Vac and Vcycle

Luis Villazon Esteban

On behalf of
Andrew McNab

University of Manchester
LHCb and GridPP

Overview

- Vac and Vcycle
- Vacuum model
- Machine/job features
- Accounting
- Usage
- Future plans

Vac and Vcycle

- Both VM Lifecycle Managers
- Vac is a standalone daemon you run on each worker node machine to create its VMs
- Vcycle manages VMs on IaaS Clouds like OpenStack
  - Can be run at the site, by the experiment, or by regional groups like GridPP
- Both developed at Manchester as part of GridPP Clouds/VMs effort
  - With help from Lancaster, Oxford, IC, CERN, LHCb and ATLAS
- Both make very similar assumptions about how the VMs behave
  - The same LHCb and ATLAS VMs working in production on Vac and Vcycle

“Pilot VM” lifecycle

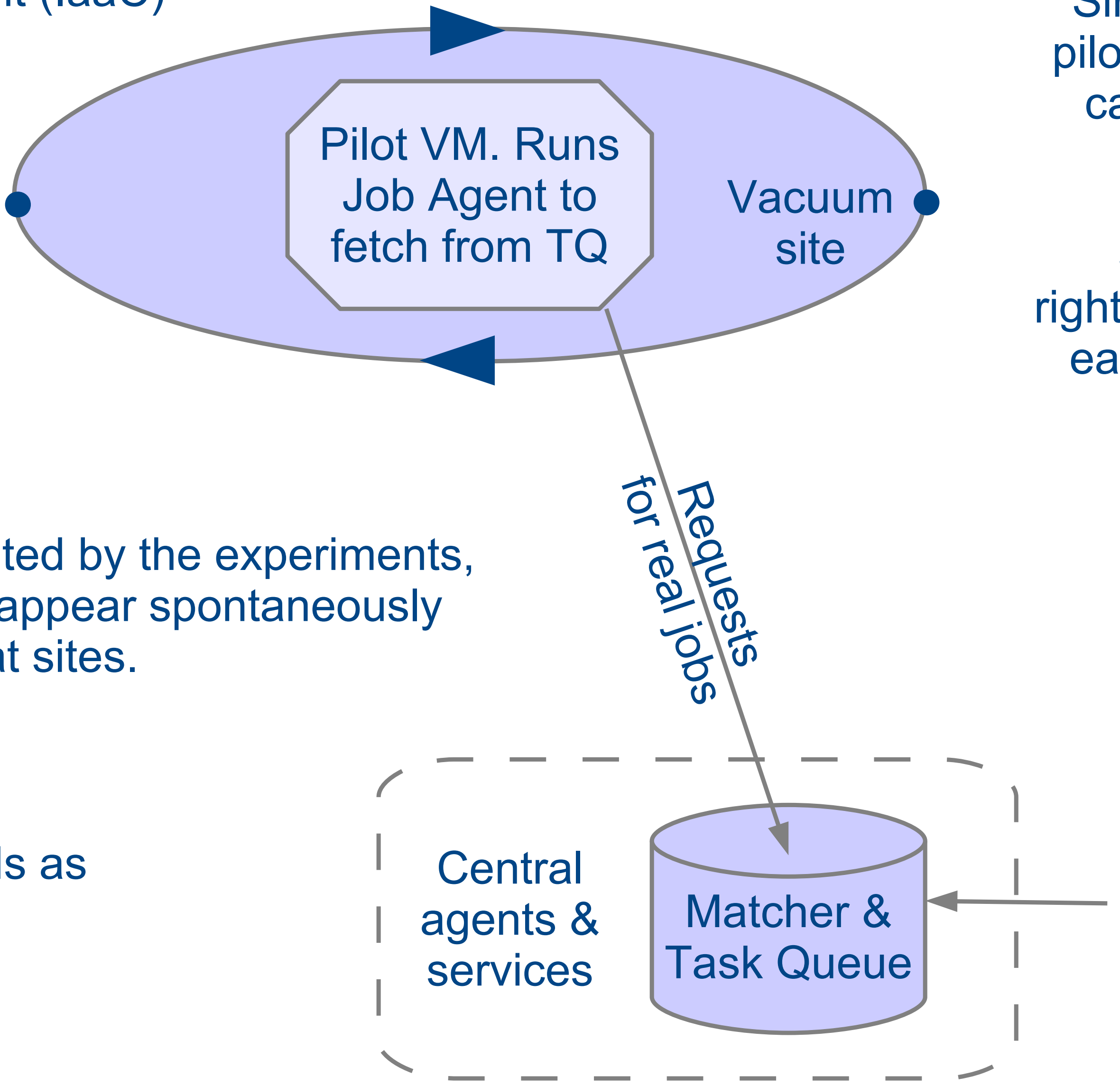
- Vac and Vcycle assume the VMs have a defined lifecycle
- Need a boot image and user_data file with contextualisation
  - Experiment provides procedure to make a site-wide user_data file
- Virtual disks and boot media defined and VM started
- machinefeatures and jobfeatures directories may be used by the VM to get wall time limits, number of CPUs etc
- The VM runs and its state is monitored
- VM executes shutdown -h when finished or if no more work available
  - Maybe also update a heartbeat file and so stalled or overruning VMs are killed
- Log files to /etc/machineoutputs which are saved
  - shutdown_message file can be used to say why the VM shut down

