Vac, Vcycle, VMs status and plans

Andrew McNab
University of Manchester
LHCb

Overview

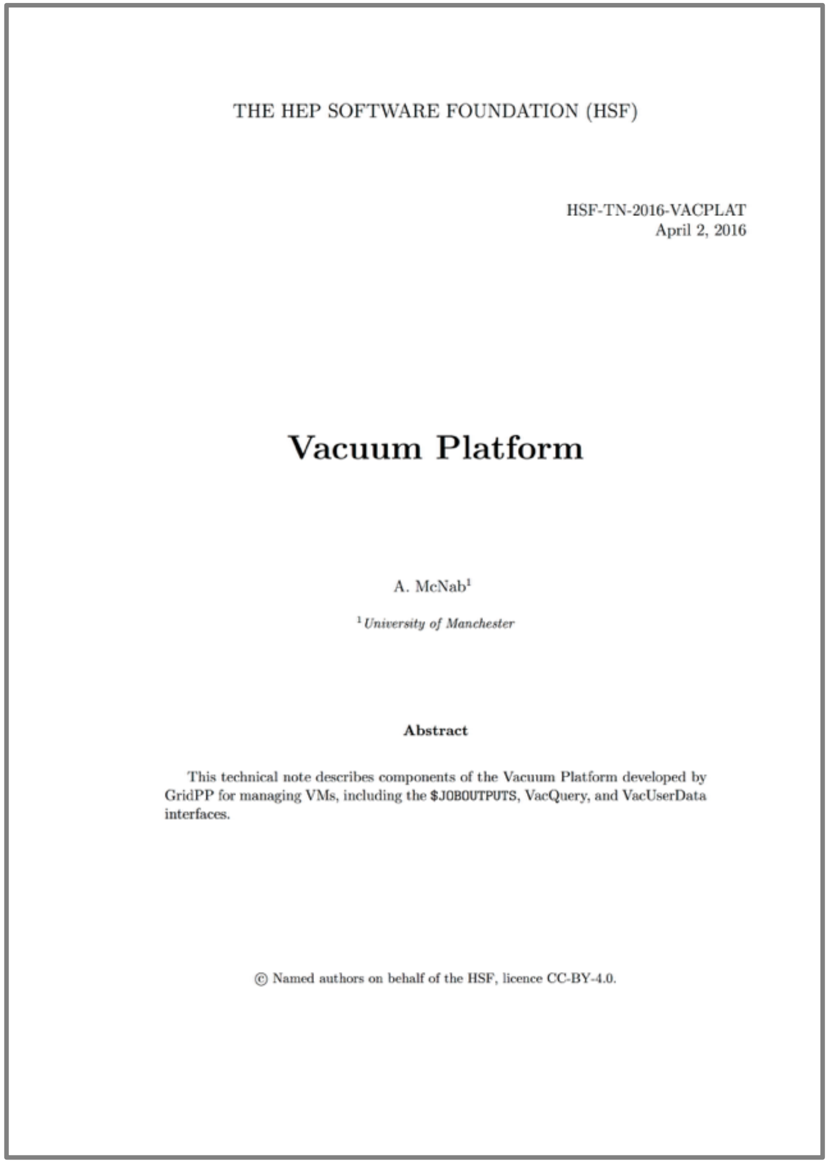
- Vac vs Vcycle
- Vacuum Platform
- Vac and Vcycle status
- LHCb and GridPP DIRAC VMs
- Cloud Init ATLAS VMs
- CMS VMs
- Next steps
  - VacMon
  - Vacuum Pipes
  - VM size mixing
  - Containers

Vac vs Vcycle recap

- Two GridPP systems aimed at running VMs
- Vac - autonomous hypervisors
  - Each VM factory machine creates VMs in response to observed demand for each type of VM
  - More mature of the two, better documentation
- Vcycle - uses OpenStack etc
  - Factories created via Cloud API in response to observed demand for each type of VM
  - Code is solid, but docs are minimal: just man pages

Vacuum Platform

- Drafting an HSF technical note describing the interfaces between VMs and Vac/Vcycle
- For VM-authors and authors of other Vac/Vcycle-like systems
- Proposing this to EGI as the basis of a “community platform”
- VacQuery / VacMon
- VacUserData
- $JOBOUTPUTS
- Vacuum Pipes (see later)



THE HEP SOFTWARE FOUNDATION (HSF)

