Vac and Vcycle status and plans

Andrew McNab
University of Manchester
GridPP, LHCb

Overview

- Vac and Vcycle
- Multipart vacuum pipes
- GOCDB capacity publishing
- OpenStack squid proxy management
- SAM/ETF tests
- Next steps

Vac vs Vcycle recap

- Two GridPP systems aimed at running VMs/Containers
- Vac - autonomous hypervisors
  - Each VM factory machine creates VMs or Docker containers in response to observed demand for each type of “logical machine”
  - Factory installation by Puppet etc or Vac-in-a-Box
- Vcycle - manages OpenStack, EC2, Google Cloud etc
  - VMs created via Cloud API in response to observed demand for each type of VM
  - Same VM definitions as Vac
- LMs are self-contained black boxes defined by experiments
  - Know how to pull in jobs to run from experiment HQ

Vcycle, OpenStack; Vac

- A lot of interest in targeting OpenStack from the new user communities in UKT0
- Many of the HTC workflows they have described could be run with VMs started on OpenStack by Vcycle
  - The VMs then contact the project's central services for work packages, jobs etc, as with LHC experiments
- If we can demonstrate this with VMs+OpenStack+Vcycle, then the same workflow should also work on Vac
  - VMs are the same
  - VMs are started "in the vacuum" by the resource providers rather than pushed into the sites by the project
- **Vac is clearly a lot simpler to operate than OpenStack!**

New since GridPP39 Lancaster

- Vac 3.0 release
  - CentOS 7 support
  - Multipart vacuum pipes
  - Capacity publishing to GOCDB
  - Docker container support
- Vcycle 1.0 release
  - CentOS 7, Vacuum pipe, GOCDB changes as with Vac
  - Squid proxy management
  - Local OpenStack flavor matching
    - “Give me a VM with 2 <= processors <= 8”

Multipart vacuum pipes

- Vacuum pipes introduced in Vac 2.0
  - Remote JSON file defining the “logical machines” the experiment wants you to run
  - URLs of user_data file for contextualisation and boot image
- In Vac 3.0 / Vcycle 1.0 they can now consist of multiple LM or container definitions
- Site defines overall target share for the experiment’s vacuum pipe
- Experiments can then define multiple VMs or containers, with target shares within the envelope chosen by the site
- This allows development VMs, single vs multiprocessor, etc

Multipart vacuum pipes

- Vac/Vcycle config for LHCb, ATLAS, **all** GridPP DIRAC Service VOs

```
[vacuum_pipe lhcb]
vacuum_pipe_url = https://lhcb-portal-dirac.cern.ch/pilot/lhcb.pipe
target_share = 4.0

[vacuum_pipe atlas]
vacuum_pipe_url = https://repo.gridpp.ac.uk/vacproject/atlas/atlas.pipe
target_share = 4.0

[vacuum_pipe gds]
vacuum_pipe_url = https://repo.gridpp.ac.uk/vacproject/gds/all-vos.pipe
target_share = 2.0
```

- That's it. 19 VOs from 9 lines of configuration.
- You also need to install the X.509 cert/key that the VMs or containers will use to authenticate to DIRAC, PanDA
  - But just one pair per vacuum pipe
  - Same pair used to authenticate the VMs to DIRAC for any VO

LHCb vacuum pipe


```
{
 "cache_seconds": 3600,
 "machinetypes":
 [
  {
   "suffix": "vm-prod",
   "target_share": 1,
   "accounting_fqan": "/lhcb/Role=NULL/Capability=NULL",
   "backoff_seconds": 600,
   "fizzle_seconds": 600,
   "heartbeat_file": "heartbeat",
   "heartbeat_seconds": 700,
   "machine_model": "cernvm3",
   "min_wallclock_seconds": 172800,
   "max_wallclock_seconds": 300000,
   "root_image": "https://lhcbproject.web.cern.ch/lhcbproject/Operations/VM/cernvm3.iso",
   "user_data": "https://lhcb-portal-dirac.cern.ch/pilot/user_data_vm_prod",
   "user_data_file_hostcert": "hostcert.pem",
   "user_data_file_hostkey": "hostkey.pem"
  },
  {
   "suffix": "squid",
   "target_share": 0,
   "accounting_fqan": "/lhcb/Role=NULL/Capability=NULL",
   "backoff_seconds": 600,
   "fizzle_seconds": 600,
   "heartbeat_file": "heartbeat",
   "heartbeat_seconds": 700,
   "machine_model": "vm-raw",
   "max_wallclock_seconds": 2592000,
   "user_data": "https://lhcb-portal-dirac.cern.ch/pilot/user_data_squid"
  }
 ]
}
```