Vac's Vacuum Model

Infrastructure-as-a-Client (IaaC)

Since we have the pilot framework, we can do something much simpler.

Strip the system right down and have each physical host at the site create the VMs itself.



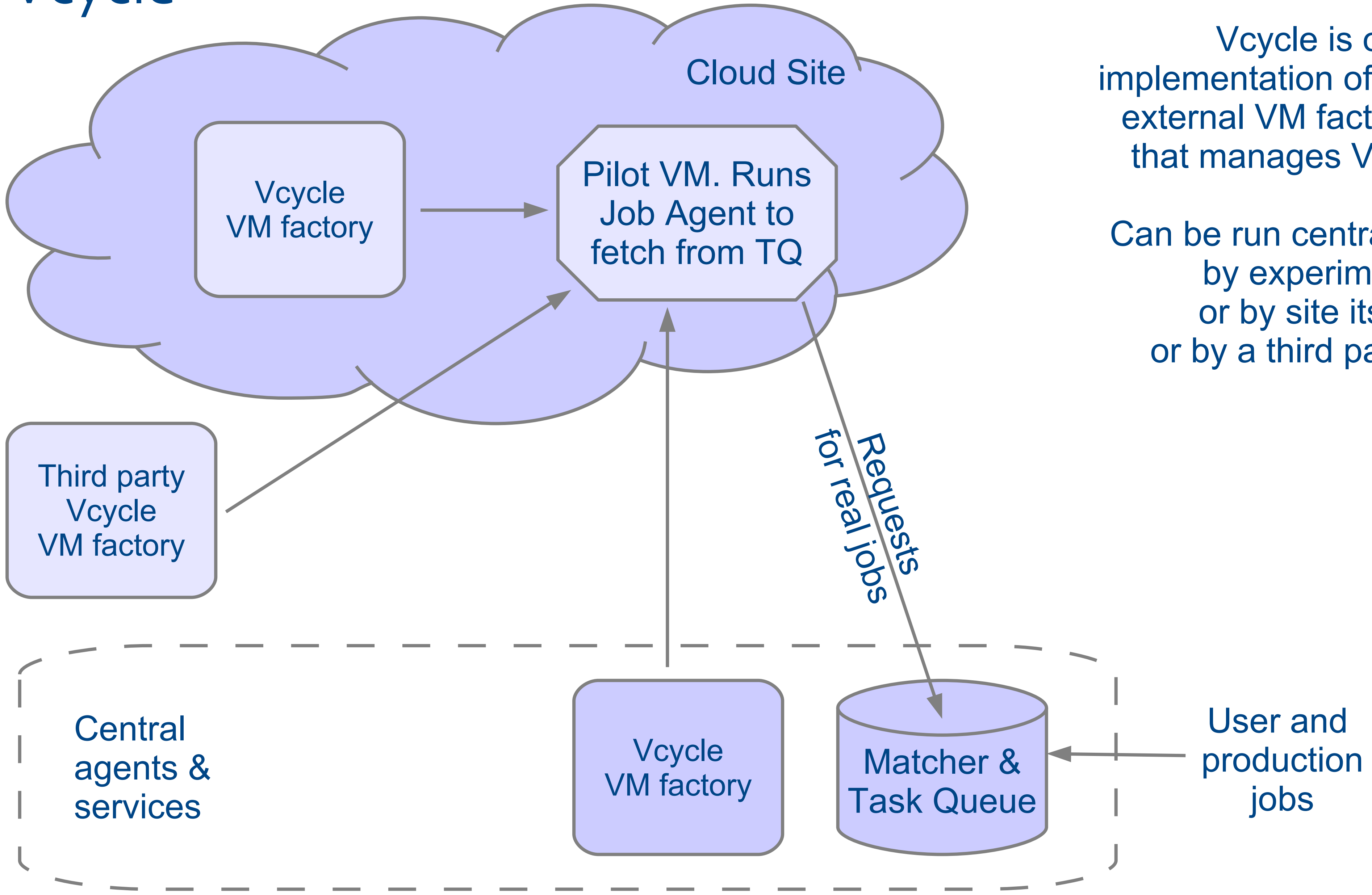
Instead of being created by the experiments, the virtual machines appear spontaneously "out of the vacuum" at sites.