HSF-TN-2016-VACPLAT
April 2, 2016

Vacuum Platform

A. McNab[1]

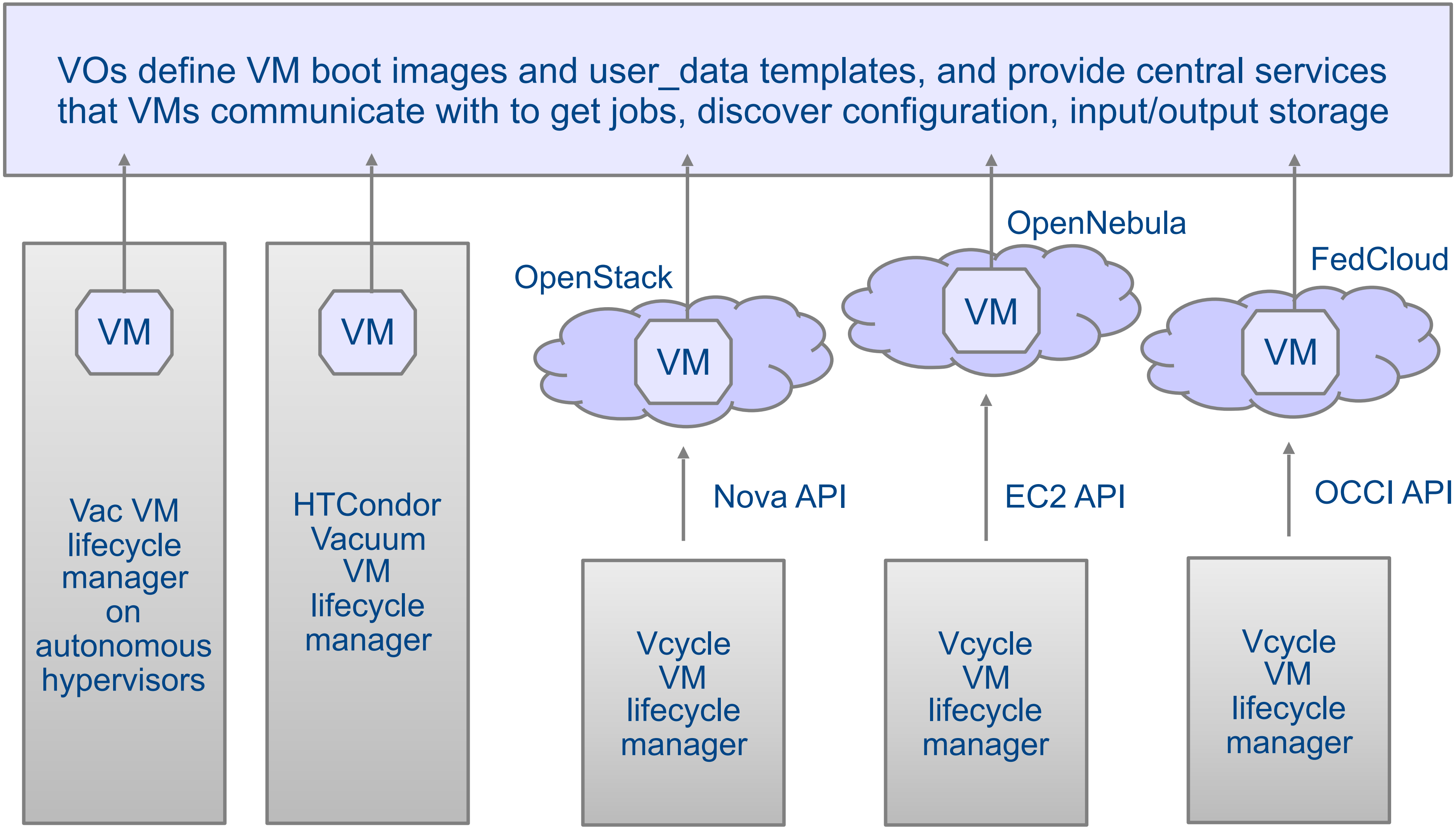
[1] *University of Manchester*

Abstract

This technical note describes components of the Vacuum Platform developed by GridPP for managing VMs, including the $JOBOUTPUTS, VacQuery, and VacUserData interfaces.

© Named authors on behalf of the HSF, licence CC-BY-4.0.

