



Integrating VMs into WLCG

Andrew McNab

University of Manchester,
GridPP, and LHCb



Overview

- Scope
- Pilot VMs
- CernVM as WLCG worker node
- Sites, CEs, and Spaces
- Machine/job Features Task Force
- Traceability Task Force
- APEL accounting
- VM efficiency measurements
- SAM Tests



Scope

- This talk is based on our experiences in integrating CernVM-based VMs into WLCG to run jobs for LHC experiments
 - Not talking about running worker nodes that happen to be in VMs (eg lxbatch)
 - And not covering ways of creating VM worker nodes which are then accessed as part of a batch system (eg Cloud Scheduler)
 - Although some of the details are relevant to these scenarios to
 - These VMs are for ATLAS+LHCb (Manchester) and CMS (RAL)
- Also explain how some WLCG task forces interact with the problem of running jobs in VMs
- And some details of accounting and monitoring



Pilot VMs

- Assume the VMs have a defined lifecycle
- Need a boot image and user_data file with contextualisation
- Virtual disks and boot media defined and VM started
- The VM runs and executes the experiment's pilot framework client
 - Uses hostcert/key or proxy in contextualization to authenticate
 - ATLAS and LHCb are currently one payload per VM lifetime
 - CMS already runs multiple payloads per VM lifetime
- VM does shutdown -h when finished or if no more work available
 - Maybe also update a heartbeat file so stalled or overrunning VMs are killed
- Log files to /etc/machineoutputs which are saved
 - shutdown_message file can be used to say why the VM shut down
- Compatible with Vac, Vcycle, or Condor Vacuum lifecycle managers

CernVM as a WLCG worker node

- WLCG sites typically install and configure WNs with RPMS, Puppet, YAIM, local scripts
 - Frequently leads to job failures when an RPM is missing from WN
- VMs allow the experiments to have complete uniformity on WNs
- We use vm-bootstrap script, run by contextualization to set things up
- /cvmfs/grid.cern.ch is used to provide middleware
- Create /scratch from a 40GB+ partition provided by the hypervisor
 - Much faster than copy-on-write overlay root partition
- Increase the cvmfs cache to 11GB and put it on /scratch partition
- Create swap file on /scratch partition
- Set up /etc/machinefeatures, /etc/jobfeatures, Ganglia, etc

Sites, CEs, Spaces, WNs

- People interacting with WLCG sites are expecting worker nodes to be accessed through a Compute Element gateway, with one or more CEs making up a site
- Logging, monitoring, shifters etc are assuming this is still the case
- We follow this pattern with the VM-based sites
- Both Vac and Vcycle provide a FQDN space name to the VM, which the VMs give to the pilot framework as the CE name
- VMs set a FQDN hostname again for logging and monitoring, using the value from Vac or Vcycle via the contextualization
- If the pilot framework can record the batch job ID, we supply the UUID of the VM
 - This allows shifters to talk to site admins and be able to cross reference things right down to the level of the libvirt log files

Machine/job Features Task Force

- Proposed in HEPiX and current (finishing) WLCG task force
- Allows site/host to communicate details of machine and the job slot to the job or VM
 - HS06, shutdown time for VM/host, CPU and memory limits, ...
- One key file per key/value pair, in one of two directories
- Jobs use variables pointing to directories in the filesystem or to URLs
- The Vac factory node offers these directories to its VMs via NFS over its internal private network
 - Also a writeable NFS directory for log files, shutdown reason, heartbeat files etc
- Vcycle does this using HTTP(S) web server on the Vcycle machine
 - URLs discovered from OpenStack metadata machinefeatures and jobfeatures keys
- So VMs try these methods in turn so can set \$MACHINEFEATURES and \$JOBFEATURES for the jobs



VM Traceability Task Force

- This year WLCG has started a TF to look at traceability with VMs
- See Ian Collier's talks at last and next GDB
- Created in response to concerns from sites that it is harder for them to investigate security incidents inside VMs
- Scope includes:
 - Understanding what the introduction of VMs changes
 - What VM architects should do to promote traceability
 - How logging can be supplied to the site
 - How the site can monitor network traffic to/from VMs
 - How sites might keep or snapshot VMs for forensics
- Still in early stages, and would benefit from more experiments