Ideally use same VMs as with IaaS clouds

Vacuum model

- For the experiments, VMs appear by “spontaneous production in the vacuum”
  - Like virtual particles in the physical vacuum: they appear, potentially interact, and then disappear
- Following the CHEP 2013 paper:
  - *“The Vacuum model can be defined as a scenario in which virtual machines are created and contextualized for experiments by the resource provider. The contextualization procedures are supplied in advance by the experiments and launch clients within the virtual machines to obtain work from the experiments' central queue of tasks.”*
- At many sites, 90% of the work is done by 2 or 3 experiments
  - So a simple, reliable way of running their “baseload” of jobs is worthwhile
- cvmfs and pilots mean a small user_data file is all the site needs
  - Experiments can provide a script to create the site's user_data

Vcycle



Vcycle is one implementation of an external VM factory that manages VMs

Can be run centrally by experiment or by site itself or by a third party

Vcycle implementation

- Applies Vac ideas to OpenStack etc IaaS resources
- Experiment-neutral, and can be run by experiment or site or 3rd party
- Vcycle daemon creates VMs using user_data file
- Watches what they do
- Backs off if they are failing to stay running
  - No work? Fatal errors?
  - Can also use shutdown messages to make better decisions
- Provides machine/job features via HTTP
  - Also used to collect log files, shutdown messages, and heartbeat updates
- Currently uses OpenStack native nova API
  - OCCI fork has been done by WLCG team at CERN (Luis Villazon Esteban)
  - Reintegration and EC2 support in progress