Vacuum platform



VOs define VM boot images and user_data templates, and provide central services that VMs communicate with to get jobs, discover configuration, input/output storage

VM

Vac VM lifecycle manager on autonomous hypervisors

VM

HTCondor Vacuum VM lifecycle manager

OpenStack

VM

Nova API

Vcycle VM lifecycle manager

OpenNebula

VM

EC2 API

Vcycle VM lifecycle manager

FedCloud

VM

OCCI API

Vcycle VM lifecycle manager

Vcycle status

- VMs created via Cloud API in response to observed demand for each type of VM
- Default is OpenStack plugin
- EC2 plugin written at the start of 2016
  - Only tested with OpenStack EC2 so far!
- OCCI (EGI), DBCE, and Azure (MS) plugins contributed by CERN
- Vcycle is used to manage LHCb OpenStack tenancy at CERN (500 VMs)
- Also LHCb tenancy at CC-IN2P3 and GridPP at Imperial
- Code and man pages good, but no admin guide etc

Vac status

- Vac 01.00 release ready (after this meeting)
- Now provides an OpenStack-compatible environment to VMs, but using autonomous hypervisors (VM factories)
  - Much simpler implementation: 3500 lines of Python vs 1,000,000 for OpenStack
- Work done on refactoring VacQuery UDP protocol used for inter-VM-factory communication to make it
  - more scalable
  - more robust against even high (50%) packet loss levels
- Squid-on-factory configuration included in Puppet module
- Machine/Job Features updated for HSF-TN-2016-02
- Fixes/improvements - thanks to useful feedback from sites!

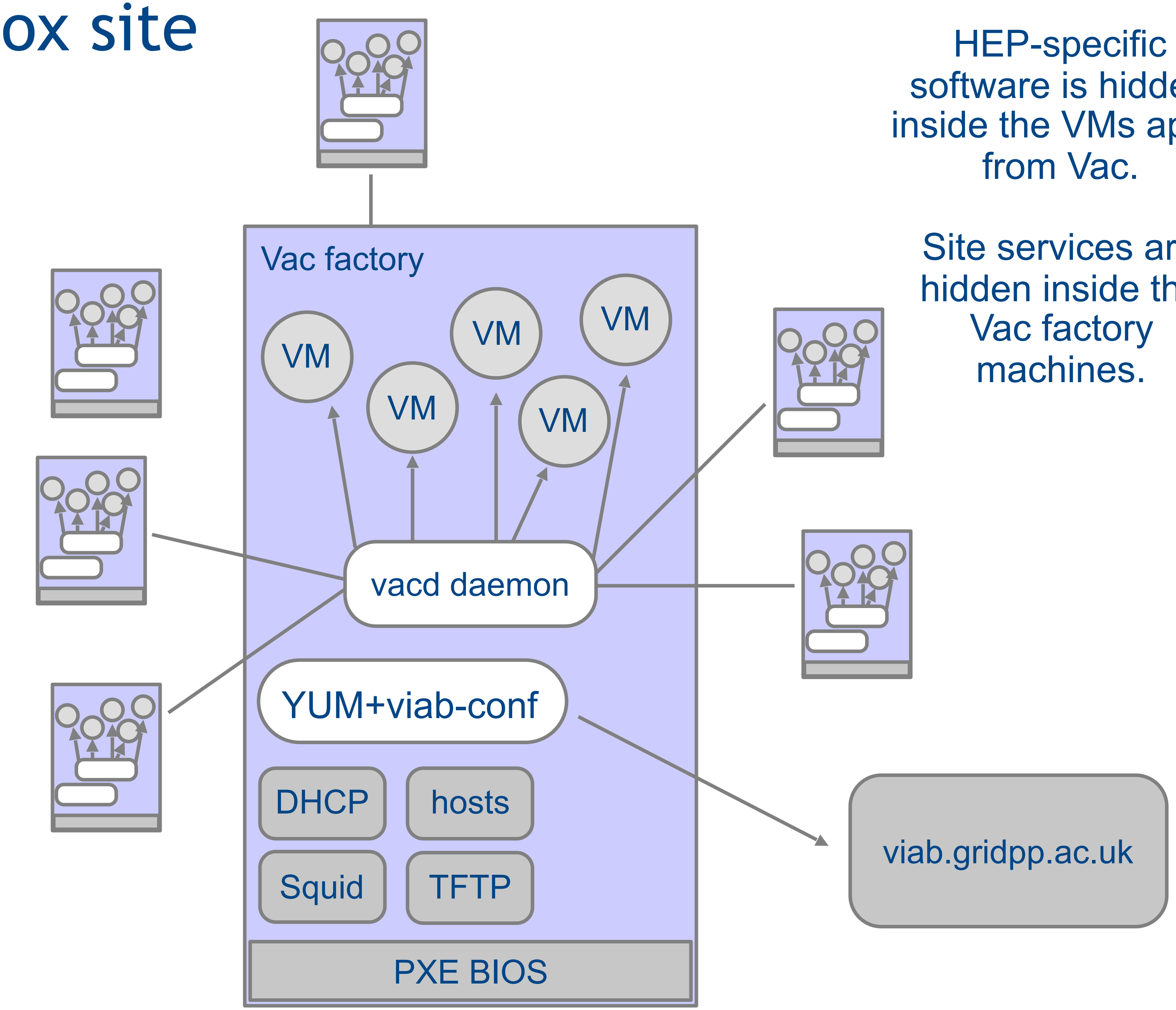
Vac-in-a-Box site

Simpler than installing via Puppet, Ansible etc.

