, John Kelly, Ian Collier

STFC Rutherford Appleton Laboratory

2015 WLCG Collaboration Workshop

Science & Technology Facilities Council

GridPP
UK Computing for Particle Physics

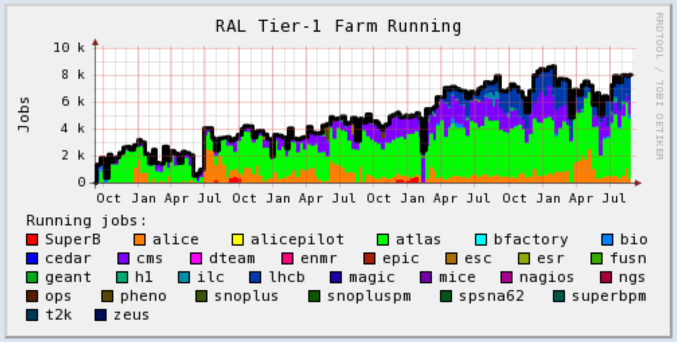
Outline

- Introduction
- Migration to HTCondor
- Experience
- Functionality & features
- Monitoring
- Ongoing work & future plans

GridPP UK Computing for Particle Physics

Introduction

- RAL is a Tier-1 for all 4 LHC experiments
  - Also support ~12 non-LHC experiments, including non-HEP
- Computing resources
  - 560 worker nodes, over 12K cores
  - Generally have 40-60K jobs submitted per day



Increase in running robs while running Torque + Maui

Torque + Maui

- Torque + Maui had been used for many years at RAL
- Many issues
  - Severity and number of problems increased as size of farm increased
- Problems included
  - pbs_server, maui sometimes unresponsive
  - pbs_server needed to be restarted sometimes due to excessive memory usage
  - Job start rate sometimes not high enough to keep the farm full
    - Regularly had times when had many idle jobs but farm not full
  - Regular job submission failures on CEs - *Connection timed out-qsub: cannot connect to server*
  - Unable to schedule jobs to the whole-node queue
    - We wrote our own simple scheduler for this, running in parallel to Maui
  - Didn't handle mixed farm with SL5 and SL6 nodes well
  - DNS issues, network issues & problematic worker nodes cause it to become very unhappy
- Increasing effort just to keep it working

GridPP
UK Computing for Particle Physics

Choosing a new batch system

- In August 2012 started looking for an alternative – criteria:
  - Integration with WLCG community
    - Compatabile with grid middleware
    - APEL accounting
  - Integration with our environment
    - E.g. Does it require a shared filesystem?
  - Scalability
    - Number of worker nodes
    - Number of cores
    - Number of jobs per day
    - Number of running, pending jobs
  - Robustness
    - Effect of problematic WNs on batch server
    - Effect if batch server is down
    - Effect of other problems (e.g. network issues)
  - Support
  - Procurement cost
    - Licenses, support
    - Avoid commercial products if at all possible
  - Maintenance cost
    - FTE required to keep it running
  - Essential functionality
    - Hierarchical fairshares
    - Ability to limit resources
    - Ability to schedule multi-core jobs
    - Ability to place limits on numbers of running jobs for different users, groups, VOs
  - Desirable functionality
    - High availability
    - Ability to handle dynamic resources
    - Power management
    - IPv6 compatibility

GridPP UK Computing for Particle Physics

Choosing a new batch system

- Considered, tested & eventually rejected the following technologies:
  - LSF, Univa Grid Engine
    - Avoid commercial products unless absolutely necessary
  - Open-source Grid Engines
    - Competing products, not sure which has best long-term future
    - Communities appear much less active than SLURM & HTCondor
    - Existing Tier-1s using Univa Grid Engine rather than open-source
  - Torque 4 + Maui
    - Maui problematic
    - Torque 4 seems less scalable than alternatives
  - SLURM
    - Carried out extensive testing and comparison with HTCondor
    - Found that for our use case:
      - Very fragile, easy to break
      - Unable to get to work reliably above 6000 jobs slots

GridPP
UK Computing for Particle Physics

Choosing a new batch system

- HTCondor chosen as replacement for Torque + Maui
  - Has the features we require
  - Seems very stable
  - Easily able to run 16,000 simultaneous jobs
    - Prior to deployment into production we didn't try larger numbers of jobs
      - Didn't expect to exceed this number of slots within the next few years
    - Didn't do any tuning – it "just worked"

Migration to HTCondor

GridPP
UK Computing for Particle Physics

Migration to HTCondor

- Setup main components of new batch system
  (in parallel to the old batch system)
  - Central managers
  - A few worker nodes (old hardware)
  - CEs
- After initial testing
  - Added one generation of older worker nodes (~1000 cores)
    - Capacity beyond LHC MoU commitment
  - Requested ATLAS start submitting to the new CEs
    (in addition to the existing CEs associated with the old batch system)
  - Fixed any issues that came up
  - Later requested CMS start submitting to the new CEs

Migration to HTCondor

- While this testing was ongoing
  - Added monitoring
    - Nagios
    - Ganglia
  - Checked that APEL accounting was accurate & working
  - Wrote internal documentation
    - Service description, installation procedures, …
    - On-call documentation
- Next steps
  - Testing with ALICE, LHCb, & selected non-LHC VOs
- Once migration to HTCondor approved by Change Control team
  - Migrated 50% of CPU to HTCondor
  - Within a few months migrated remaining CPU

GridPP
UK Computing for Particle Physics

Migration to HTCondor

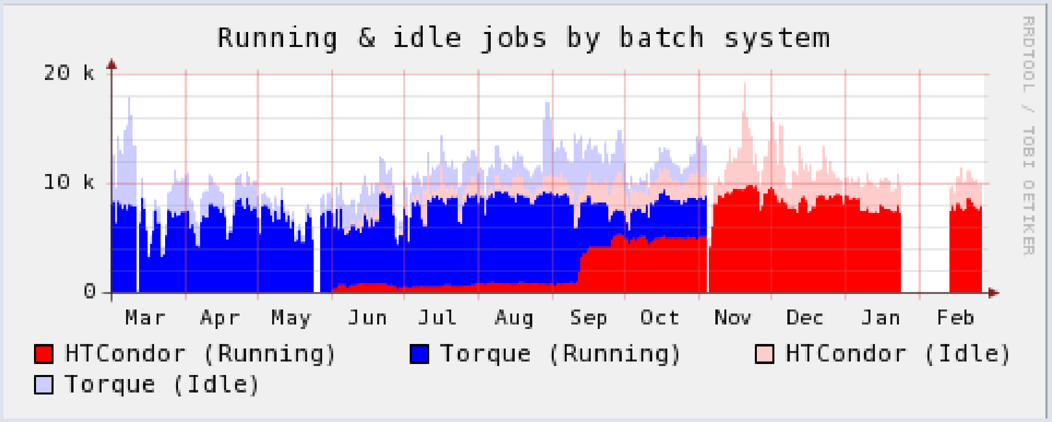
- We combined
  - Migration from Torque to HTCondor
  - Migration from SL5 to SL6

  therefore re-installed workers nodes from scratch
- Alternatives
  - Remove Torque & add HTCondor
  - Add HTCondor, then remove Torque later
    - Can have them running at the same time on the same worker node
    - We initially did some testing with sleep jobs in HTCondor while production jobs were running under Torque on the same worker nodes

Migration to HTCondor

- Migration timeline

2012 Aug	Started evaluating alternatives to Torque / Maui
	(LSF, Grid Engine, Torque 4, HTCondor, SLURM)
2013 Jun	Began testing HTCondor with ATLAS & CMS
	~1000 cores from old WNs beyond MoU commitments
2013 Aug	Choice of HTCondor approved by Change Control team
2013 Sep	HTCondor declared production service
	Moved 50% of pledged CPU resources to HTCondor
2013 Nov	Migrated remaining resources to HTCondor



