GridPP

UK Computing for Particle Physics

Experience with dynamically provisioned worker nodes at the RAL Tier-1

Andrew Lahiff, Ian Collier, Orlin Alexandrov

01 Nov 2013, HEPiX Fall 2013 Workshop

Outline

- STFC Scientific Computing Dept cloud
- What we did
  - Adding virtual worker nodes to a batch system
  - Images
  - Provisioning worker nodes
- Testing
  - Virtualisation overhead
  - Dynamically provisioned worker nodes
- Monitoring
- Next steps

SCD cloud

- Prototype cloud
  - Gain practical experience
  - Test potential use cases of a private cloud
- Using StratusLab
  - Based on OpenNebula
  - Easy to setup – Quattor configuration provided
- Using iSCSi & LVM based persistent disk storage
  - Caches images
  - Instantiation very fast, ~20-30 secs or less
- Hardware
  - Cloud front end on a VM (hosted on Hyper-V)
  - Persistent disk store on a 18TB retired disk server
  - Hypervisors: ~ 100 retired worker nodes (8 cores, 16 GB RAM)

SCD cloud

- Usage
  - Available to all SCD staff on a self-service basis
  - Very useful for testing & development
  - Around 30 active users
- However
  - The cloud resources are largely idle
  - Our batch system has the opposite problem
    - Many idle jobs, not enough resources
- Can we easily make use of the idle cloud resources in our batch system?

Adding to the batch system

- Main aspects of this:
  1. Provisioning virtual machines as they are required
  2. Adding the new worker nodes to the batch system
  3. Deciding when to remove virtual worker nodes
- Recently started migrating to HTCondor
  - Adding & removing worker nodes is trivial
    - New worker nodes advertise to the collector
    - Need the appropriate privileges to join the HTCondor pool
  - No complicated procedures that other batch systems may require, e.g. Torque

Images

GridPP UK Computing for Particle Physics

- EMI-2 SL6 worker node, same as our physical production worker nodes
  - Only images created by RAL Tier-1 sys admins are used, and are therefore trusted
  - Experiments/users can't provide images
    - In the future this could change
- Privileges compared to physical worker nodes
  - Generally the same
    - E.g. CASTOR permissions
  - Except
    - HTCondor pool password not built in to images (inserted during instantiation)

GridPP
UK Computing for Particle Physics

Health-check

- Since worker nodes are being created dynamically, need to ensure we're not just creating black holes
- Using same health-check script as on our physical HTCondor worker nodes
- Runs as a startd cron
- Current checks:
  - CVMFS
  - Read-only file system
  - Space left on job scratch area partition
  - Swap usage
- Prevents jobs from starting if there's a problem
  - E.g. if atlas.cern.ch CVMFS broken, only prevents new ATLAS jobs from starting