Per-site dashboard at viab.gridpp.ac.uk

Kickstart from the website.

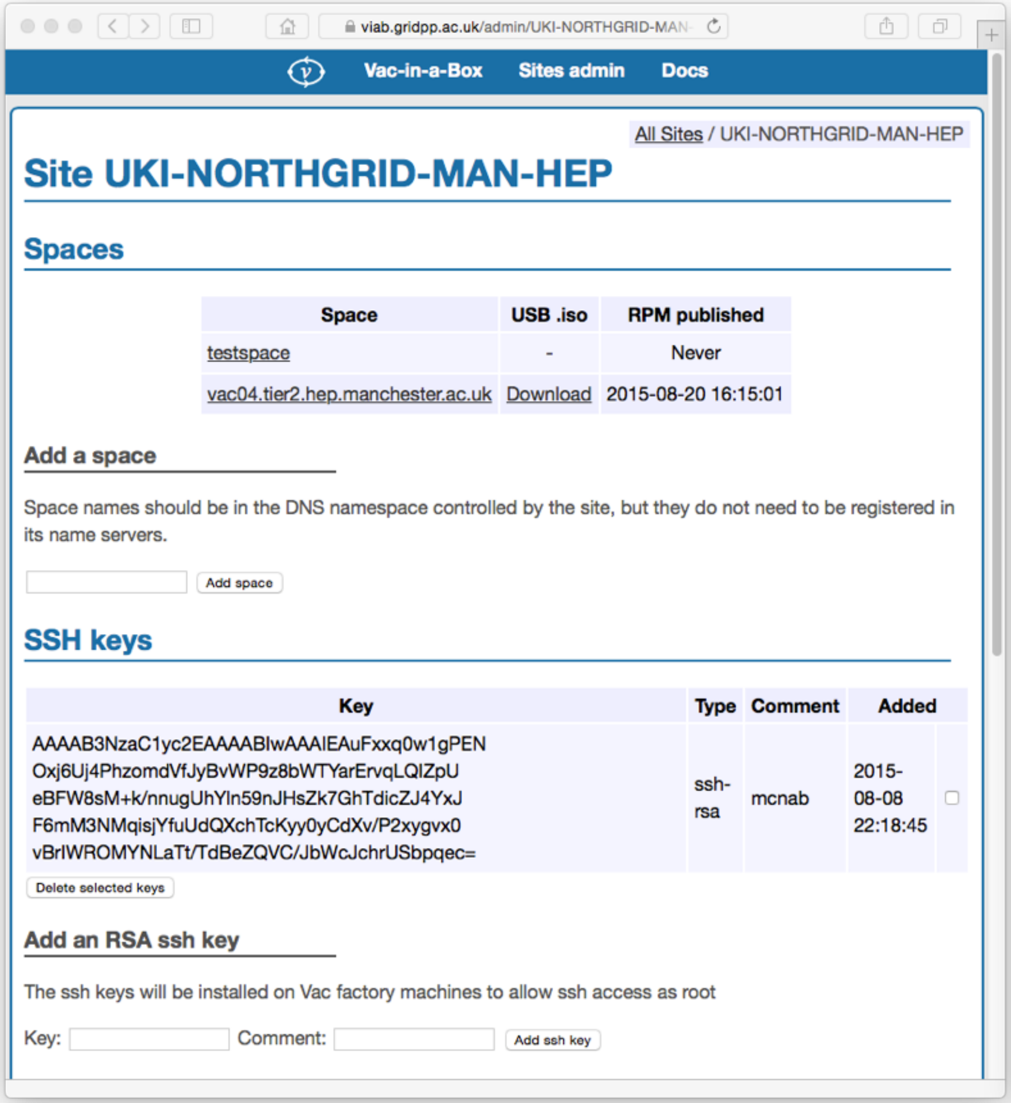
viab-conf RPM with configuration, via autoupdates from YUM repo.



HEP-specific software is hidden inside the VMs apart from Vac.

Site services are hidden inside the Vac factory machines.

Vac-in-a-Box dashboard



All Sites / UKI-NORTHGRID-MAN-HEP

Site UKI-NORTHGRID-MAN-HEP

Spaces

Space	USB .iso	RPM published
testspace	-	Never
vac04.tier2.hep.manchester.ac.uk	Download	2015-08-20 16:15:01

Add a space

Space names should be in the DNS namespace controlled by the site, but they do not need to be registered in its name servers.