GridPP
UK Computing for Particle Physics

Experience

GridPP
UK Computing for Particle Physics

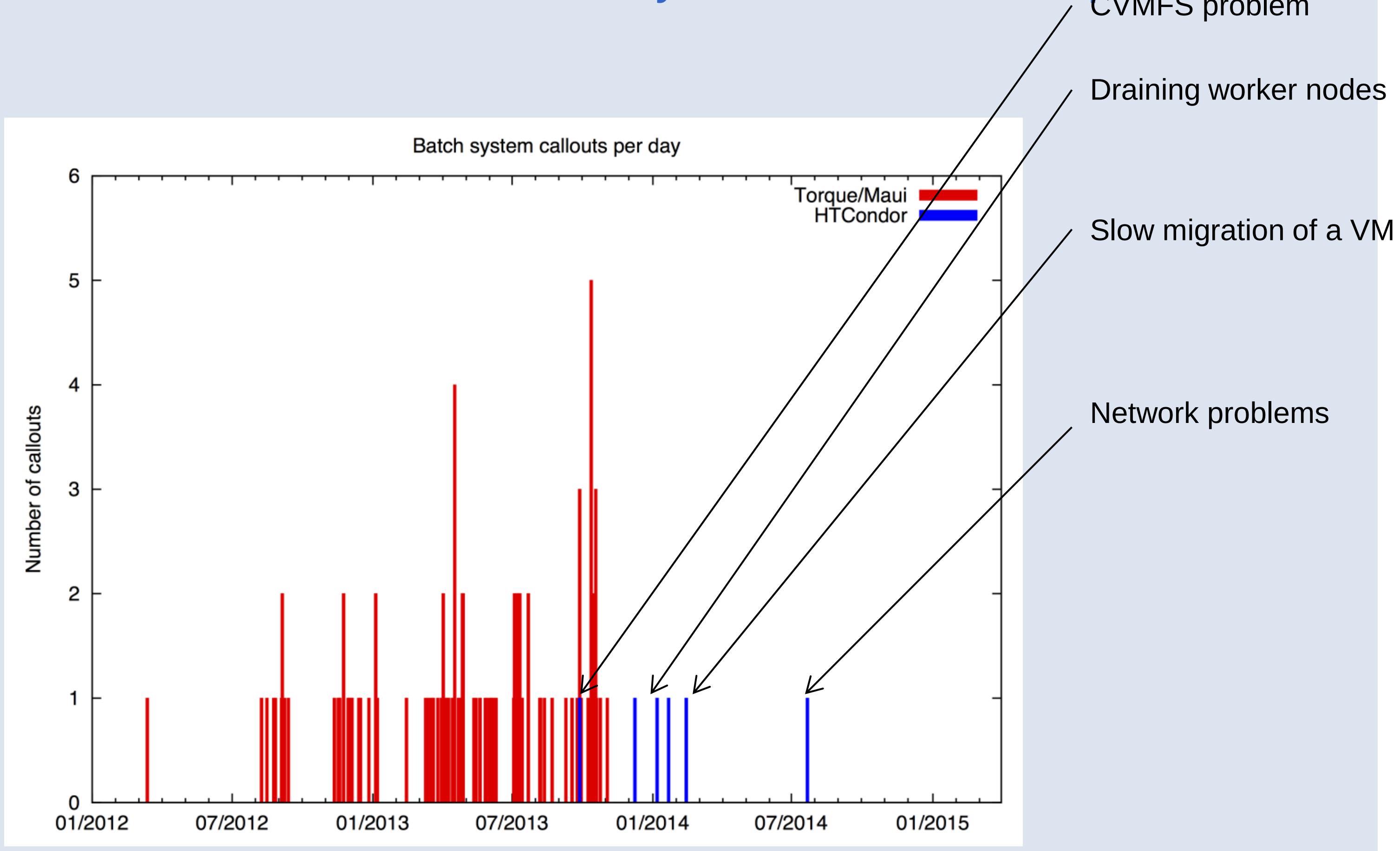
Experience

- Experience over past 2 years with HTCondor
  - Very stable operation
    - Generally just ignore the batch system & everything works fine
    - Staff don’t need to spend all their time fire-fighting problems
      - No more days spent studying the Torque source code trying to understand obscure error messages
  - No changes needed as the HTCondor pool increased in size from ~1000 to >10000 cores
  - Job start rate much higher than Torque / Maui even when throttled
    - Farm utilization much better
  - Upgrades easy
    - Central managers/CEs: HTCondor restarts itself after detecting binaries have been updated
    - Worker nodes: configured to drain themselves then restart after binaries are updated
  - Very good support

GridPP UK Computing for Particle Physics

Experience

- Significant reduction in numbers of callouts after migration to HTCondor
  - **None** of the callouts below were directly due to HTCondor problems