GridPP
UK Computing for Particle Physics

Provisioning worker nodes

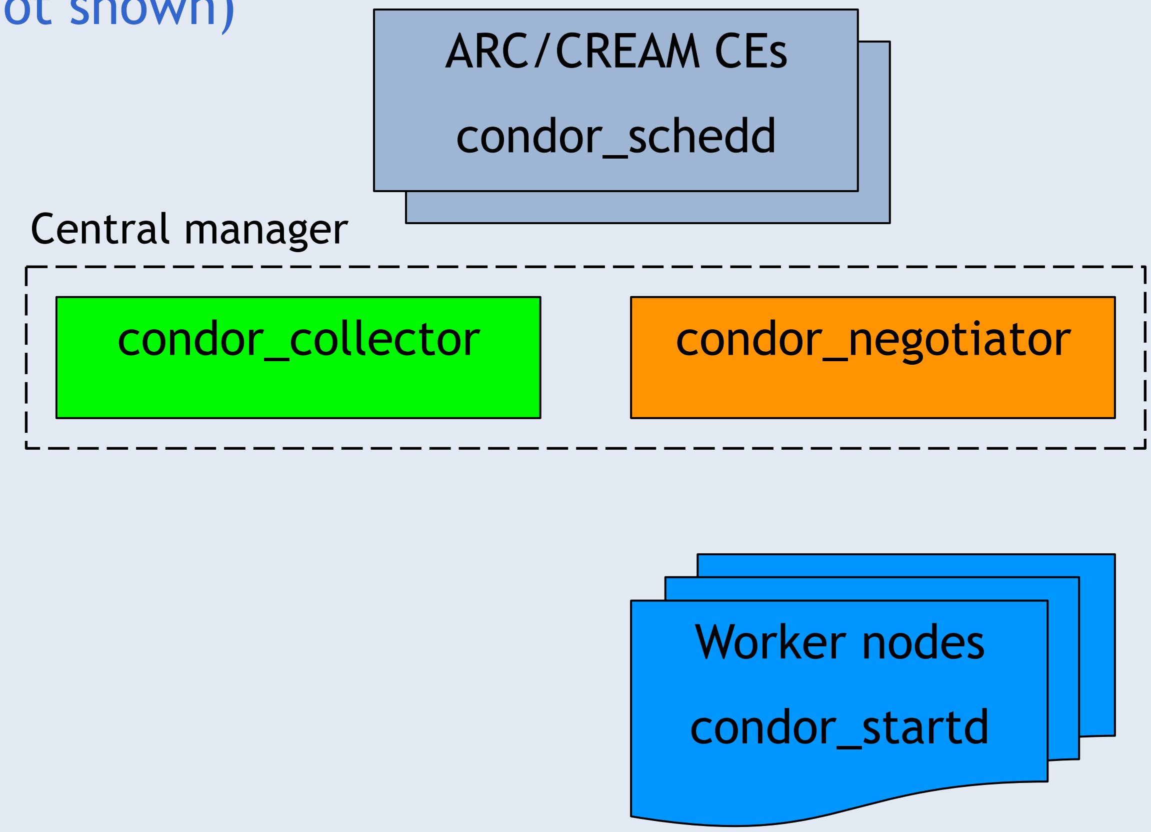
- Based on the following idea in the SLURM documentation on elastic computing
  - "SLURM's Elastic Computing logic relies heavily upon the existing power save logic"
- Apply the same method, using HTCondor's existing power management features
  - Entering a low power state
    - HIBERNATE expression can define when a slot is ready to enter a low power state. When true for all slots, machine will go into the specified low power state.
  - Machines in the low power state are "offline"; the collector can keep offline ClassAds
  - Returning from a low power state
    - condor_rooster daemon responsible for waking up hibernating machines under specified conditions
    - By default will send UDP Wake On LAN, but this can be replaced by a user specified script

Provisioning worker nodes

- Advertise appropriate offline ClassAd(s) to the collector
- condor_rooster
  - Enable this daemon (not enabled by default)
  - Configure to run appropriate command to instantiate a VM
    - Contextualisation using CloudInit
    - HTCondor pool password inserted into the VM
    - Volatile disks for /tmp, job scratch area & CVMFS cache created on hypervisor's local disk
- When there are idle jobs
  - Negotiator can match jobs to the offline ClassAd
    - Configured so that online machines are preferred to offline
  - condor_rooster daemon notices this match, instantiates a VM

