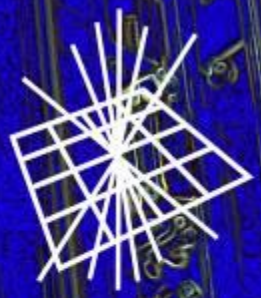


GridPP

UK Computing for Particle Physics

Tier-1 Batch System Report

Andrew Lahiff, Alastair Dewhurst, John Kelly, Ian Collier

5 June 2013, HEP SYSMAN

Current batch system

- Current production batch system
  - 656 worker nodes, 9424 job slots
  - Torque 2.5.12, Maui 3.3.1
- Issues
  - pbs_server, maui sometimes unresponsive
  - pbs_server needs to be restarted sometimes due to excessive memory usage
  - Job start rate sometimes not high enough to keep the farm full
  - Regular job submission failures - *Connection timed out-qsub: cannot connect to server*
  - Unable to schedule jobs to the whole-node queue
  - Sometimes schedules SL5 jobs on SL6 worker nodes & vice-versa
  - DNS issues, network issues & problematic worker nodes cause it to "require attention"

Alternatives

- Tested the following technologies
  - Torque 4 + Maui
  - LSF
  - Grid Engine
  - SLURM
  - HTCondor
- Identified SLURM & HTCondor as suitable for further testing at larger scales
  - ✗ LSF, Univa Grid Engine, Oracle Grid Engine
    - Avoid expensive solutions unless all open source products unsuitable
  - ✗ Torque 4 + Maui
    - Still need to use Maui
  - ✗ Open source Grid Engines
    - Competing products, not clear which has best long-term future; neither seems to have communities as active as SLURM & HTCondor

HTCondor vs SLURM

- Larger scale tests with 110 worker nodes
- HTCondor
  - No problems running 18000 jobs
    - Need to ensure machines with job queues (schedds) have enough memory
  - No problems with > 200000 pending jobs
- SLURM
  - Stability problems experienced when running > ~6000 jobs
    - Everything fine when no jobs are completing or starting (!)
  - Queries (sinfo, squeue, …) and job submission fail:

    Socket timed out on send/recv operation
  - Using FIFO scheduling helps
    - Cannot use this in production
  - Some activities (restarting SLURM controller, killing large numbers of jobs, …) trigger unresponsiveness
    - Takes many hours to return to a stable situation

HTCondor vs SLURM

- Currently, HTCondor seems like a better choice for us than SLURM
  - Move onto next stage of testing with HTCondor
    - Testing with selected VOs *(more info later)*
  - Will continue try to resolve problems with SLURM
- SLURM
  - Advertised as handling up to 65536 nodes, hundreds of thousands of processors
  - However, in the Grid community
    - Need fairshares, “advanced” job scheduling, accounting, …
    - Tier-2s using SLURM in production have less job slots than we tried
      - e.g. Estonia have 4580 job slots
    - Other people who have tested SLURM did so at smaller scales than us
  - We have tried identical configuration to a SLURM Tier-2, without success at large scales

HTCondor

- Features we're using
  - High-availability of central manager
  - Hierarchical fairshares
  - Partitionable slots: available cores & memory divided up as necessary for each job
  - Concurrency limits, e.g. restrictions on max running jobs for specific users
- Features not tried yet
  - IPv6
  - Saving power: hibernation of idle worker nodes
  - CPU affinity, per-job PID namespaces, chroots, per-job /tmp directories, cgroups, …
- Queues
  - HTCondor has no traditional queues – jobs specify their requirements
    - e.g. SL5 or SL6, number of cores, amount of memory, …

Compatibility with Middleware

- HTCondor & EMI-3 CREAM CE
  - BUpdaterCondor & condor-*_.sh scripts exist
    - Very easy to get a CREAM CE working successfully with HTCondor
  - Script for publishing dynamic information doesn't exist
- What about ARC CEs?
  - Could migrate from CREAM to ARC CEs
  - Benefits of ARC CEs
    - Support HTCondor (including all publishing)
    - Simpler than CREAM CEs (no YAIM, no Tomcat, no MySQL, …)
  - Accounting
    - ARC CE accounting publisher (JURA) can send accounting records directly to APEL using SSM
      - Not tried yet