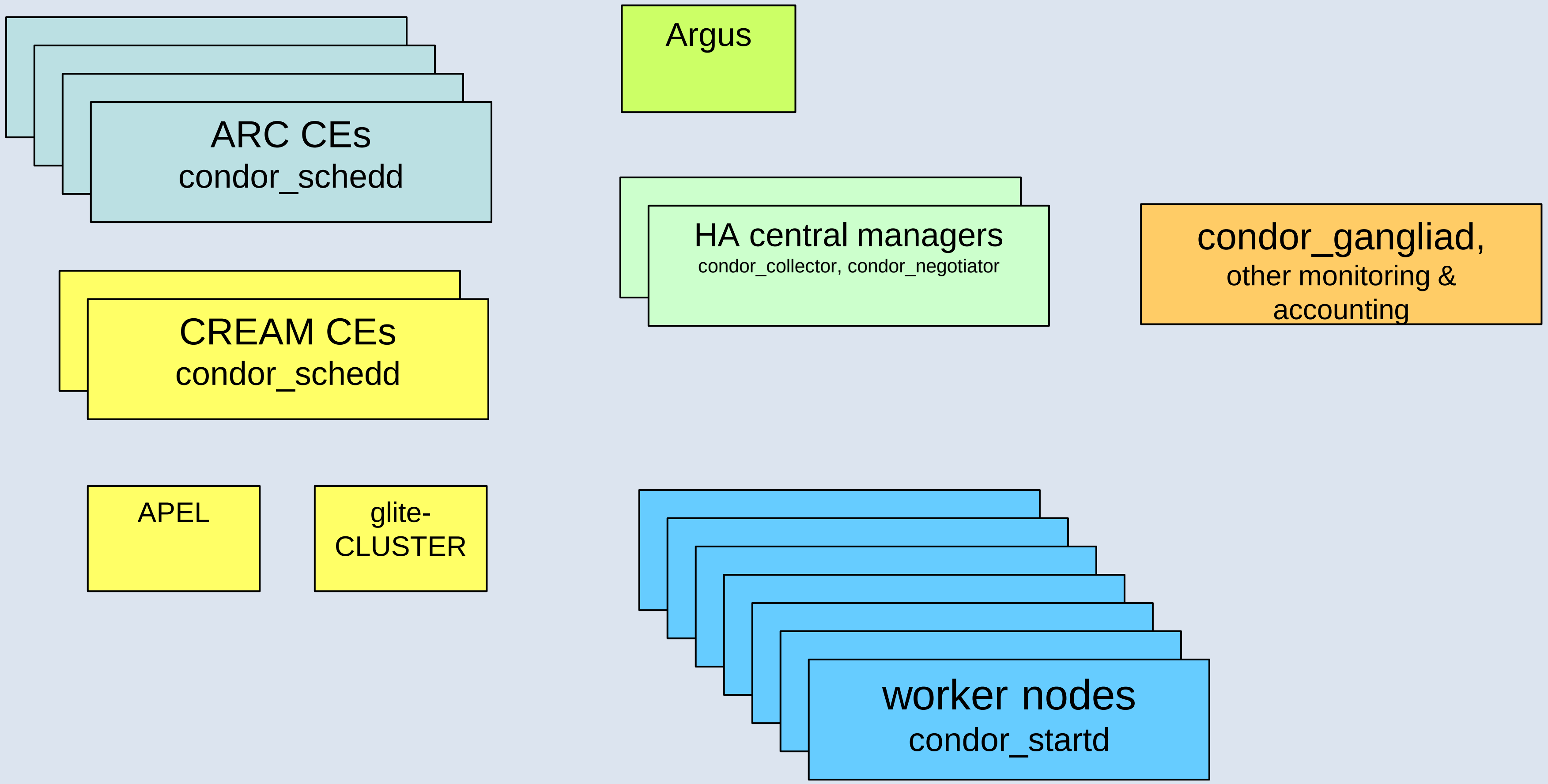


Functionality

GridPP
UK Computing for Particle Physics

Architecture

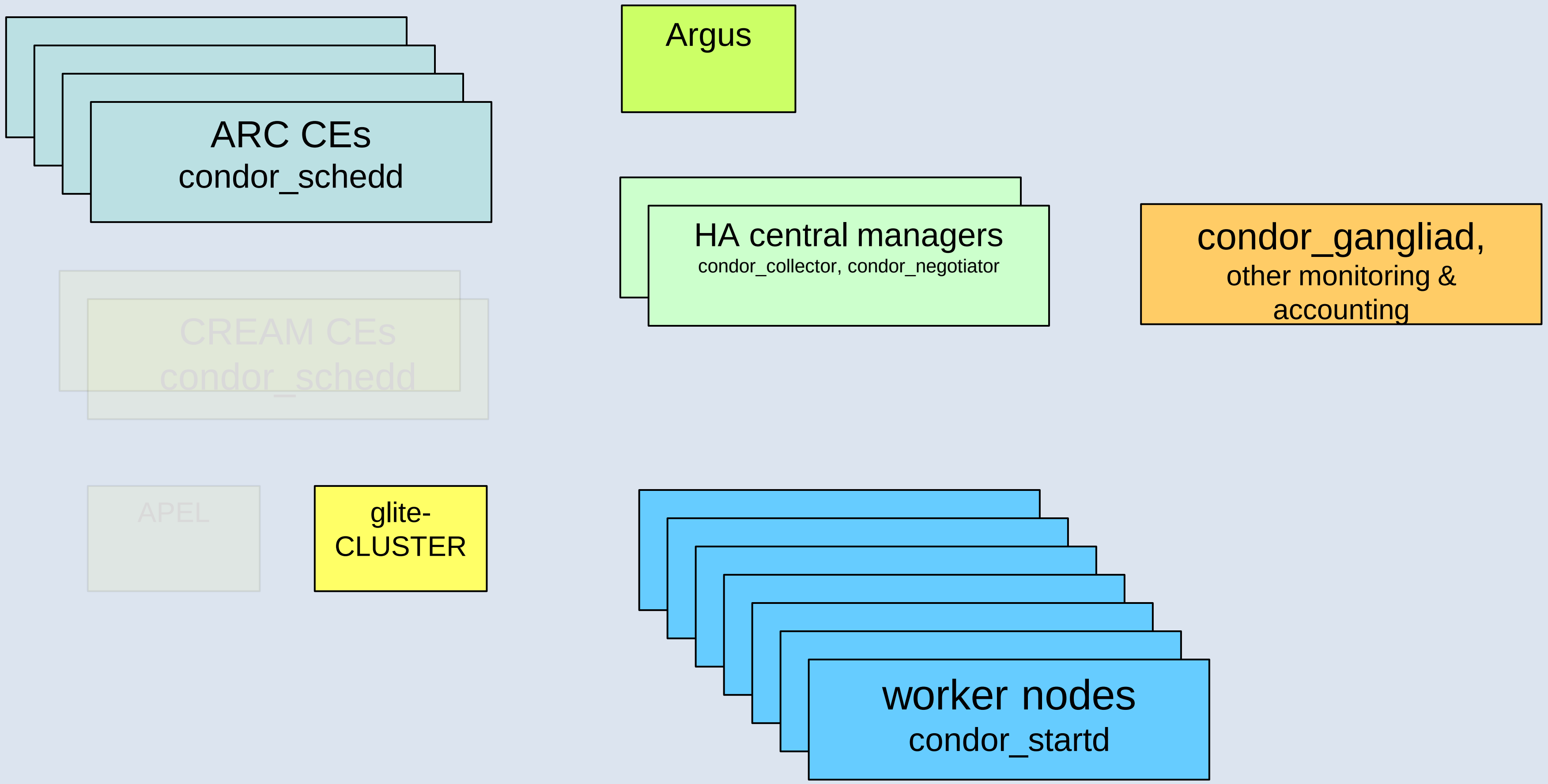
- Overview of the HTCondor pool & associated middleware…