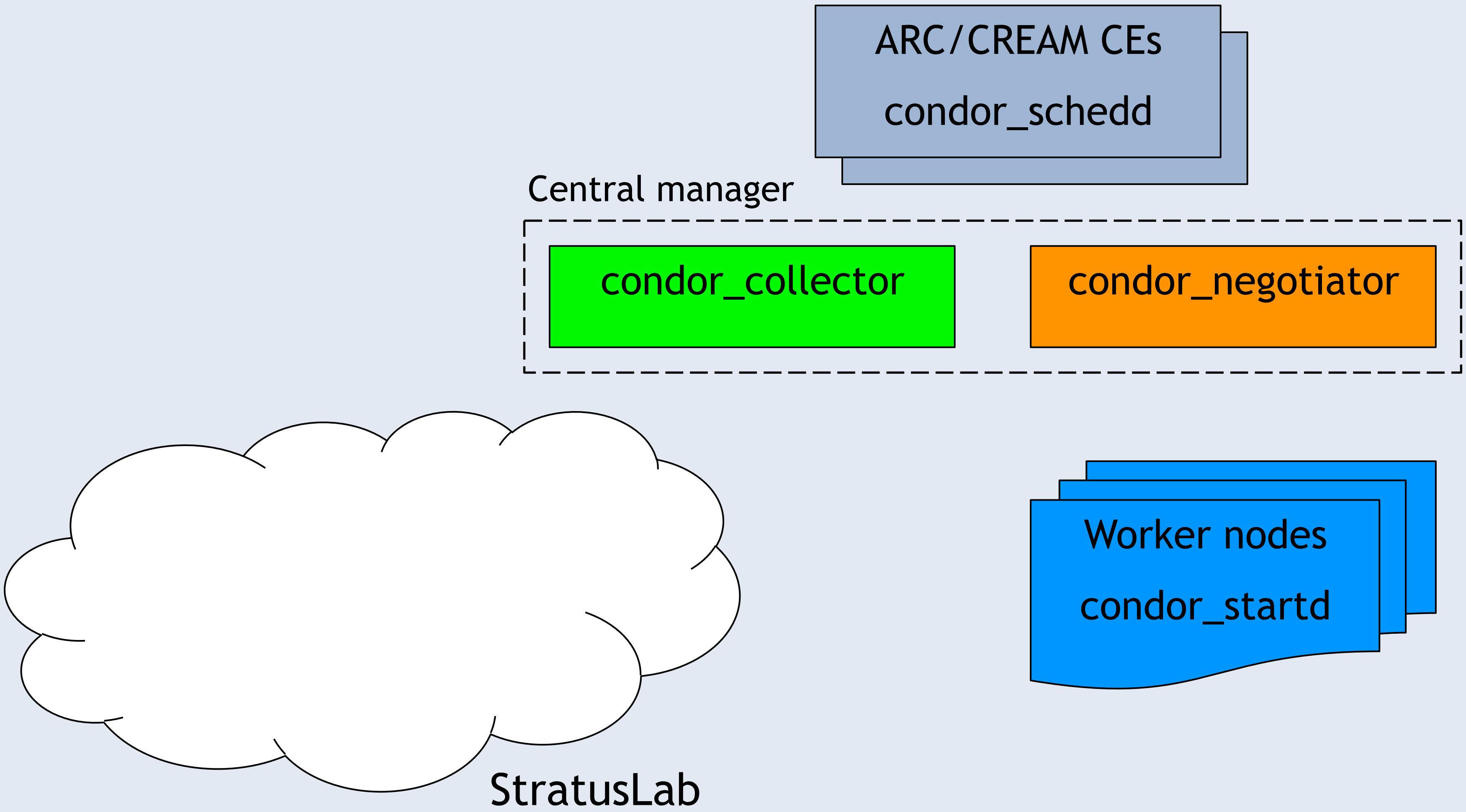
Provisioning worker nodes

- Original batch system
  (2nd central manager not shown)