Add space

SSH keys

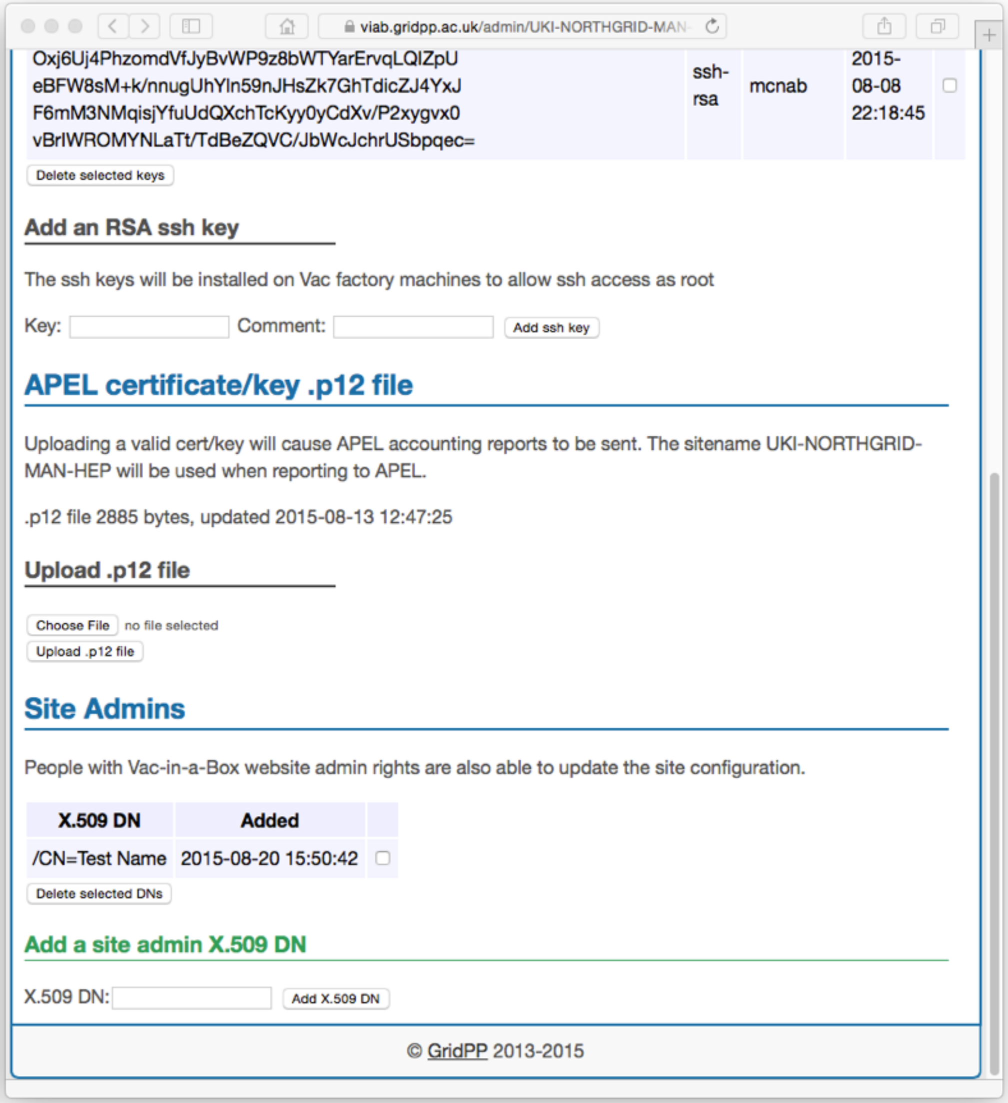
Key	Type	Comment	Added	
AAAAB3NzaC1yc2EAAAABIwAAAIEAuFxxq0w1gPEN Oxj6Uj4PhzomdVfJyBvWP9z8bWTYarErvqLQIZpU eBFW8sM+k/nnugUhYIn59nJHsZk7GhTdicZJ4YxJ F6mM3NMqisjYfuUdQXchTcKyy0yCdXv/P2xygvx0 vBrIWROMYNLaTt/TdBeZQVC/JbWcJchrUSbpqec=	ssh-rsa	mcnab	2015-08-08 22:18:45	☐

Delete selected keys

Add an RSA ssh key

The ssh keys will be installed on Vac factory machines to allow ssh access as root

Key: Comment: Add ssh key



Oxj6Uj4PhzomdVfJyBvWP9z8bWTYarErvqLQIZpU eBFW8sM+k/nnugUhYIn59nJHsZk7GhTdicZJ4YxJ F6mM3NMqisjYfuUdQXchTcKyy0yCdXv/P2xygvx0 vBrIWROMYNLaTt/TdBeZQVC/JbWcJchrUSbpqec=	ssh-rsa	mcnab	2015-08-08 22:18:45	☐

Delete selected keys

Add an RSA ssh key

The ssh keys will be installed on Vac factory machines to allow ssh access as root

Key: Comment: Add ssh key

APEL certificate/key .p12 file

Uploading a valid cert/key will cause APEL accounting reports to be sent. The sitename UKI-NORTHGRID-MAN-HEP will be used when reporting to APEL.