Current status

- "Almost" production quality service setup
  - HTCondor 7.8.8 with 2 central managers
  - 2 ARC CEs (EMI-3 3.0.1-1), using LCAS/LCMAPS + Argus
  - 112 8-core EMI-2 SL6 worker nodes
- Testing
  - Evaluation period up to ~3 months using resources beyond WLCG pledges
  - Aim to gain experience running "real" work
    - Stability, reliability, functionality, dealing with problems, …
  - Initial testing with ATLAS and CMS
    - ATLAS: production & analysis SL6 queues
    - CMS: Monte Carlo production (integration testbed)
  - Next steps
    - Ramp up CMS usage
    - Try other VOs, including non-LHC VO submission through WMS

Backup slides

Test

- Test to reproduce problems seen with production batch system
  - High job submission rate (1000 jobs as quickly as possible)
  - High job query rate (1000 qstat's or equivalent)

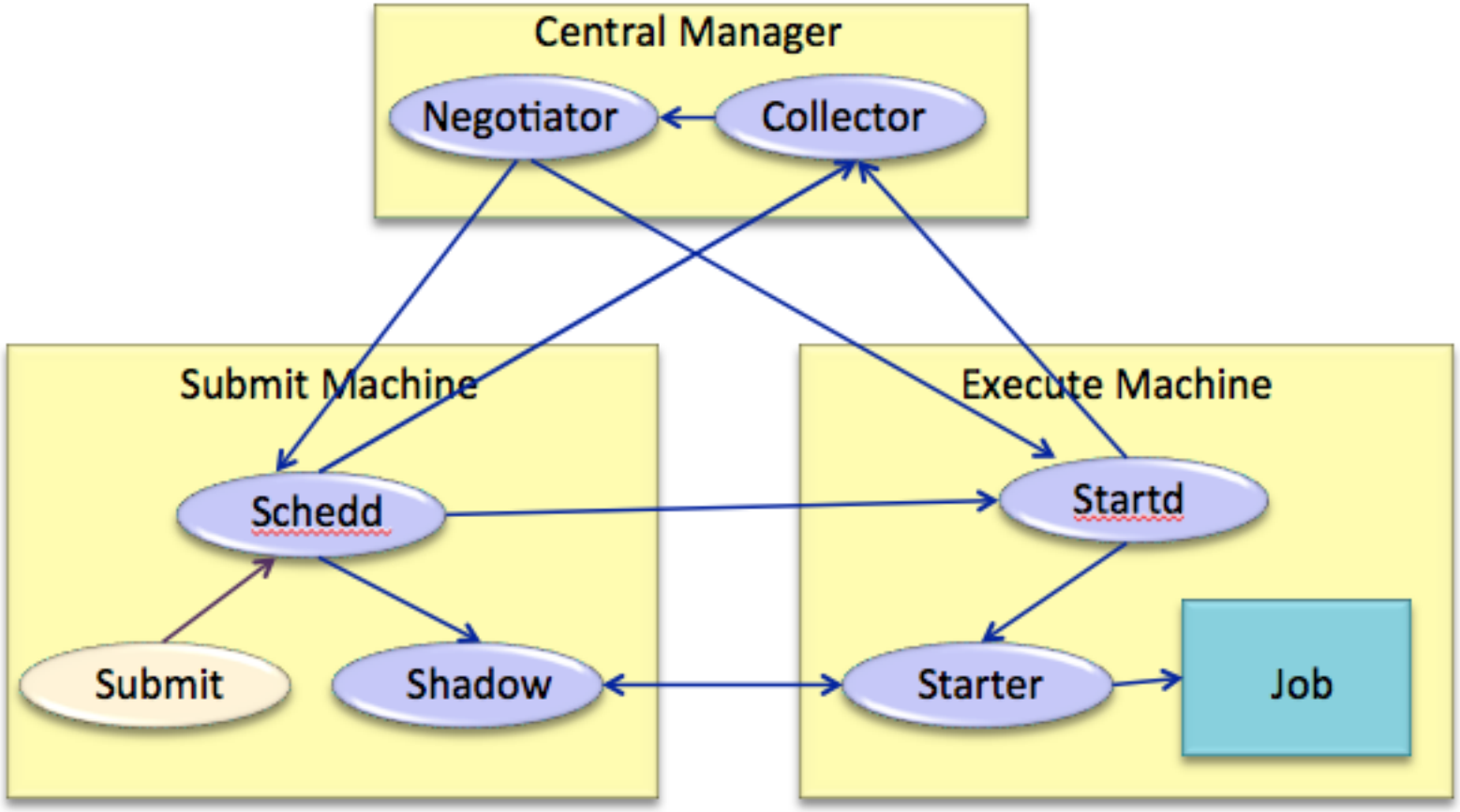
	Torque 2.5.12	Torque 4.1.4	Torque 4.2.0	Condor	LSF	Grid Engine	SLURM
High job submission rate	1	5,3	2,3				
High job query rate	4	6	3				