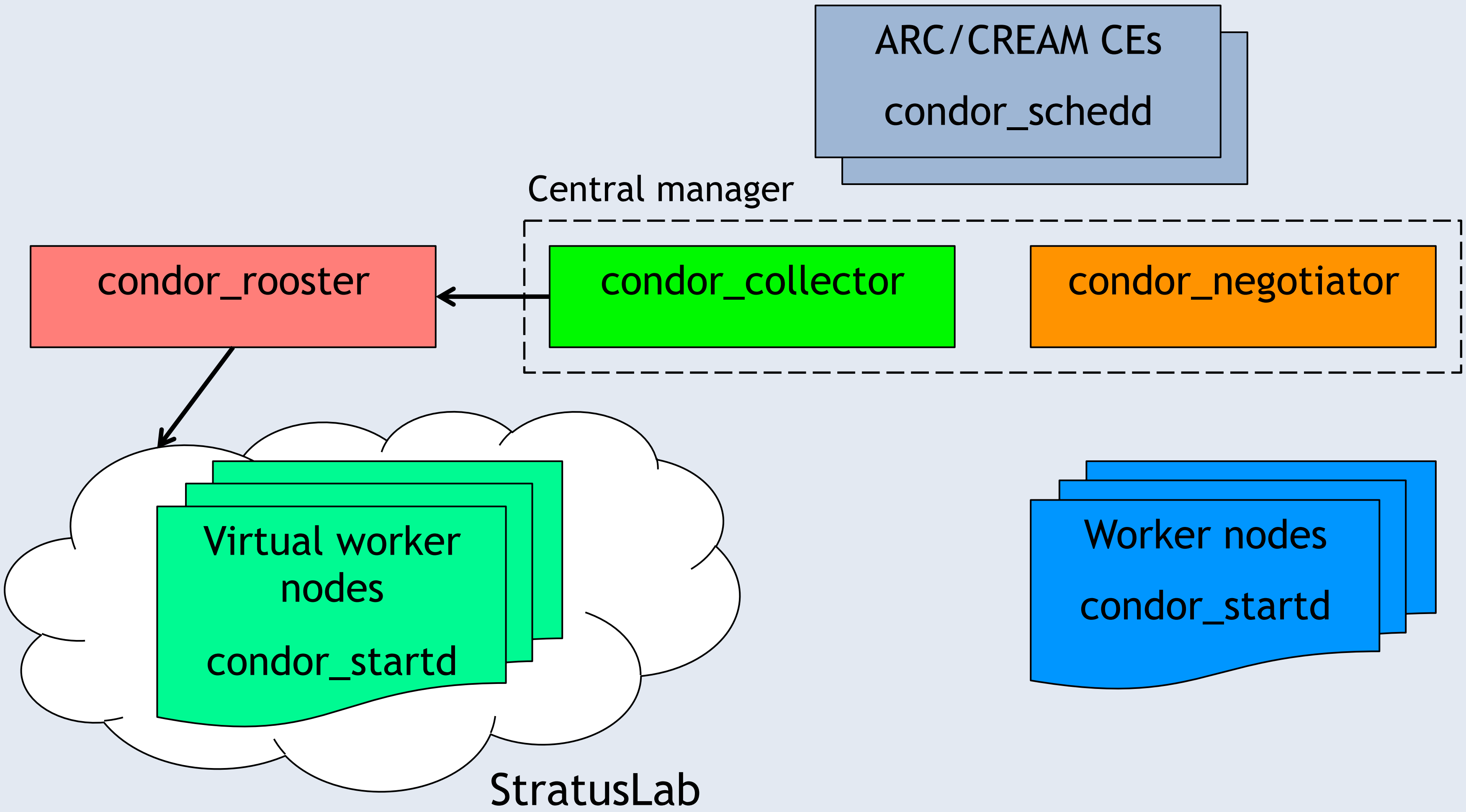
Provisioning worker nodes

- Cloud resources



Provisioning worker nodes

- Using cloud resources



GridPP
UK Computing for Particle Physics

VM lifetime

- Using short-lived VMs
  - Only accept jobs for a limited time period before shutting down
- HTCondor on the worker node controls everything
  - START expression
    - new jobs allowed to start only for a limited time period since the VM was instantiated
    - new jobs allowed to start only if the VM is healthy
  - HIBERNATE expression
    - VM is shutdown after machine has been idle for too long
- Very simple
  - Don’t need external systems trying to determine if VMs are idle or not

VM lifetime

- Benefits of short-lived VMs
  - Rolling updates easy, e.g. kernel errata
- Problems
  - Inefficiencies due to constantly draining worker nodes
  - Could be resolved by
    - Using single core VMs rather than multicore
    - Run short jobs if possible while long running jobs are finishing
- Alternatively, could use long-running VMs
  - Giving back resources to other cloud users takes longer

Fairshares

- Opportunistic use of the cloud
  - Almost always have idle jobs in the batch system
    - Would just completely take over the cloud
  - Can specify a fixed quota, but this isn't really enough
  - Unfortunately clouds don't support fairshares
- Currently using a simple method
  - Choose cores & memory per VM for cloud worker nodes such that there will always be some resources left per hypervisor
    - E.g. with 8 core 16 GB hypervisors, use 4 core 12 GB VMs
  - Simple starting point, but not ideal
    - E.g. what if other users want VMs with large numbers of CPUs or memory?

Virtualisation overheads

- Preliminary tests
  - 8 core physical & virtual worker nodes, same underlying hardware
- HEPSPEC06