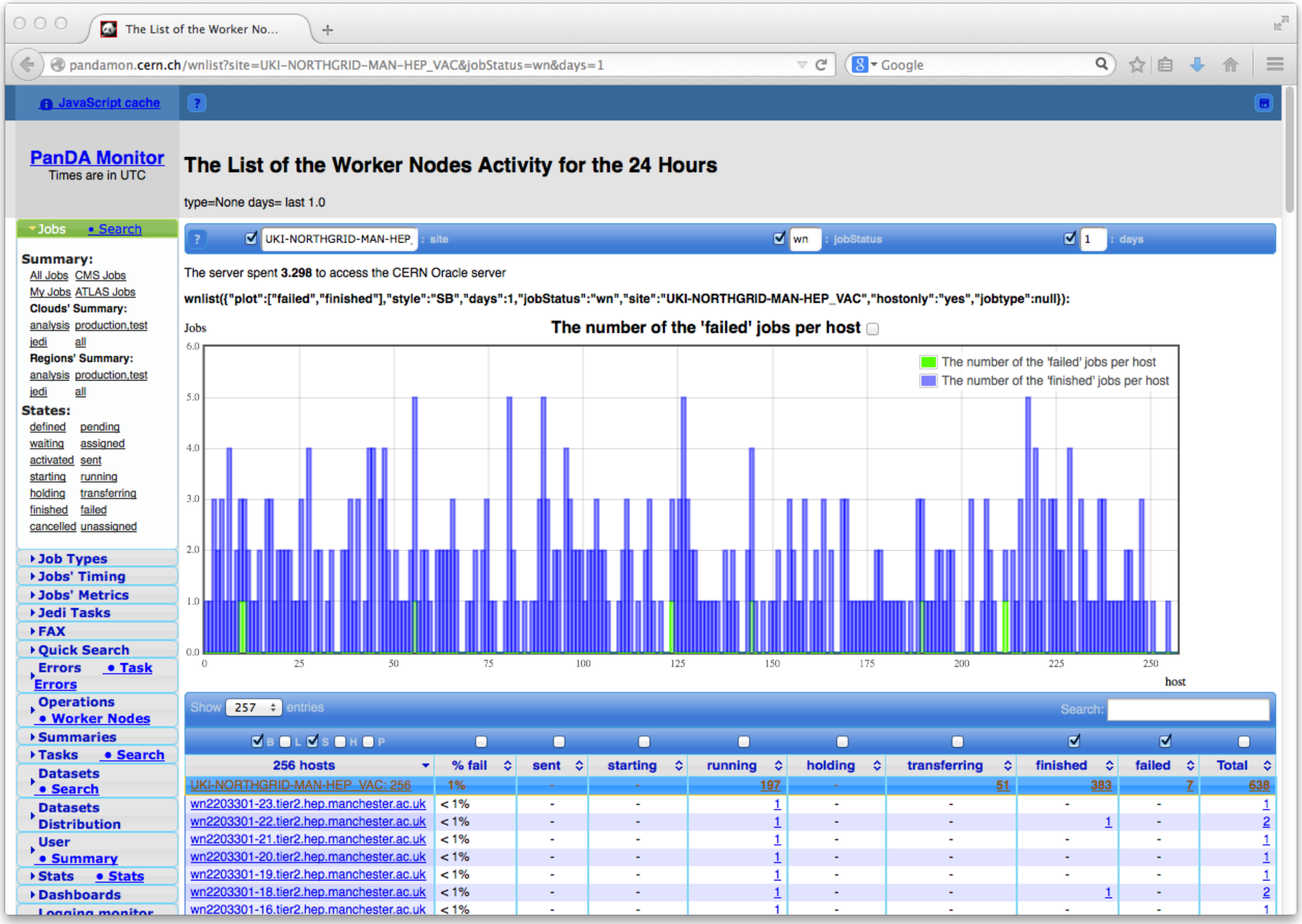
Machine/job features

- Proposed HEPiX protocol and current 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
- 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 use HTTP(S) web server on the Vcycle machine
- The basis for several scenarios for telling VMs and payload jobs what resources they have and how long they can run for
  - Want graceful termination of VMs to avoid disrupting payload jobs
  - /etc/machinefeatures/shutdowntime always set using max_wallclock_seconds

Vac and Vcycle at sites

- Manchester
  - Vac running ATLAS and LHCb VMs
  - Vcycle instance managing Imperial (LHCb/ATLAS) and CERN (ATLAS) VMs
- Lancaster
  - Vac running ATLAS and LHCb VMs
  - Vac being added to old farm now
- Oxford
  - Vac running ATLAS and LHCb VMs
- Imperial
  - GridPP OpenStack tenancy “gridpp-vcycle” has LHCb and ATLAS VMs
- CERN
  - LHCb tenancy managed by Vcycle on LHCb vobox at CERN
  - (A few) ATLAS VMs managed by Vcycle at Manchester

ATLAS Panda jobs running on Vac VMs



The List of the Worker No...

pandamon.cern.ch/wnlist?site=UKI-NORTHGRID-MAN-HEP_VAC&jobStatus=wn&days=1

JavaScript cache

PanDA Monitor
Times are in UTC

The List of the Worker Nodes Activity for the 24 Hours

type=None days= last 1.0

Jobs • Search

UKI-NORTHGRID-MAN-HEP : site — wn : jobStatus — 1 : days

Summary:
All Jobs CMS Jobs
My Jobs ATLAS Jobs
Clouds' Summary:
analysis production,test
jedi all
Regions' Summary:
analysis production,test
jedi all
States:
defined pending
waiting assigned
activated sent
starting running
holding transferring
finished failed
cancelled unassigned

Job Types
Jobs' Timing
Jobs' Metrics
Jedi Tasks
FAX
Quick Search
Errors • Task Errors
Operations • Worker Nodes
Summaries
Tasks • Search
Datasets • Search
Datasets Distribution
User • Summary
Stats • Stats
Dashboards
Logging monitor

The server spent **3.298** to access the CERN Oracle server

wnlist({"plot":["failed","finished"],"style":"SB","days":1,"jobStatus":"wn","site":"UKI-NORTHGRID-MAN-HEP_VAC","hostonly":"yes","jobtype":null}):