- Torque 2.5.12
  - 90% job submissions failed
  - 70-80% job queries failed
- Torque 4.x
  - < 10% failure rate
- HTCondor, LSF, Grid Engine, SLURM
  - 100% successful

HTCondor: basic principles

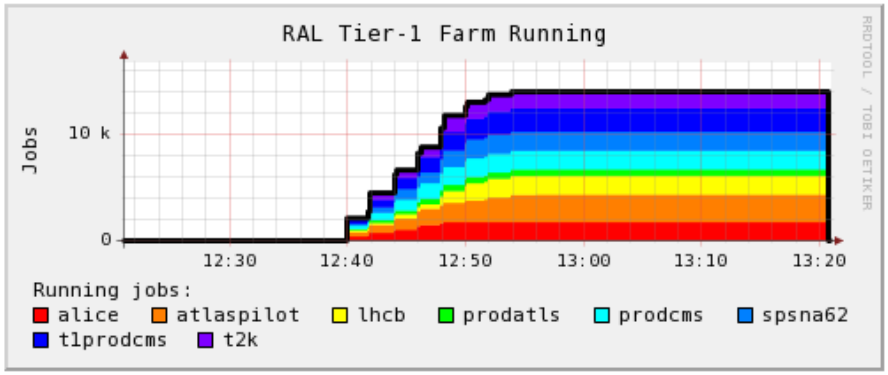
- Two parts
  - Jobs
  - Machines/Resources
- Jobs state their requirements, preferences & attributes
  - E.g. I require a Linux/x86 platform
  - E.g. I am owned by atlas001
  - E.g. I prefer a machine owned by CMS
- Machines state their requirements, preferences & attributes
  - E.g. I require jobs that require less than 4 GB of RAM
  - E.g. I am a Linux node; I have 16 GB of memory
- HTCondor matches jobs to resources
  - Also has priorities, fairshares, etc taken into account

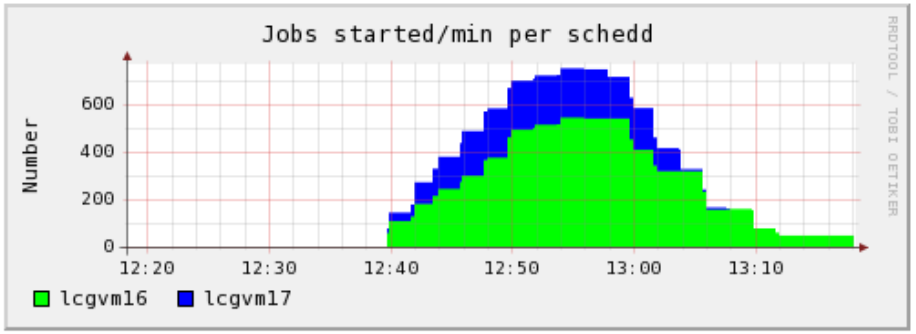
HTCondor overview

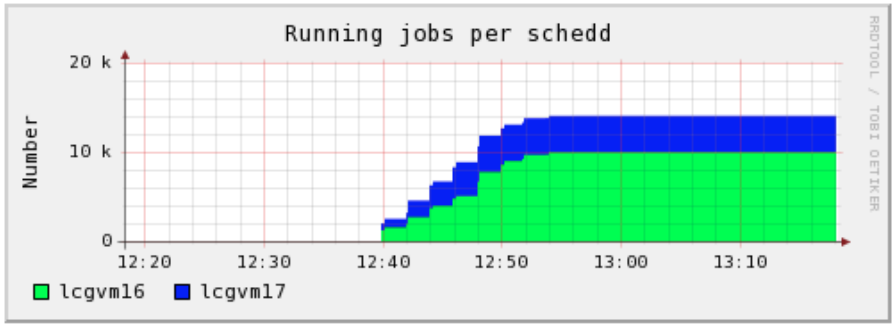


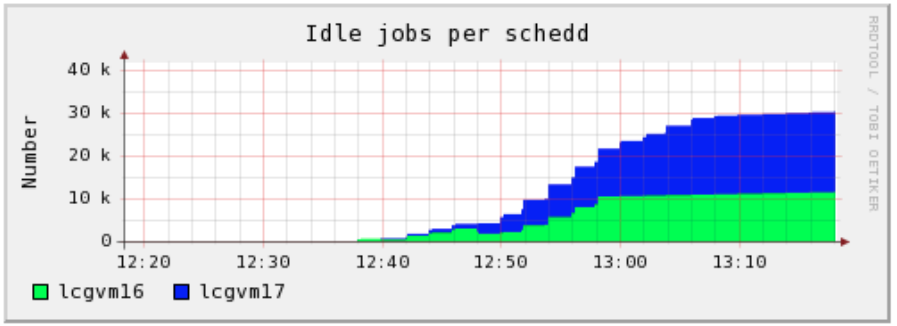
Basic ganglia monitoring

- Monitoring showing
  - Running & idle jobs
  - Jobs starting, exiting & submitted per schedd (CE)