.p12 file 2885 bytes, updated 2015-08-13 12:47:25

Upload .p12 file

Choose File no file selected

Upload .p12 file

Site Admins

People with Vac-in-a-Box website admin rights are also able to update the site configuration.

X.509 DN	Added	
/CN=Test Name	2015-08-20 15:50:42	☐

Delete selected DNs

Add a site admin X.509 DN

X.509 DN: Add X.509 DN

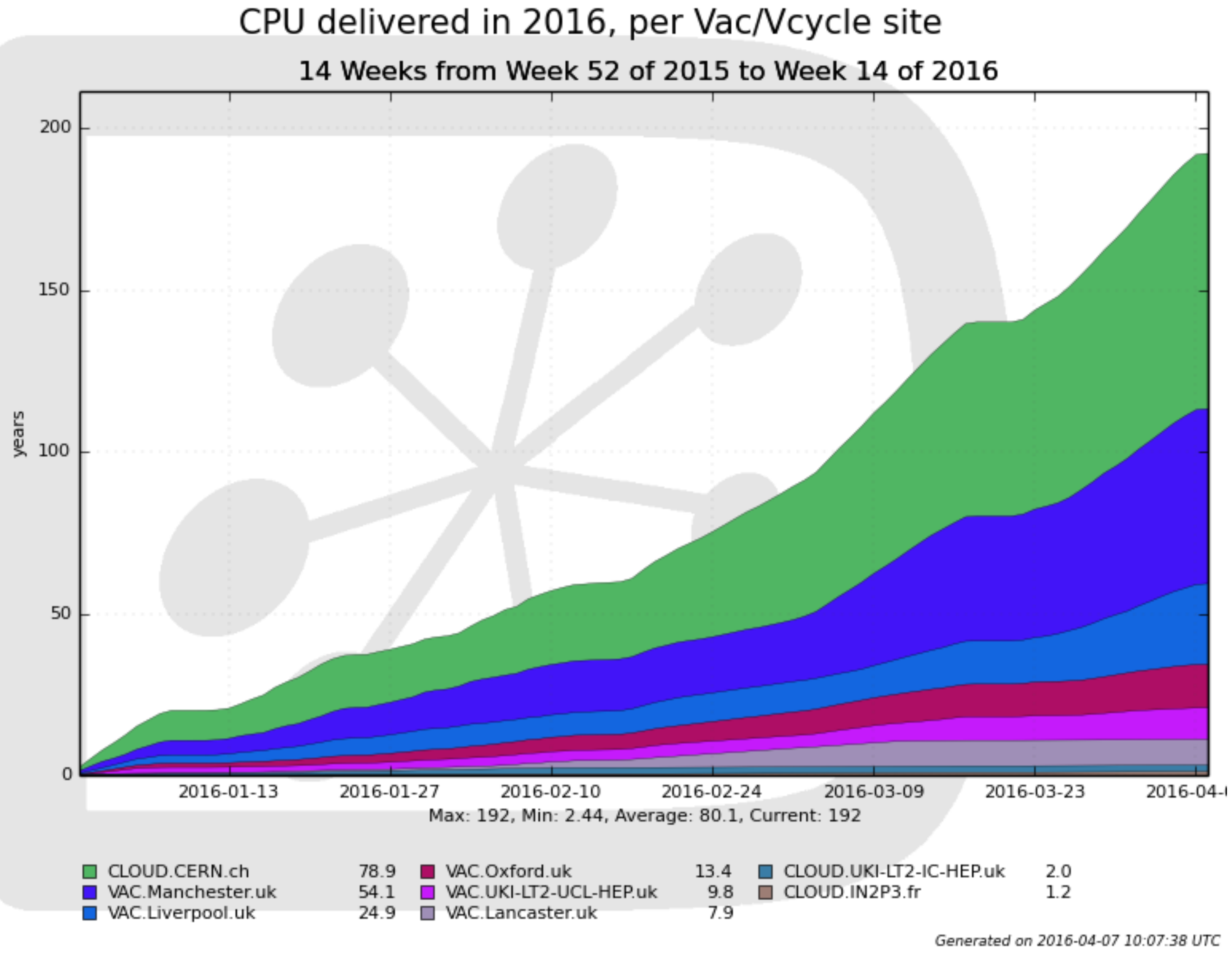
© GridPP 2013-2015

Now on to VMs …

LHCb DIRAC and GridPP DIRAC VMs

- Converging structure of the different DIRAC VMs
  - LHCb is generalising the DIRAC Pilots so the existing modular format can accommodate batch and VM scenarios for LHCb and other flavours of DIRAC
- This will become a standard part of DIRAC, with a generic “DIRAC VM” based on Cloud Init
- GridPP DIRAC will be the first to benefit from this
- Also supports multiple concurrent payloads per pilot