The number of the 'failed' jobs per host

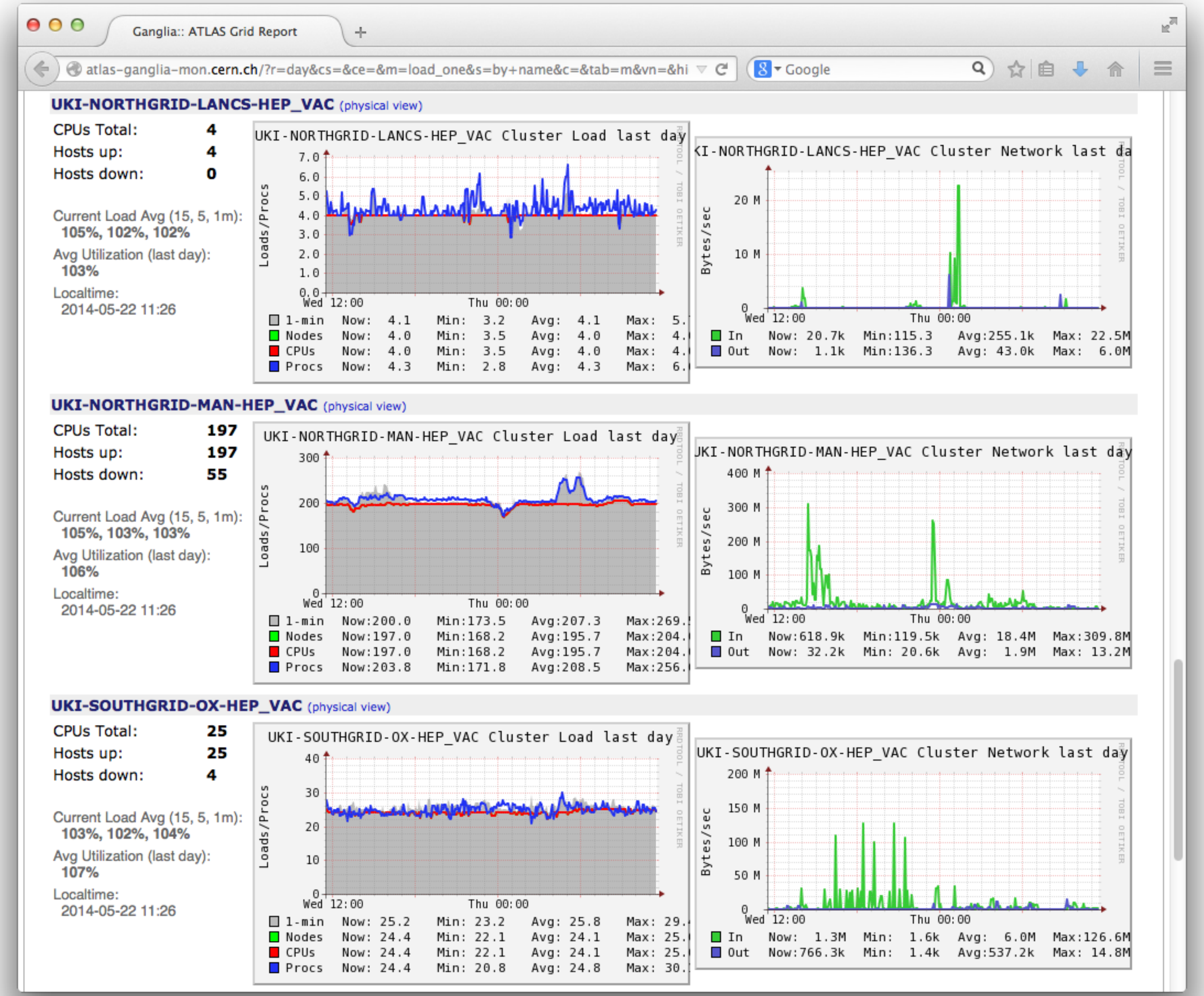
Jobs
6.0
5.0
4.0
3.0
2.0
1.0
0.0
0
25
50
75
100
125
150
175
200
225
250
host
🟩 The number of the 'failed' jobs per host
🟦 The number of the 'finished' jobs per host

Show 257 entries   Search:

✓ B ☐ L ✓ S ☐ H ☐ P

256 hosts	% fail	sent	starting	running	holding	transferring	finished	failed	Total
UKI-NORTHGRID-MAN-HEP_VAC: 256	1%	-	-	197	-	51	383	7	638
wn2203301-23.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	-	-	1
wn2203301-22.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	1	-	2
wn2203301-21.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	-	-	1
wn2203301-20.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	-	-	1
wn2203301-19.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	-	-	1
wn2203301-18.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	1	-	2
wn2203301-16.tier2.hep.manchester.ac.uk	< 1%	-	-	1	-	-	-	-	1

Ganglia monitoring of ATLAS Vac sites


Ganglia:: ATLAS Grid Report
atlas-ganglia-mon.cern.ch/?r=day&cs=&ce=&m=load_one&s=by+name&c=&tab=m&vn=&hi
Google

UKI-NORTHGRID-LANCS-HEP_VAC (physical view)

CPUs Total:	4
Hosts up:	4
Hosts down:	0

Current Load Avg (15, 5, 1m):
105%, 102%, 102%
Avg Utilization (last day):
103%
Localtime:
2014-05-22 11:26

```
UKI-NORTHGRID-LANCS-HEP_VAC Cluster Load last day
1-min   Now:  4.1    Min:  3.2    Avg:  4.1    Max:  5.
Nodes   Now:  4.0    Min:  3.5    Avg:  4.0    Max:  4.
CPUs    Now:  4.0    Min:  3.5    Avg:  4.0    Max:  4.
Procs   Now:  4.3    Min:  2.8    Avg:  4.3    Max:  6.
```

```
KI-NORTHGRID-LANCS-HEP_VAC Cluster Network last da
In    Now: 20.7k   Min:115.3   Avg:255.1k   Max: 22.5M
Out   Now:  1.1k   Min:136.3   Avg: 43.0k   Max:  6.0M
```

UKI-NORTHGRID-MAN-HEP_VAC (physical view)

CPUs Total:	197
Hosts up:	197
Hosts down:	55

Current Load Avg (15, 5, 1m):
105%, 103%, 103%
Avg Utilization (last day):
106%
Localtime:
2014-05-22 11:26

```
UKI-NORTHGRID-MAN-HEP_VAC Cluster Load last day
1-min   Now:200.0    Min:173.5    Avg:207.3    Max:269.
Nodes   Now:197.0    Min:168.2    Avg:195.7    Max:204.
CPUs    Now:197.0    Min:168.2    Avg:195.7    Max:204.
Procs   Now:203.8    Min:171.8    Avg:208.5    Max:256.
```

```
JKI-NORTHGRID-MAN-HEP_VAC Cluster Network last day
In    Now:618.9k   Min:119.5k   Avg: 18.4M   Max:309.8M
Out   Now: 32.2k   Min: 20.6k   Avg:  1.9M   Max: 13.2M
```

UKI-SOUTHGRID-OX-HEP_VAC (physical view)