GridPP UK Computing for Particle Physics

Architecture

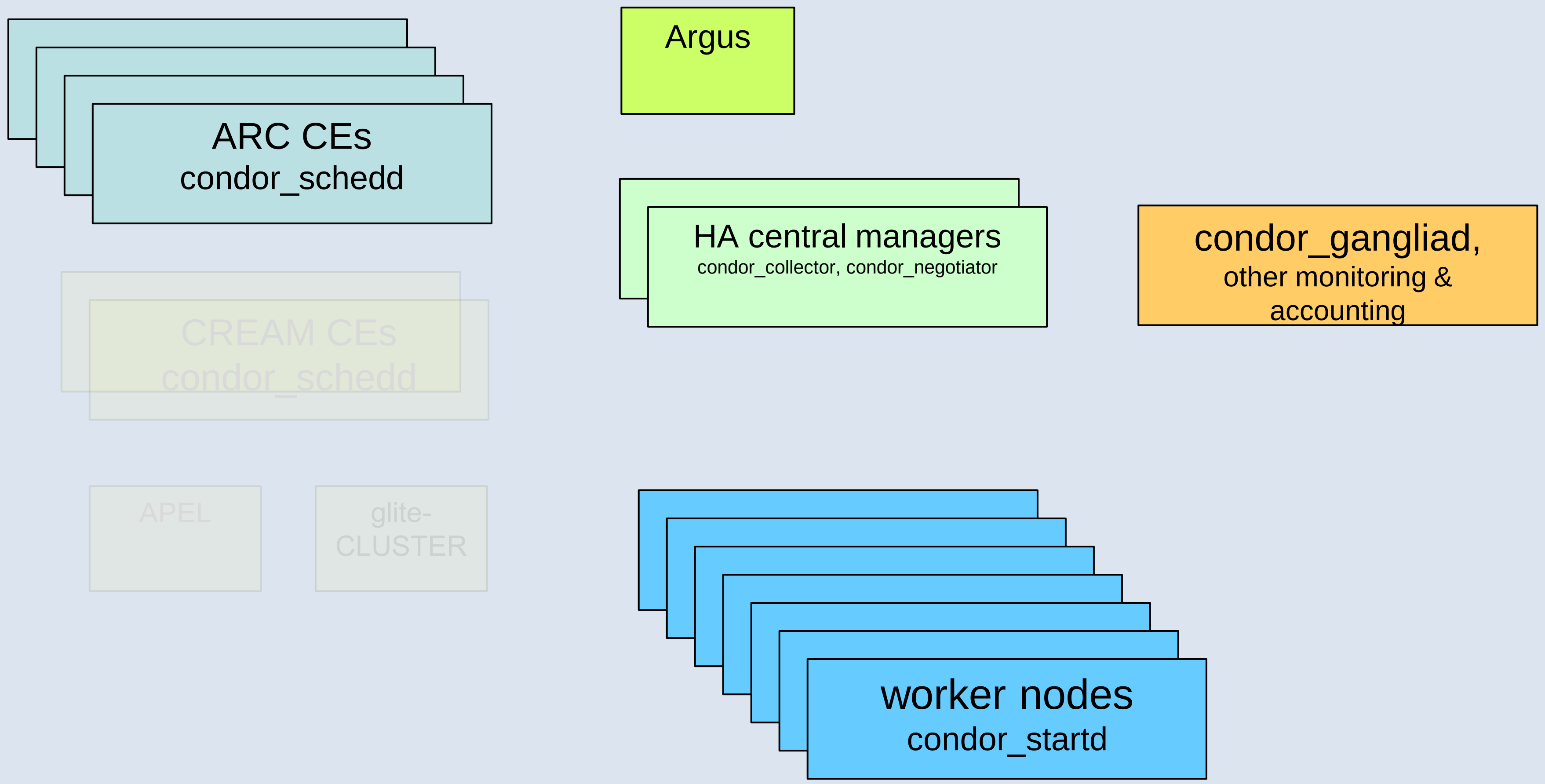
- …in May…



GridPP UK Computing for Particle Physics

Architecture

- …then possibly even this



GridPP
UK Computing for Particle Physics

Evolution

- Features in use at RAL

Beginning Hierarchical accounting groups, partitionable slots
HA central managers
PID namespaces
Jobs can't interfere with or even see other processes on the WN
Issue with ATLAS pilots killing themselves

July 2014 MOUNT_UNDER_SCRATCH (+ lcmaps-plugins-mount-under-scratch)
Jobs have their own /tmp, /var/tmp

July 2014 CPU cgroups
Jobs restricted to the number of cores they request, unless there are free cores available

Feb 2015 Memory cgroups
Using soft limits – jobs can exceed the memory they requested if there is memory available on the machine

GridPP
UK Computing for Particle Physics

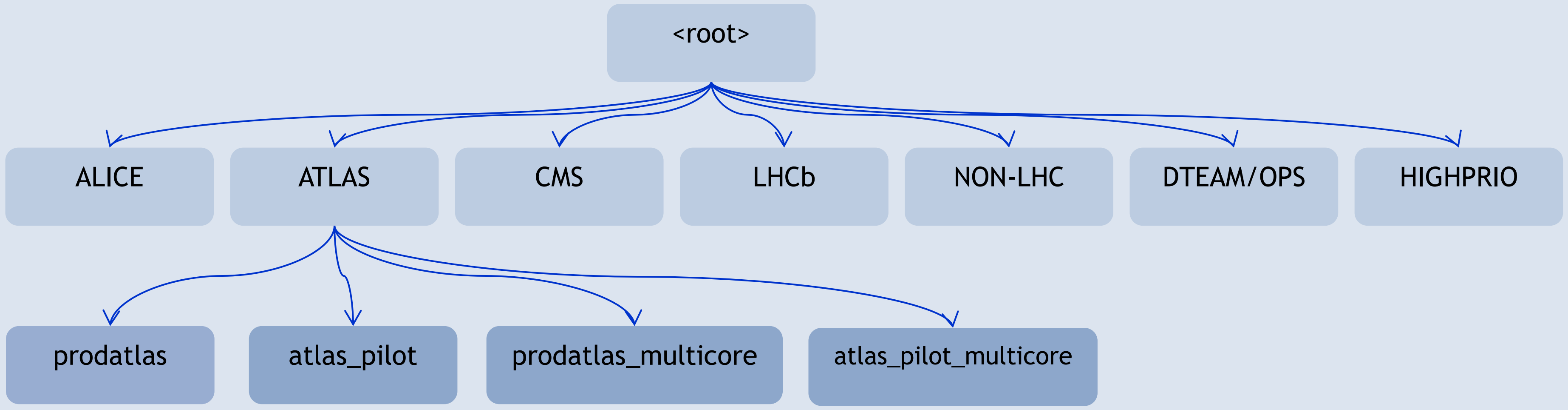
Partitionable slots

- Can configure worker nodes to have either static slots or partitionable slots (or both)
  - For environments where jobs have different requirements for numbers of cores and memory, partitionable slots make the most sense
- Partitionable slots
  - Can configure each worker node to have a single partitionable slot, containing all resources of the machine (CPUs, memory, disk, …)
  - These resources can then be divided up as necessary for jobs
    - Dynamic slots created from the partitionable slot
    - When dynamic slots exit, merge back into the partitionable slot
    - When any single resource is used up (e.g. memory), no more dynamic slots can be created

GridPP UK Computing for Particle Physics

Fairshares

- Accounting group setup at RAL (only ATLAS subgroups shown)



- Configuration
  - Negotiator configured to consider DTEAM/OPS and HIGHPRIO groups before all others
  - VO CE SUM test jobs forced to be in HIGHPRIO group

GridPP
UK Computing for Particle Physics

High availability of central managers

- Multiple machines can act as the central manager
  - Each machine will run an active collector
    - All submit machines & worker nodes will report to all collectors
  - Only one machine will have an active negotiator daemon
    - High availability daemon (condor_had) runs on each central manager & decides which machine will run the negotiator
  - Shared filesystem not required
    - condor_replication daemon runs on each central manager & replicates state information
- At RAL we've been always been using 2 central managers
  - Default configuration from the documentation works fine for us
  - Enables updates, reboots, etc to be done transparently
- We haven't considered high-availability of job queues

GridPP
UK Computing for Particle Physics

Jobs & memory

- Originally
  - By default ARC CE constructs PeriodicRemove expression so that if the ResidentSetSize of the job exceeds the requested memory, the job is killed
- After enabling memory cgroups
  - Thought it would be good to only have the kernel manage memory, so stopped the ARC CE from including a memory limit in PeriodicRemove
- However, found
  - LHCb analysis jobs using > 80 GB RSS (requested 4 GB)
  - ATLAS analysis jobs using ~ 10 GB RSS (requested 3 GB)
- Therefore, re-enabled the “traditional” memory limits, but configured to kill jobs if 3x requested memory is exceeded
  - May reduce this further
- Issue with memory cgroups
  - Under specific circumstances, if one job uses too much memory all cgroups are killed on the worker node (reported to HTCondor

GridPP
UK Computing for Particle Physics

Multi-core jobs

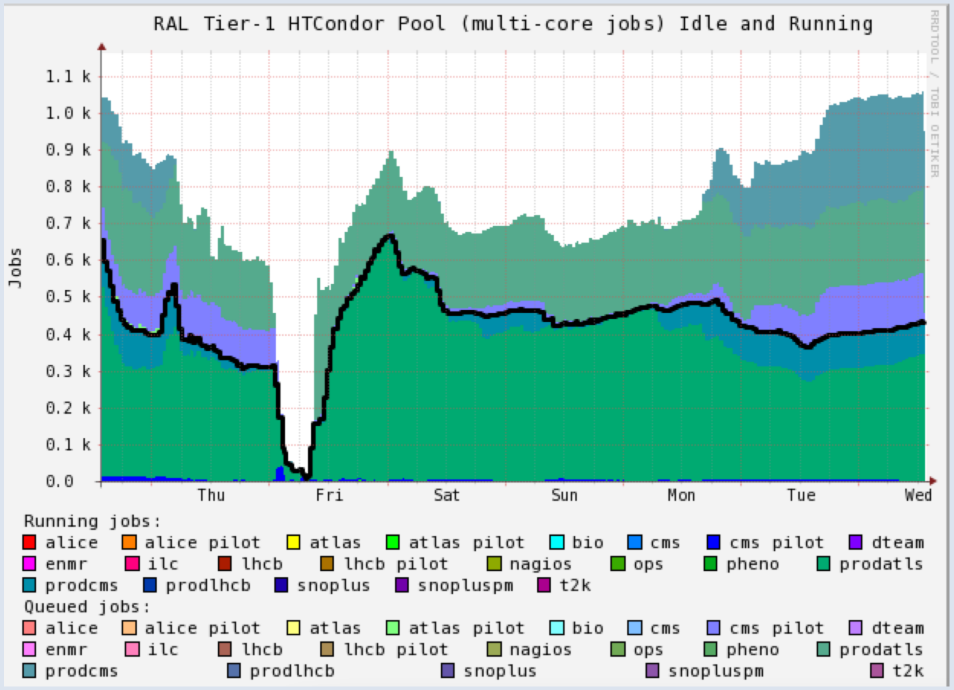
- Current situation
  - ATLAS have been running multi-core jobs since Nov 2013
  - CMS started submitting multi-core jobs in early May 2014
- Did a little tweaking early last year
  - Added accounting groups for multi-core jobs
  - Set GROUP_SORT_EXPR so that multi-core jobs are considered before single-core jobs
  - Defrag daemon enabled, configured so that
    - Drain 8 cores, not whole nodes
    - Pick WNs to drain based on how many cores they have that can be freed
  - Demand for multi-core jobs not known by defrag daemon
    - By default defrag daemon will constantly drain same number of WNs
    - Simple cron to adjust defrag daemon configuration based on demand
      - Uses condor_config_val to change DEFRAG_MAX_CONCURRENT_DRAINING