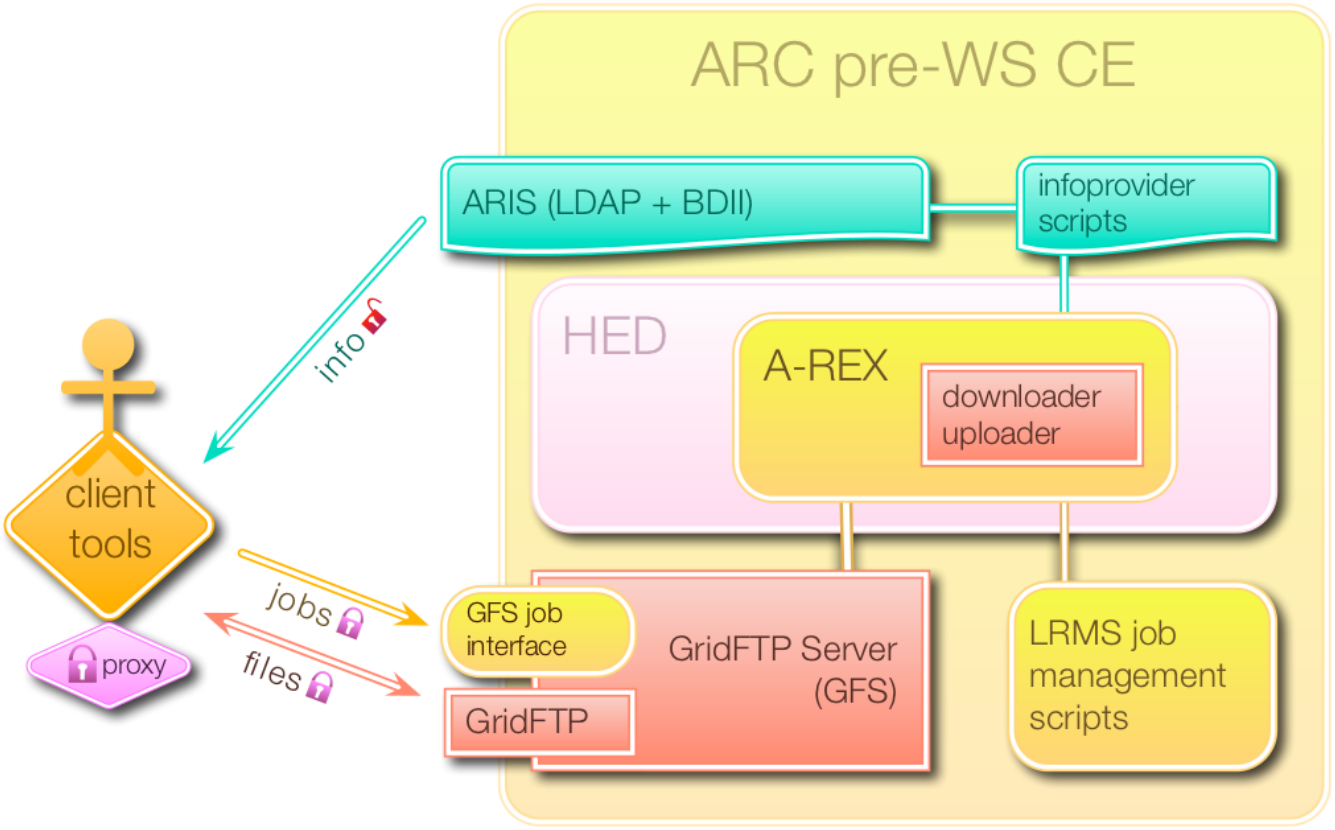








ARC CE


ARC pre-WS CE
ARIS (LDAP + BDII)
infoprovider scripts
HED
A-REX
downloader uploader
info
client tools
proxy
jobs
files
GFS job interface
GridFTP
GridFTP Server (GFS)
LRMS job management scripts