Syslog prototype: machinefeatures

- Done as part of WLCG Traceability TF
- Added to LHCb VMs
- If `/etc/machinefeatures/syslog` exists, then it is used to create configuration file `/etc/rsyslog.d/vm.conf` in VM to send a copy of syslog messages to an external rsyslogd
- Vac can add custom values to machinefeatures based on configuration file
- Current tests with rsyslogd on each factory machine
 - Site can then store messages on factory for X days, filter and forward to a central service, etc as desired
- UDP syslog protocol probably a good choice as universally supported now and very easy to gateway to other protocols in the future

Syslog prototype: contextualization

- Machinefeatures has the advantage of being VM-architecture neutral
- However, script to access it is only run at the end of the boot up
- To get earlier messages, need vm.conf file created at the start
- To address this for us, Jakob has prototyped the ability to supply files to CernVM in the uCernVM configuration
 - base64 encoded tar.gz unpacked in “/” so could be used for all sorts of other cool customisations too!
- Looking at how best to provide this:
 - Site creates appropriate base64 blob? (Always the same for Vac since always 169.254.169.254:514)
 - Or make Vac/Vcycle aware of syslog with a syslog_server option and do it all automagically for the site admin?

APEL Accounting

- Many sites need to report usage through APEL and EGI Accounting
- This morning's talk explain how Vac and Vcycle write out APEL SSM messages to disk when VMs finish
- To fit this into what people expect in WLCG (and EGI) context:
 - Site admin can give the “GOCDB” name in the configuration, included in the messages
 - UKI-NORTHGRID-MAN-HEP
 - The space name and vmtype are used to construct the SubmitHost
 - vac02.tier2.hep.manchester.ac.uk/vac-atlasprod
 - The VM UUID is used as the LocalJobId
 - The space name is converted into a “user” DN
 - /DC=uk/DC=ac/DC=manchester/DC=hep/DC=tier2/DC=vac02
 - The FQAN for the experiment is included if given in the configuration
 - /lhcb/Role=NULL/Capability=NULL
- The net result is the records end up associated with the site in exactly the same way as records from other conventional grid/batch CEs at the site

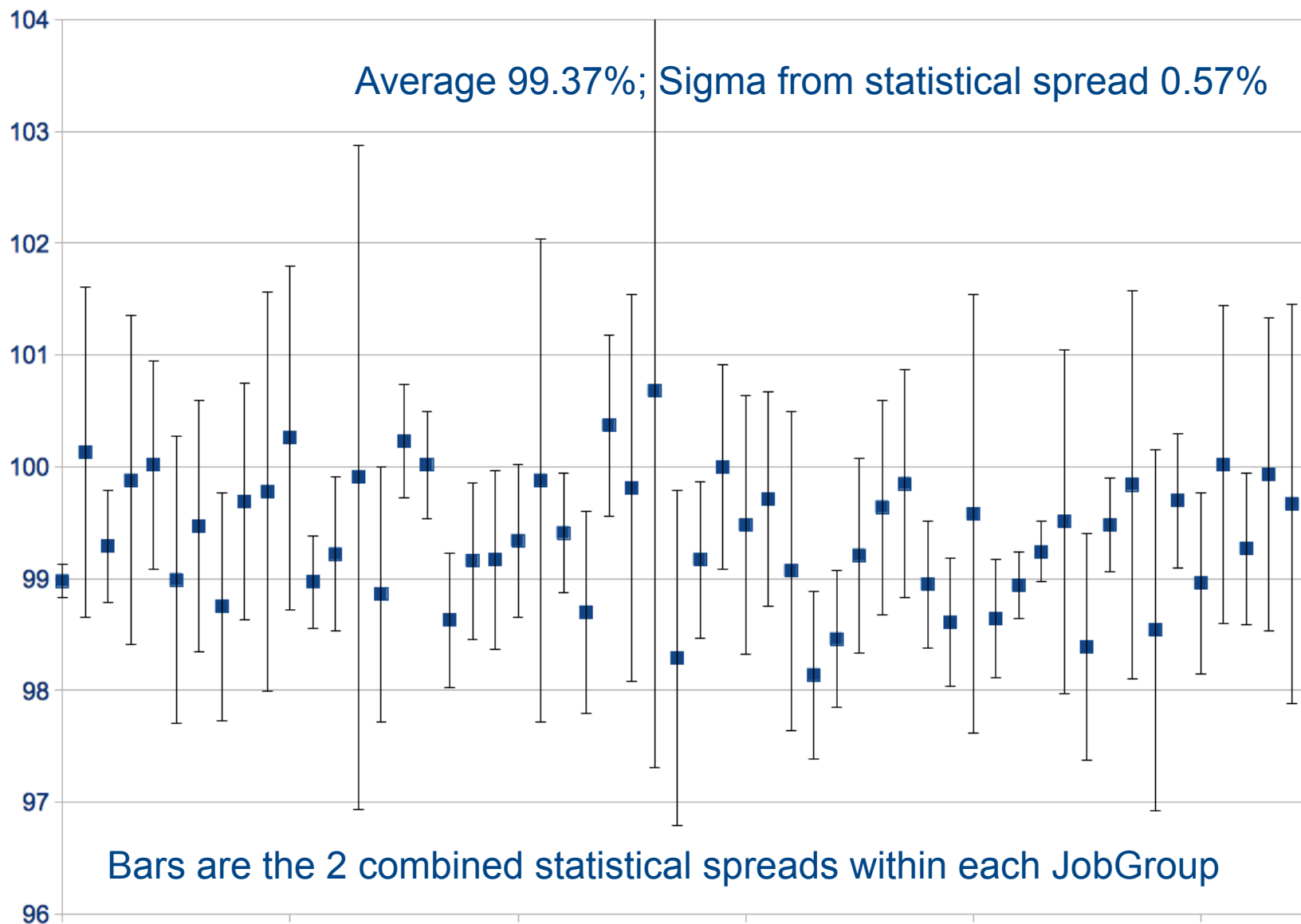
Efficiency measurements for HEPiX May'14