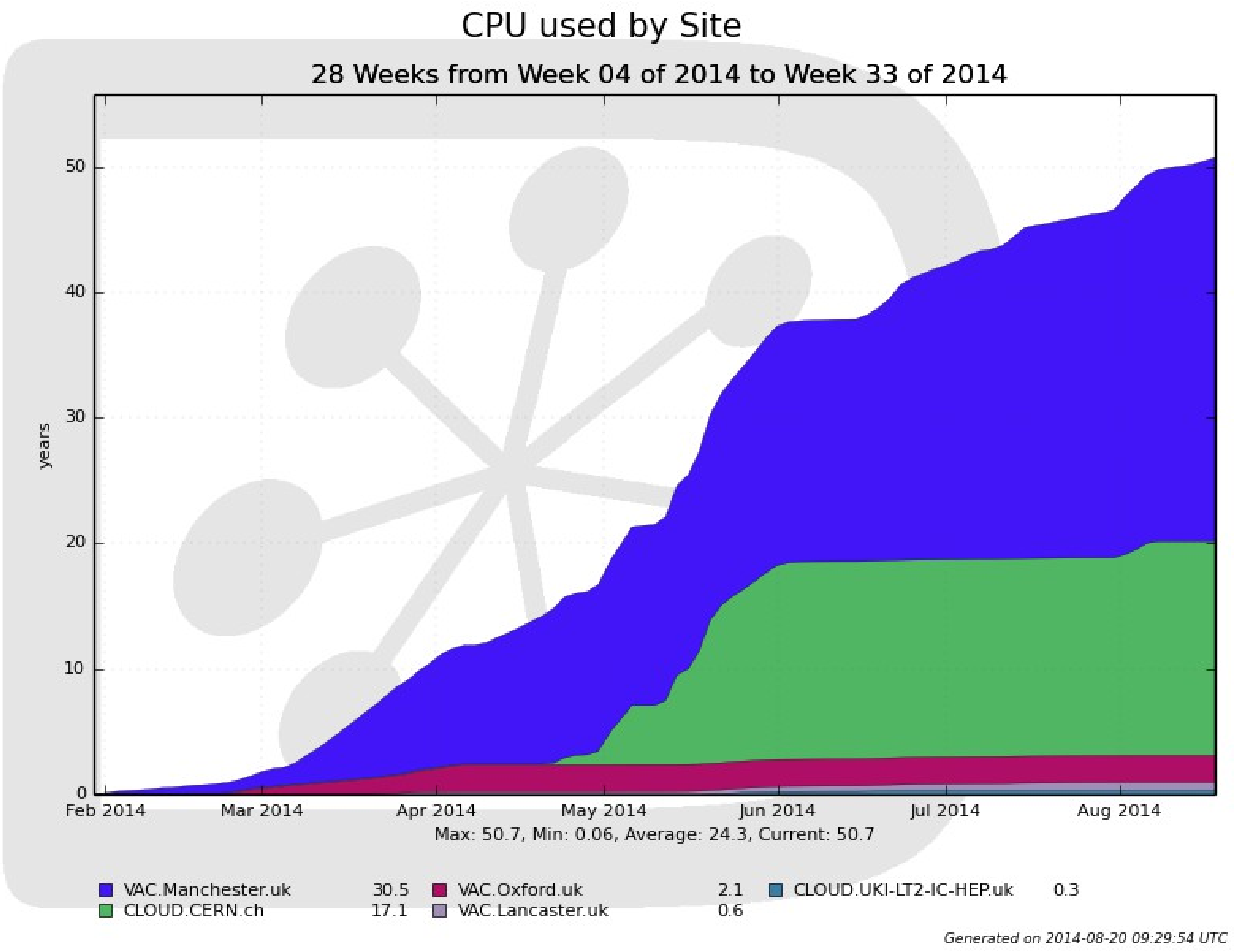
CPUs Total:	25
Hosts up:	25
Hosts down:	4

Current Load Avg (15, 5, 1m):
103%, 102%, 104%
Avg Utilization (last day):
107%
Localtime:
2014-05-22 11:26

```
UKI-SOUTHGRID-OX-HEP_VAC Cluster Load last day
1-min   Now: 25.2    Min: 23.2    Avg: 25.8    Max: 29.
Nodes   Now: 24.4    Min: 22.1    Avg: 24.1    Max: 25.
CPUs    Now: 24.4    Min: 22.1    Avg: 24.1    Max: 25.
Procs   Now: 24.4    Min: 20.8    Avg: 24.8    Max: 30.
```

```
UKI-SOUTHGRID-OX-HEP_VAC Cluster Network last day
In    Now:  1.3M   Min:  1.6k   Avg:  6.0M   Max:126.6M
Out   Now:766.3k   Min:  1.4k   Avg:537.2k   Max: 14.8M
```

LHCb CPU time from Vac and Vcycle VMs



Vcycle work at CERN

- Using Vcycle to run VMs in tenancies available through EGI
  - See the presentation on the WLCG use case.
- Thinking of the problem in terms of wanting to provide a “steady pressure” of job slots as long as there are payloads waiting to be run
- Vcycle is one way of doing this, as it monitors whether VMs find payloads to run or not
- Have produced a version of Vcycle that uses OCCI API rather than OpenStack's Nova API
- Using this for ATLAS, LHCb and looking at CMS. Would be interested in collaboration with ALICE too.
- Of interest as EGI aims to take operational responsibility for the cloud infrastructure and to take care of APEL accounting etc.

Summary

- Vac provides a simple way for sites to run VMs
- Vcycle provides a simple way to manage VMs on OpenStack
- Both demonstrated with ATLAS and LHCb production jobs
  - Measured VM efficiency for MC is very good now

- GridPP is in a very good position in this field
- They are able to run production work in VMs if/when the experiments or projects start pushing in that direction

- See http://www.gridpp.ac.uk/vac/ for RPMs, Yum repo, links to GitHub, docs, man pages etc