Physical	69.45
Virtual	64.82

- CMS MC rereco

	CPU Efficiency	Time per Event
Physical	99.2%	14.7s
Virtual	97.3%	16.1s

Virtualisation overheads

GridPP
UK Computing for Particle Physics

- Xrootd copy 2 GB file to worker node

Physical	79 MB/s
Virtual	22 MB/s

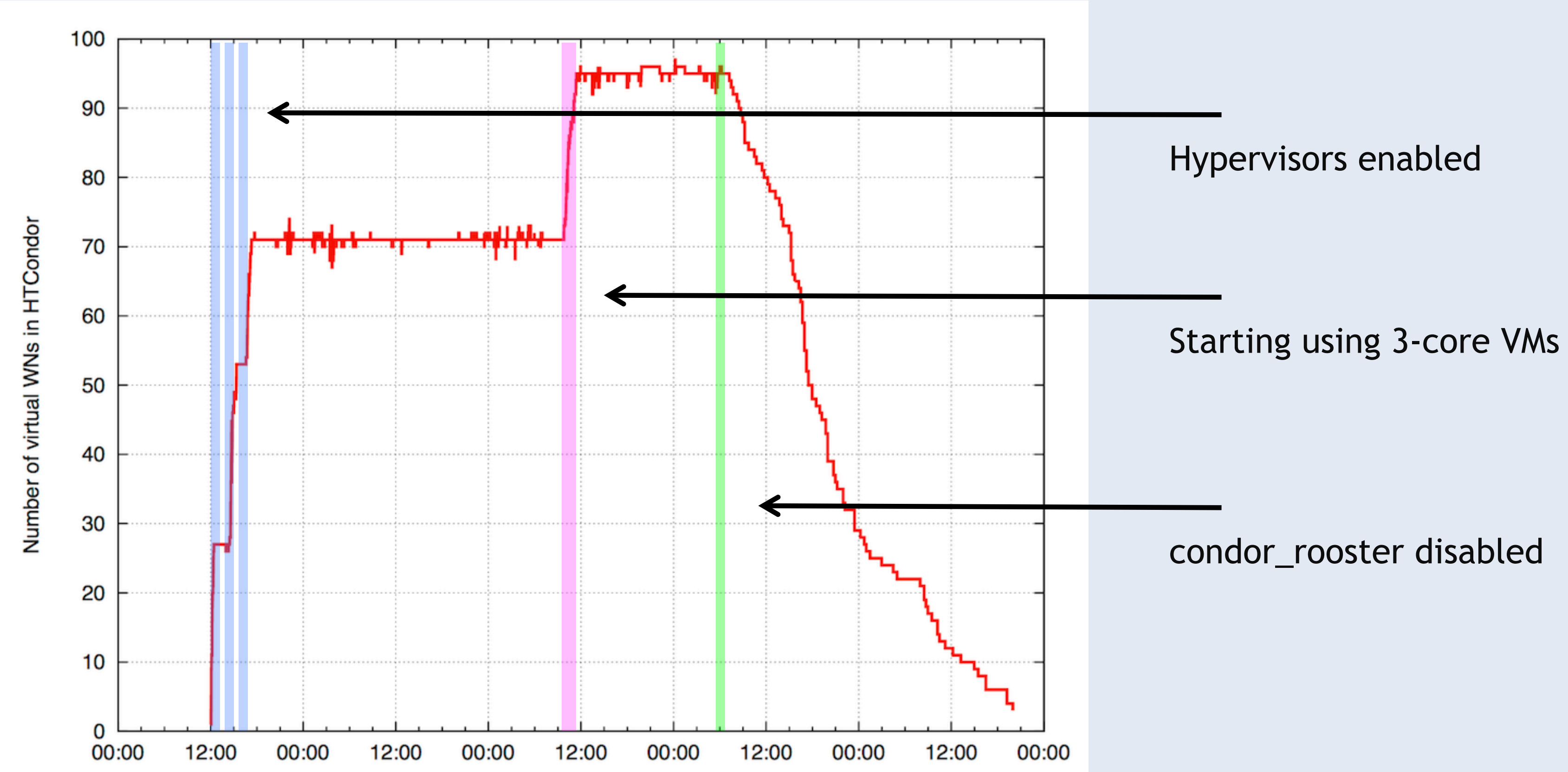
- Iozone single-stream test

	Write	Read
Physical	42 MB/s	56 MB/s
Virtual	9 MB/s	55 MB/s

- Note: these are all preliminary results, and we haven't attempted any tuning yet