GridPP UK Computing for Particle Physics

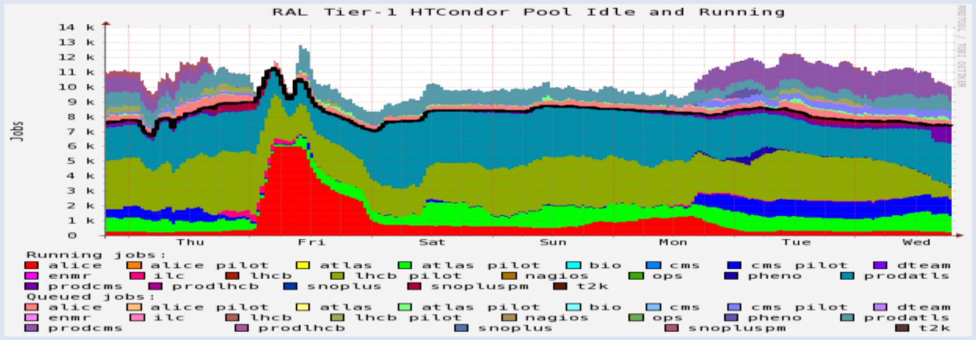
Multi-core jobs

- Running & idle multi-core jobs

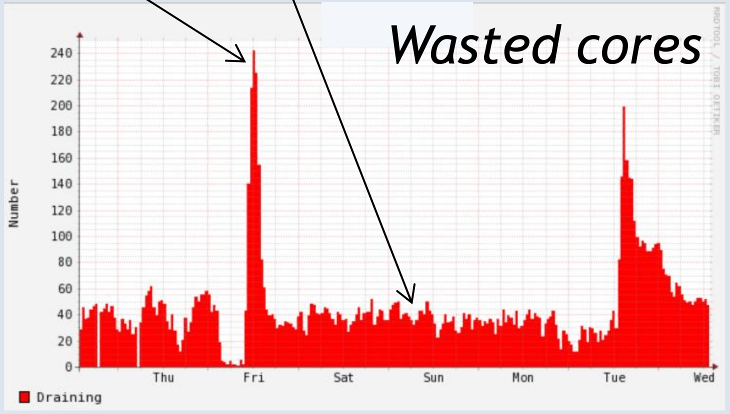
Multi-core jobs

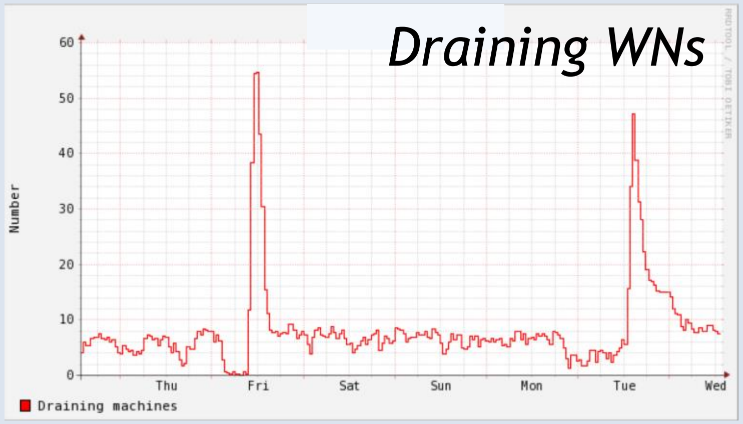


All jobs



2.4% 0.4%





Monitoring

GridPP UK Computing for Particle Physics

Monitoring

- Monitoring used for the RAL HTCondor pool
  - Ganglia
  - Nagios
  - Elasticsearch
  - (HTCondor startd cron)

GridPP
UK Computing for Particle Physics

Worker node health check

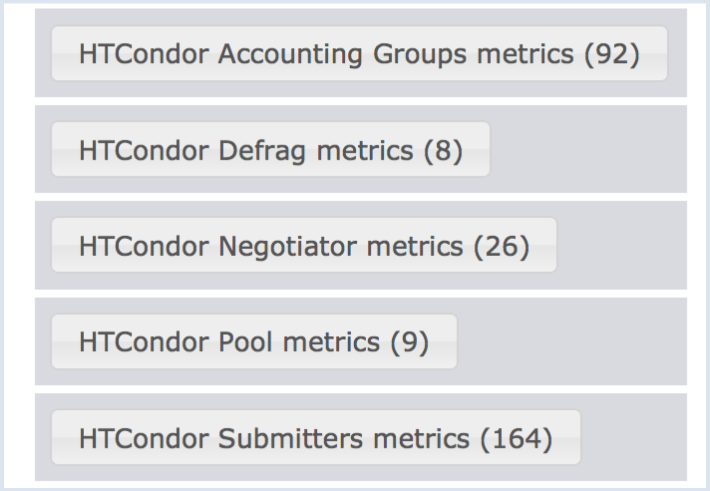
- Want to ignore worker nodes as much as possible
  - Any problems shouldn't affect new jobs
- Startd cron
  - Script checks for problems on worker nodes
    - Disk full or read-only
    - CVMFS
    - Swap usage
    - …
  - Prevents jobs from starting in the event of problems
    - If problem with ATLAS CVMFS, then only prevents ATLAS jobs from starting
    - CVMFS usually "self-heals" eventually
  - Information about problems made available in machine ClassAds
    - Can easily identify WNs with problems, e.g.

```
# condor_status –const 'NODE_STATUS =!= "All_OK”’ -af Machine NODE_STATUS
lcg0980.gridpp.rl.ac.uk Problem: CVMFS for alice.cern.ch
lcg0981.gridpp.rl.ac.uk Problem: CVMFS for cms.cern.ch Problem: CVMFS for
lhcb.cern.ch
```

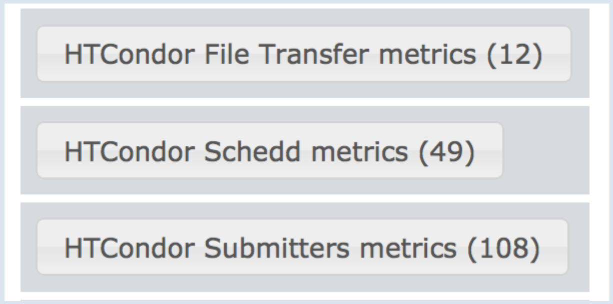
Ganglia

- condor_gangliad
  - Runs on a single host (any host)
  - Gathers daemon ClassAds from the collector
  - Publishes metrics to ganglia with host spoofing
    - Uses ganglia library rather than gmetric where possible
  - Examples