- What is today's intrinsic cost of using VMs rather than real machines?
 - Lots of hard work done by many people: VT-x, kvm, qemu-kvm, ksm, virtio
- Can get overall cpu/wall efficiency measurements from accounting
- At Manchester, these are comparable for Batch vs VMs
 - So for April/May 2014, the weighted average ATLAS+LHCb efficiencies for batch and Vac VMs were both 92.0%
 - That doesn't distinguish between setup CPU, overhead CPU, and payload CPU
- People have made direct benchmark comparisons or done test runs
 - But can we measure this with routine workloads too?
- We had machines from the same 2010 purchase running both
 - 288 VM slots on Vac; and ~1000 PBS batch job slots
- Could do direct comparison of efficiency of production payload jobs

VM vs Batch efficiency with LHCb MC

- LHCb Monte Carlo production is organised as JobGroups
 - Each job in a JobGroup is part of one production, with the same parameters
 - Expect similar amount of time in event generator, detector physics, reco etc
- We run enough jobs at Manchester that jobs from the same JobGroup will be run on both Batch and VM
 - Sample was 5534 (3216+2318) jobs in 56 JobGroups, 15-22 May 2014
- Using LHCb DIRAC monitoring of payload jobs, can calculate the CPU/Walltime efficiencies for the Batch jobs and for the VM jobs within the same JobGroup separately
- Can then take quotient to give VM/Batch efficiency ratio
 - Also calculate statistical spreads within each group
- The machines we used were dual 6-core Westmere from 2010, with HT enabled, 2GB/slot, and running 24 batch jobs or 24 VMs

VM eff. / Batch eff., per LHCb MC JobGroup



15-22 May
2014

56 LHCb MC
JobGroups;
5534 jobs

3216 Batch
jobs; 2318 VM
jobs

Machines:
12-core
Westmere,
HT enabled,
2GB/slot,
24 job or
VM slots



SAM tests

- WLCG SAM tests are jobs submitted to sites via CREAM or ARC CE by the experiments
- This model isn't possible with the Pilot VMs since they are started by the site rather than the experiment
- Solution needed for VM-only sites otherwise they will not be able to demonstrate their availability and reliability for WLCG pledges
 - Also important for identifying problems too of course ...
- Looking at running SAM tests in the VM as part of initialisation
 - Perhaps just run it a fraction of the time
 - Reporting via messages as with conventional SAM tests



Summary

- Described steps to use CernVM as a WLCG worker node
 - “Pilot VMs”
- WLCG Machine/Job Features task force concluding
 - Very useful to provide information to VMs
- WLCG Traceability task force starting
 - Prototype syslog logging in LHCb VMs
- VM accounting can be integrated into grid-level APEL accounting
- Have demonstrated equivalent VM/native performance for LHCb MC
- WLCG experiments’ SAM tests need to be supported in VMs



Extra slides

The Masonry Problem...



The Masonry Problem