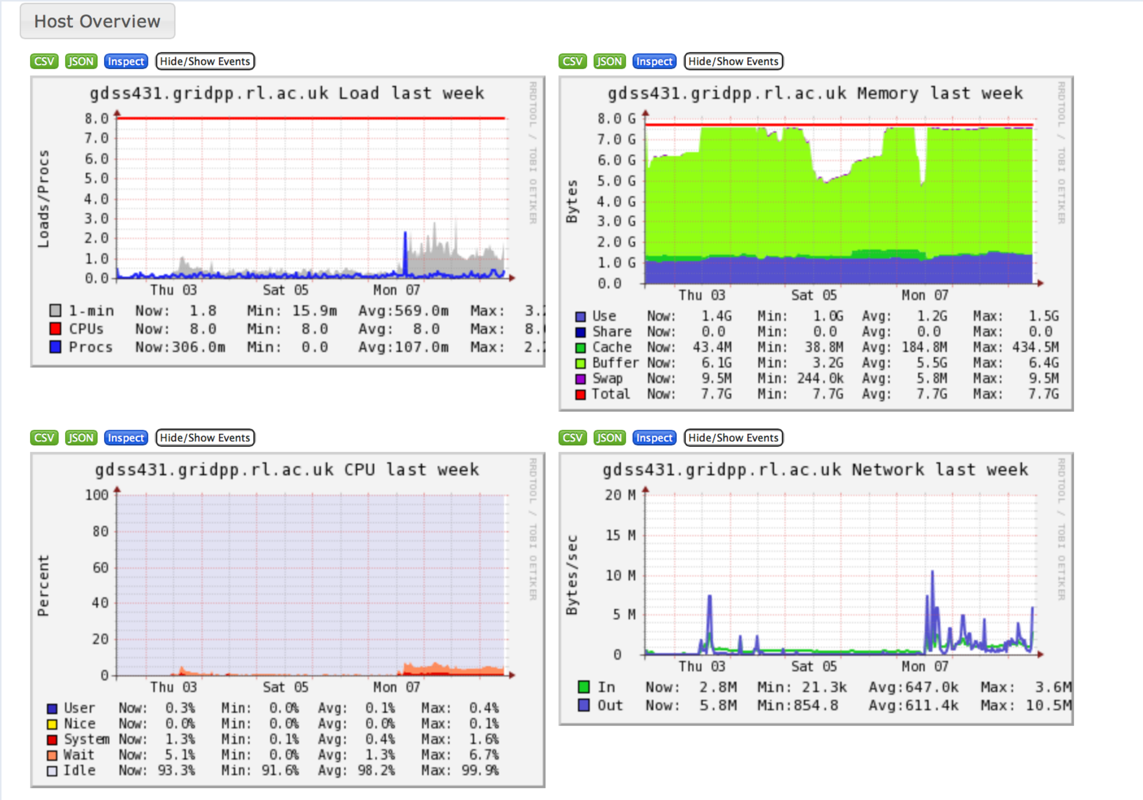
Testing

- Testing in production
  - Initial test with production HTCondor batch system
  - Ran around 11,000 real jobs, including all LHC VOs
  - Started with 4-core 12 GB VMs, then changed to 3-core 9GB VMs



Testing

- Load on persistent disk storage
  - Only /tmp, job scratch area, CVMFS cache on disk local to the hypervisor. Everything else on p-disk server.
  - Expected this to be the main source of scaling problems, but in small-scale testing there wasn’t much load



Monitoring

- Physical worker node monitoring at the RAL Tier-1
  - Pakiti
    - Monitors status of nodes with respect to patches
  - Logging
    - Log files (e.g. /var/log/messages) sent to central loggers
  - Nagios
    - 28 checks, including
      - CVMFS repositories
      - Space left on different partitions
      - Swap
      - Load
      - Dmesg
      - Linux NTP drift
      - Temperature
      - …

Monitoring

- Issues with dynamic resources
  - Nagios designed for static resources
    - Can't handle machines coming & going
    - Probably don't need Nagios at all for virtual worker nodes
      - Provided jobs won't start if WN is broken, and broken WNs eventually shutdown
  - Maintaining a history of VMs
    - What happens if a VM is created, causes lots of jobs to fail before finally shutting down?
    - What happens if there is a security incident involving a short-lived virtual WN?
  - Effect on infrastructure
    - What if the batch system encountered scaling problems overnight when lots of new worker nodes were added?

Next steps

- More thorough testing of virtualisation overheads
- Performance tuning
  - So far no attempt has been made to optimise performance of the VMs
- Evaluation of cloud management software
  - Currently using StratusLab, but will evaluate alternatives

GridPP
UK Computing for Particle Physics

Questions?