Central manager

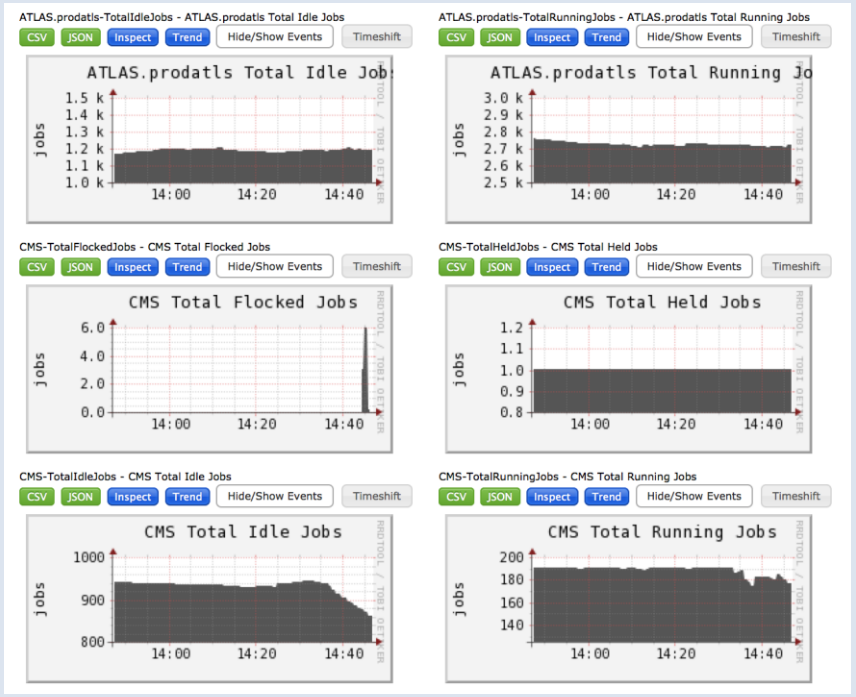

HTCondor Accounting Groups metrics (92)
HTCondor Defrag metrics (8)
HTCondor Negotiator metrics (26)
HTCondor Pool metrics (9)
HTCondor Submitters metrics (164)

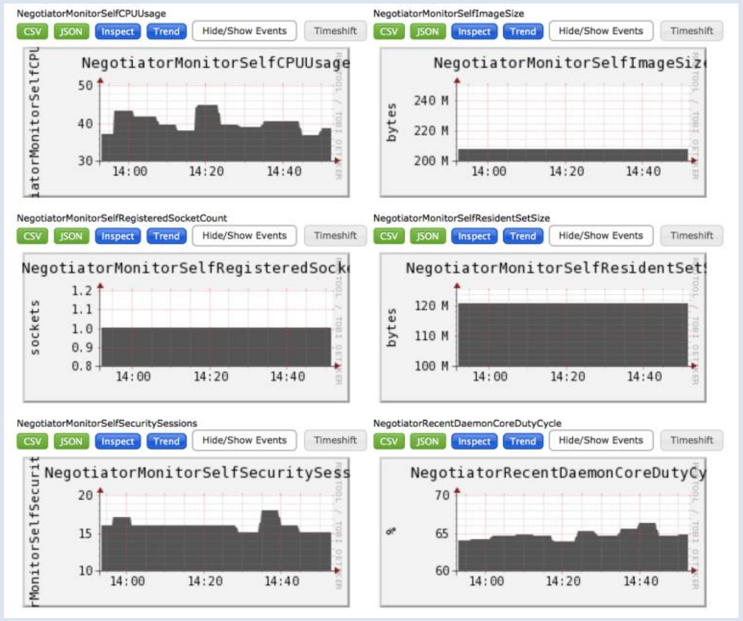
CE

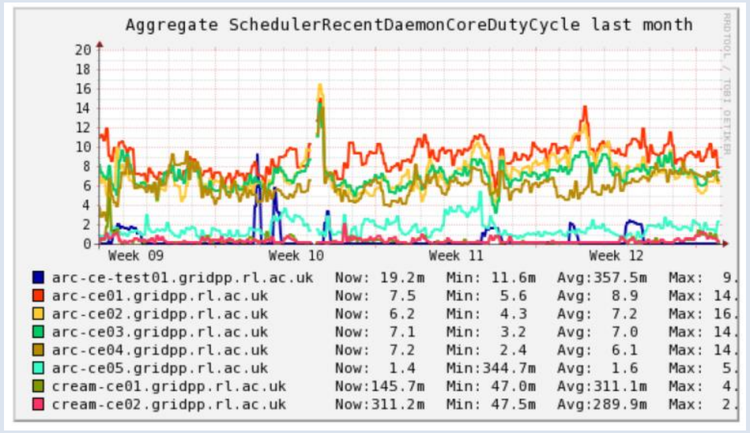

HTCondor File Transfer metrics (12)
HTCondor Schedd metrics (49)
HTCondor Submitters metrics (108)

Ganglia

- condor_gangliad


ATLAS.prodatls-TotalIdleJobs - ATLAS.prodatls Total Idle Jobs
ATLAS.prodatls Total Idle Job
ATLAS.prodatls-TotalRunningJobs - ATLAS.prodatls Total Running Jobs
ATLAS.prodatls Total Running Jo
CMS-TotalFlockedJobs - CMS Total Flocked Jobs
CMS Total Flocked Jobs
CMS-TotalHeldJobs - CMS Total Held Jobs
CMS Total Held Jobs
CMS-TotalIdleJobs - CMS Total Idle Jobs
CMS Total Idle Jobs
CMS-TotalRunningJobs - CMS Total Running Jobs
CMS Total Running Jobs
CSV
JSON
Inspect
Trend
Hide/Show Events
Timeshift
jobs
14:00
14:20
14:40


NegotiatorMonitorSelfCPUUsage
NegotiatorMonitorSelfImageSize
NegotiatorMonitorSelfRegisteredSocketCount
NegotiatorMonitorSelfResidentSetSize
NegotiatorMonitorSelfSecuritySessions
NegotiatorRecentDaemonCoreDutyCycle
CSV
JSON
Inspect
Trend
Hide/Show Events
Timeshift
bytes
sockets
%
14:00
14:20
14:40

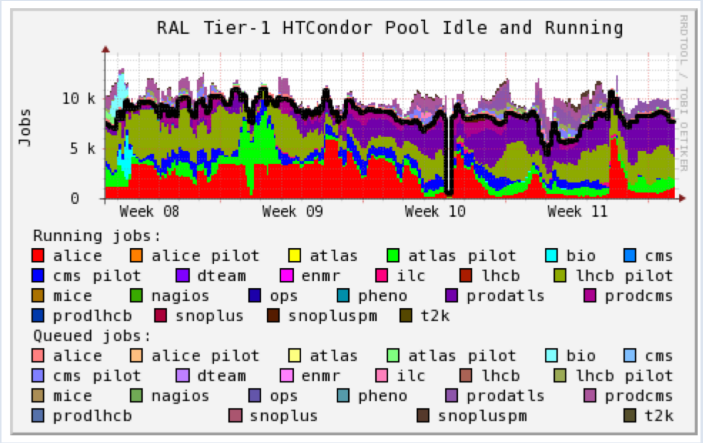


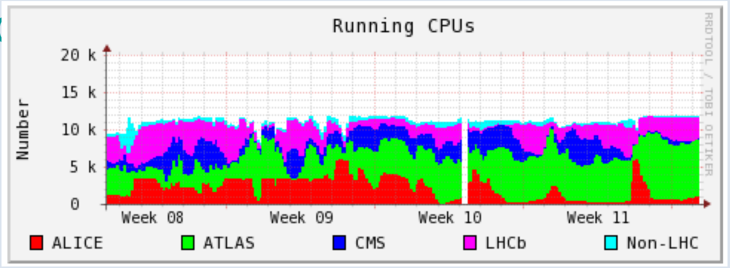
```
■ arc-ce-test01.gridpp.rl.ac.uk  Now: 19.2m  Min: 11.6m  Avg:357.5m  Max:  9.
■ arc-ce01.gridpp.rl.ac.uk       Now:  7.5   Min:  5.6   Avg:  8.9   Max: 14.
■ arc-ce02.gridpp.rl.ac.uk       Now:  6.2   Min:  4.3   Avg:  7.2   Max: 16.
■ arc-ce03.gridpp.rl.ac.uk       Now:  7.1   Min:  3.2   Avg:  7.0   Max: 14.
■ arc-ce04.gridpp.rl.ac.uk       Now:  7.2   Min:  2.4   Avg:  6.1   Max: 14.
■ arc-ce05.gridpp.rl.ac.uk       Now:  1.4   Min:344.7m  Avg:  1.6   Max:  5.
■ cream-ce01.gridpp.rl.ac.uk     Now:145.7m  Min: 47.0m  Avg:311.1m  Max:  4.
■ cream-ce02.gridpp.rl.ac.uk     Now:311.2m  Min: 47.5m  Avg:289.9m  Max:  2.
```