  - So can have efficient multiprocessor VMs even if only single processor VMs available
- In meantime, existing GridPP VMs modified to support any GridPP DIRAC sub-VO (Pheno, LSST, etc etc)

LHCb DIRAC VM in action

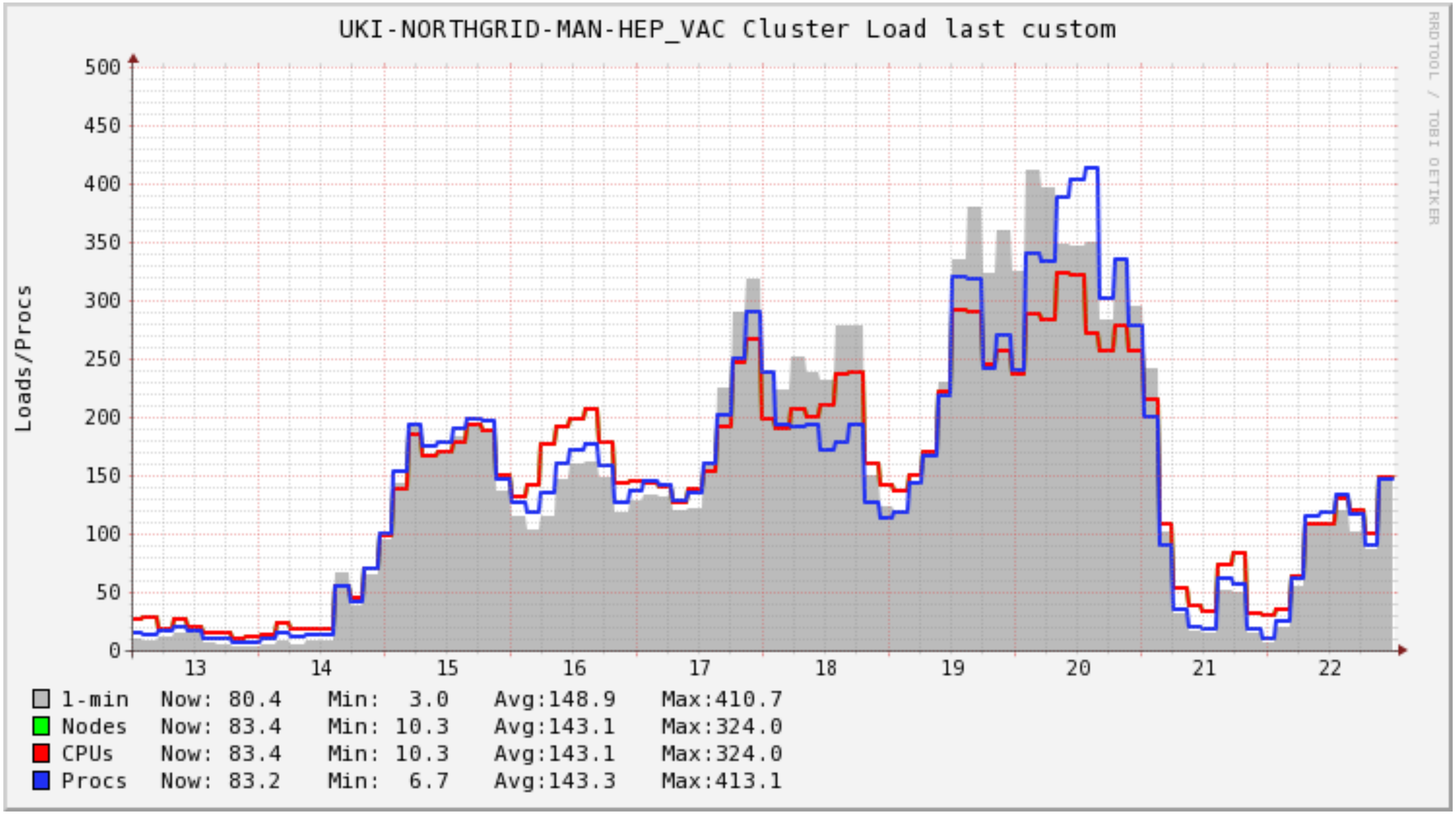


Cloud Init ATLAS VMs

- GridPP’s original ATLAS VMs just ran the PanDA pilot script inside the VM
  - This works but we’re basically on our own supporting it
- Instead we’ve made a second generation of ATLAS VMs, which use HTCondor to get pilots
  - Derived from VMs used on DBCE/CERN 2nd procurement
  - Same model as used by Sim@P1 VMs and CloudScheduler
  - Aim is to converge all ATLAS VMs as much as possible
- Corresponding HTCondor services at CERN set up as production services, alongside pilot factories etc.
- In process of rolling this out to other VM-based sites and tuning AGIS site parameters