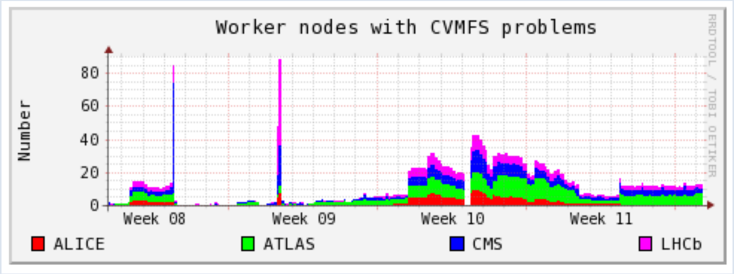
GridPP UK Computing for Particle Physics

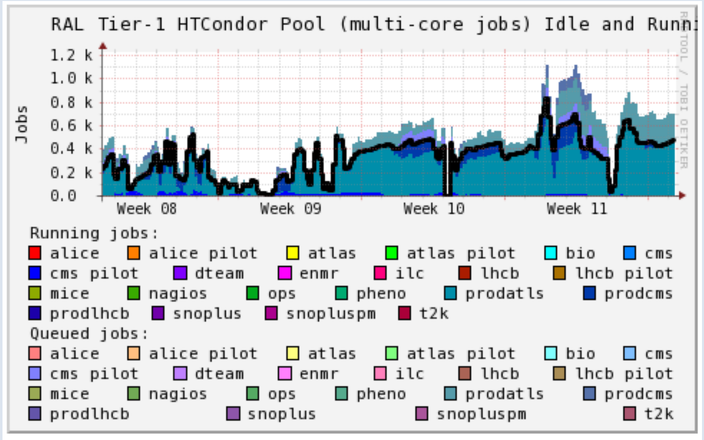
Ganglia

- Custom ganglia plots
  - gmetric scripts running on a central manager + Perl scripts on ganglia server
    - If doing this again we would use metrics from condor_gangliad as much as our









GridPP
UK Computing for Particle Physics

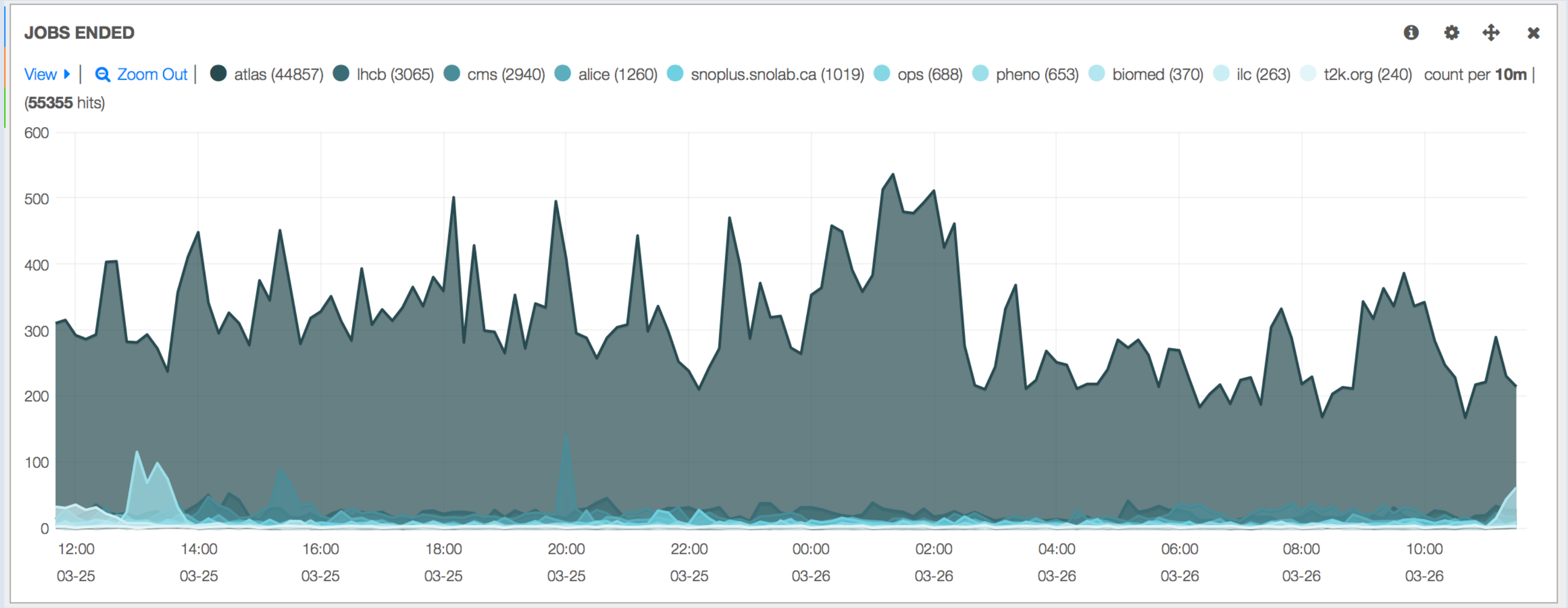
Nagios checks

- Central managers
  - Process check for condor_master
  - Check for number of collectors visible in the pool
  - Check for 1 negotiator in the pool
  - Worker node check
    - Need a minimum number of worker nodes advertised & willing to run jobs
- CEs
  - Process check for condor_master
  - Check for schedd being advertised
- Worker nodes
  - Process check for condor_master (won’t trigger pager alarm)

GridPP
UK Computing for Particle Physics

Monitoring using ELK

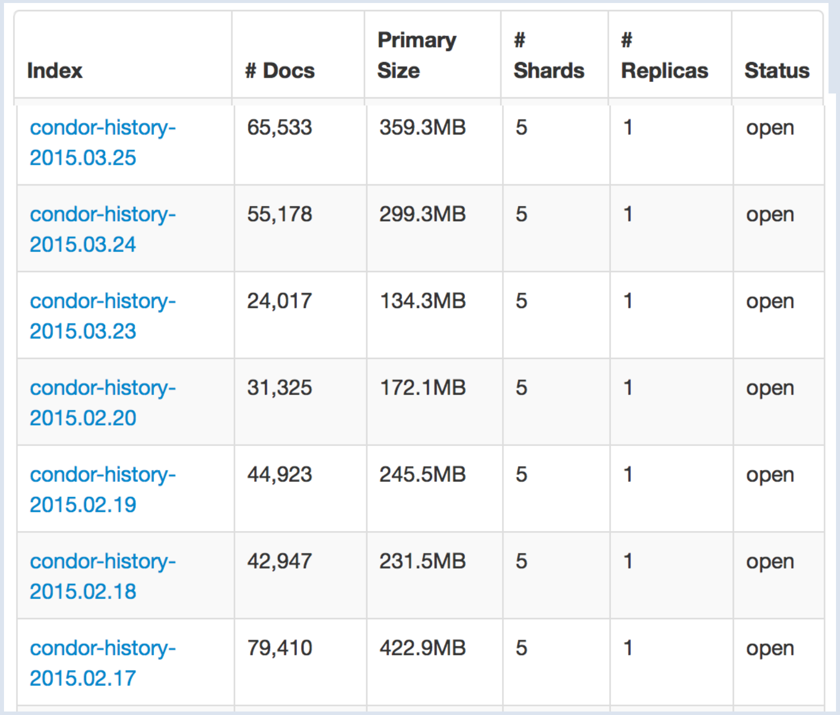
- Elasticsearch ELK stack at RAL, mostly used for CASTOR
- Adding HTCondor
  - First step: information about completed jobs
  - Wrote config file for Logstash to enable history files to be parsed
  - Added logstash to all machines running schedds



GridPP
UK Computing for Particle Physics

Monitoring using ELK

- 'Minimal' resources used
  - Generally < 80,000 documents, < 500 MB per day

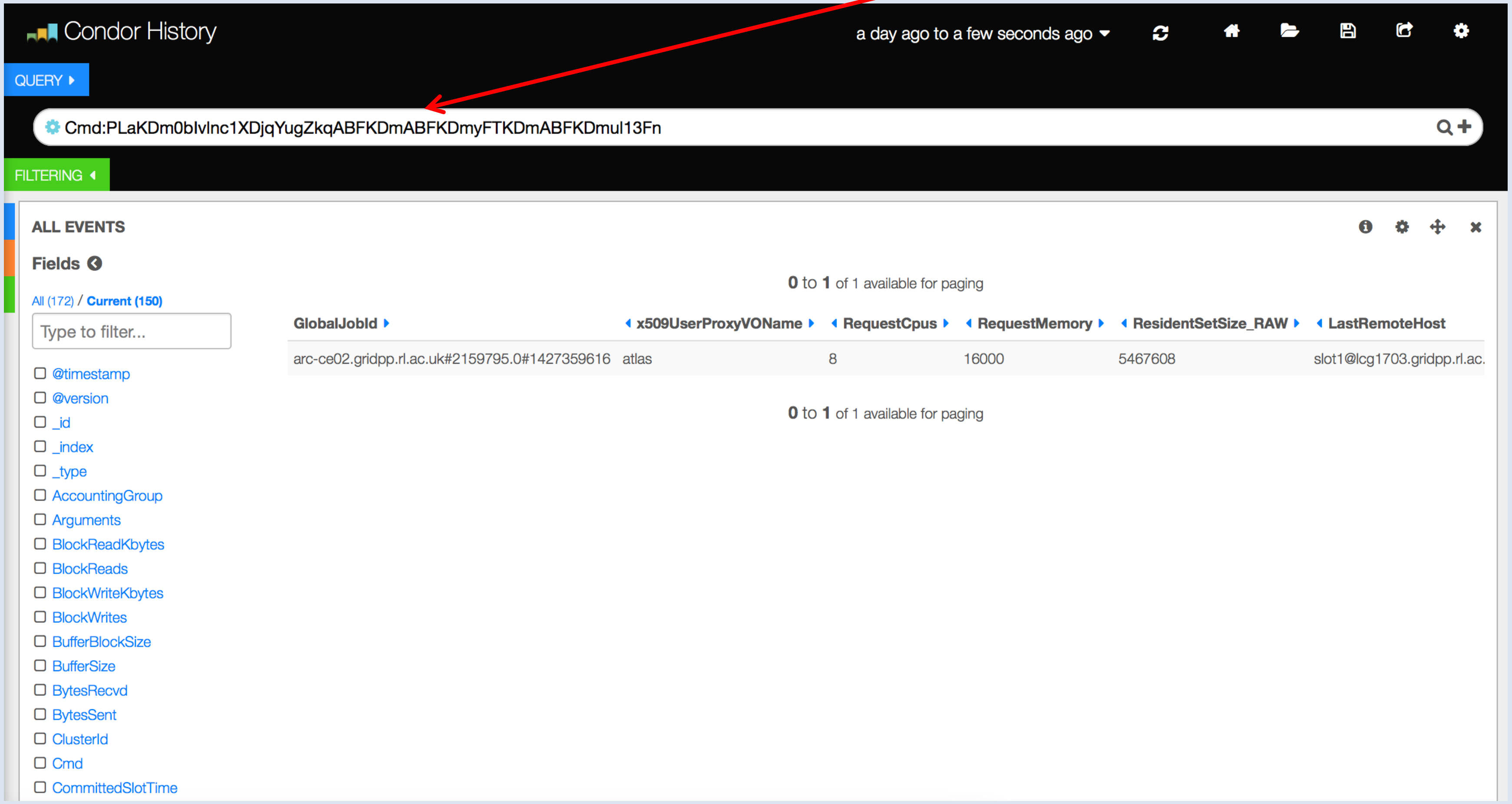


Index	# Docs	Primary Size	# Shards	# Replicas	Status
condor-history-2015.03.25	65,533	359.3MB	5	1	open
condor-history-2015.03.24	55,178	299.3MB	5	1	open
condor-history-2015.03.23	24,017	134.3MB	5	1	open
condor-history-2015.02.20	31,325	172.1MB	5	1	open
condor-history-2015.02.19	44,923	245.5MB	5	1	open
condor-history-2015.02.18	42,947	231.5MB	5	1	open
condor-history-2015.02.17	79,410	422.9MB	5	1	open

GridPP UK Computing for Particle Physics

Monitoring using ELK

- Search for information about completed jobs (faster than using condor_history)



```
Cmd:PLaKDm0bIvlnc1XDjqYugZkqABFKDmABFKDmyFTKDmABFKDmul13Fn
```

0 to 1 of 1 available for paging

GlobalJobId	x509UserProxyVOName	RequestCpus	RequestMemory	ResidentSetSize_RAW	LastRemoteHost
arc-ce02.gridpp.rl.ac.uk#2159795.0#1427359616	atlas	8	16000	5467608	slot1@lcg1703.gridpp.rl.ac.

0 to 1 of 1 available for paging

- @timestamp
- @version
- _id
- _index
- _type
- AccountingGroup
- Arguments
- BlockReadKbytes
- BlockReads
- BlockWriteKbytes
- BlockWrites
- BufferBlockSize
- BufferSize
- BytesRecvd
- BytesSent
- ClusterId
- Cmd
- CommittedSlotTime

GridPP
UK Computing for Particle Physics

Ongoing work & future plans

GridPP UK Computing for Particle Physics

Ongoing work & future plans

- Integration with private cloud
  - OpenNebula cloud setup at RAL, currently with ~1000 cores
  - Want to ensure any idle capacity is used, so why not run virtualized worker nodes?
  - Want opportunistic usage which doesn't interfere with cloud users
    - Batch system expands into cloud when batch system busy & cloud idle
    - Batch system withdraws from cloud when cloud becomes busy
  - Successfully tested, working on moving this into production
  - See posters at CHEP
- Upgrade worker nodes to SL7
  - Setup SL6 worker node environment in a chroot, run SL6 jobs in the chroot using NAMED_CHROOT functionality in HTCondor
    - Will simplify eventual migration to SL7 – can run both SL6 and SL7 jobs
  - Successfully tested CMS jobs

GridPP
UK Computing for Particle Physics

Ongoing work & future plans

- Simplification of worker nodes
  - Testing use of CVMFS grid.cern.ch for grid middleware
    - 540 packages installed vs 1300 for a normal worker node
  - HTCondor can run jobs:
    - In chroots
    - In filesystem namespaces
    - In PID namespaces
    - In memory cgroups
    - In CPU cgroups
  - Do we really need pool accounts on worker nodes?
    - With the above, one job can’t see any processes or files associated with any other jobs on the same worker node, even if the same user
    - Worker nodes and CEs could be much simpler without them!

GridPP
UK Computing for Particle Physics

Questions?