New ATLAS VMs in action

- Using new VMs: one processor per VM



▨ 1-min	Now: 80.4	Min: 3.0	Avg:148.9	Max:410.7
🟩 Nodes	Now: 83.4	Min: 10.3	Avg:143.1	Max:324.0
🟥 CPUs	Now: 83.4	Min: 10.3	Avg:143.1	Max:324.0
🟦 Procs	Now: 83.2	Min: 6.7	Avg:143.3	Max:413.1

CMS VMs

- Original CMS VMs produced by Andrew Lahiff worked well
  - Not running at the moment but could be resurrected if needed
- CMS is looking at how to put externally run VMs on a proper production basis
- Similar model to ATLAS:
  - HTCondor within the VMs talking to standard CMS HTCondor machinery at CERN
- Again, this is very compatible with the vacuum model
  - HTCondor in the VMs can wake up, talk to CERN and try to pull in jobs.
  - Shutdown if none found or work finished.

Next steps …

(a.k.a. “Jam tomorrow”)

Next steps: VacMon

- Built into Vac 00.19 onwards
  - Can send VacQuery JSON messages to one or more VacMon services
  - vacmond puts JSON into ElasticSearch
- Still working on how best to present this, probably with Kibana (have looked at Grafana too)
- Sufficient info to replicate Ganglia-style monitoring of VM factories and of experiments’ VM usage
  - eg CPU load on VM factories; total HS06 allocated per type of VM etc
- As we know, good monitoring key to maintaining sites properly
  - Could be added to ROD shifts?

Next steps: Vacuum Pipes

- “Pipelines supplying VM components to VM factories”
- To define a VM in Vac and Vcycle requires a few lines of configuration
  - URL of user_data contextualization file
  - URL of boot image
  - Times: lifetime, heartbeat timings, “fizzle time”
- Vacuum Pipes will be a single URL with all this in a JSON file
- This means that adding a new VO to a site will involve adding one URL to config
- Probably still need X.509 cert/key for authentication to VO
  - But for GridPP DIRAC, all VOs use the same cert/key

Next steps: VM size mixing

- Vac can already deal with single or multiprocessor VMs
  - But all VMs created with the same geometry
- Want to be able to provide multiprocessor VMs for ATLAS (and CMS)
- But may not have enough GridPP DIRAC payloads to fill an 8-processor VM with 8 single processor payloads on a VM factory configured for that
- Will add option to define “superslots” of (say) 8 processors, in which all VMs created with the same finish time
  - Fit 8 or 1 processor VMs into free space in superslots
- Rely on LHCb interruptible Monte Carlo jobs/VMs to soak up time left by shorter VMs?

Next steps: containers

- CernVM group now offering a technology preview of container-based machines using Docker
- Plan is to use this to offer containers managed by Vac
    - Model is to run CernVM-FS on the factory
    - So a trade-off in managing that vs advantages of containers
- It may be possible to use Linux namespaces in sufficiently late kernels to run CernVM-FS inside the container
    - User processes can have admin capabilities inside containers
- So where I say “virtual machine” in these slides, I could say “logical machine” to be general

Next steps: generic HTCondor VM

- Say you have a local Tier-3 HTCondor batch system
- And you want to run Vac for “mission critical” Tier-2 WLCG/GridPP workloads
- But want local users using direct job submission to be able to use unused capacity on WLCG’s quiet days
  - Maybe even to manage Tier-3 funded resources within your Tier-2 infrastructure of Puppet, racks, network etc
- Aim to provide a generic VM definition which can be configured to get jobs from local HTCondor
  - Usual Vac backoff and target shares mechanism for deciding when to do this
- Could be used on Vcycle/OpenStack too

Summary

- Vac and Vcycle both on a firm footing
  - Vac now at 01.00 release stage
- Vacuum Platform specification
- Clear progress/plans in getting VMs for LHCb, GridPP DIRAC, ATLAS, and CMS on to a production basis
- More things in the pipeline:
  - VacMon, Vacuum Pipes, VM size mixing, Containers