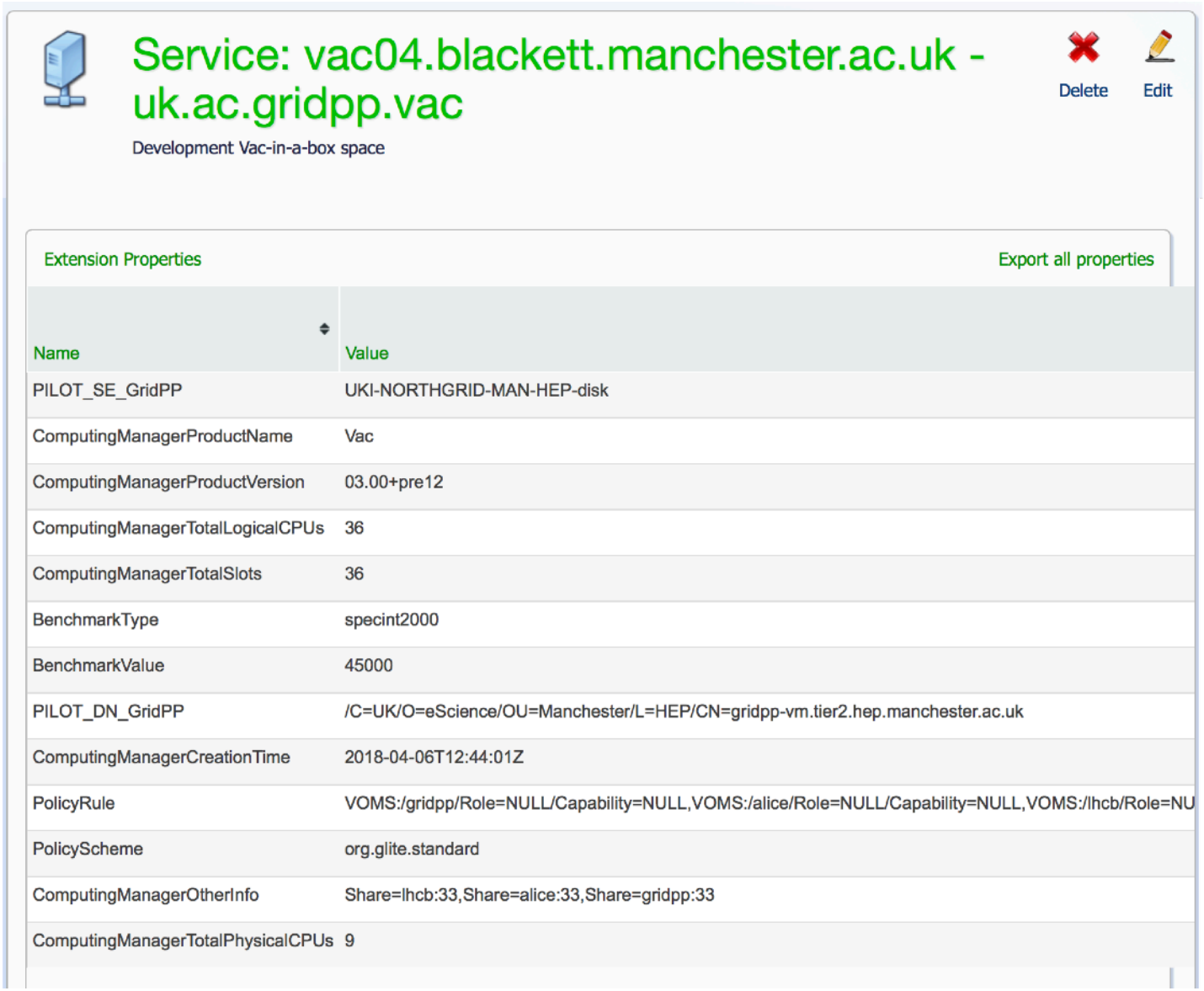
GOCDB capacity publishing

- WLCG Information Systems TF has agreed in outline to publish capacity and VO information to GOCDB
  - Part of removing the dependency on BDII
- Vac and Vcycle can now publish this automatically, based on their own knowledge
- Vcycle can just ask OpenStack for capacity
- Vac VM factories do a “census” of their neighbours every hour using the VacQuery UDP protocol
  - This builds up a record of what capacity that Vac Space (~CE) has, in terms of processors, HS06 etc
  - You can add a static file to represent machines that are down which you want to be included (more work, but also more control that way)

GOCDB capacity publishing (2)

- Vac/Vcycle then updates the GOCDB with this information
  - uk.ac.gridpp.vac and uk.ac.gridpp.vcycle service types, equivalent to CEs
  - Each service can have extension properties in GOCDB
  - Use Glue2 inspired names for properties, with the capacity and VO info
  - In GOCDB you can authorize X.509 DNs to be able to do these updates
- We still need to get this information consumed downstream (via CRIC?) so it goes into REBUS
- Since the VOs are also listed, this information could also be used by VOs to discover where they can run

GOCDB capacity publishing



Service: vac04.blackett.manchester.ac.uk - uk.ac.gridpp.vac

Development Vac-in-a-box space

Delete Edit

Extension Properties — Export all properties

Name	Value
PILOT_SE_GridPP	UKI-NORTHGRID-MAN-HEP-disk
ComputingManagerProductName	Vac
ComputingManagerProductVersion	03.00+pre12
ComputingManagerTotalLogicalCPUs	36
ComputingManagerTotalSlots	36
BenchmarkType	specint2000
BenchmarkValue	45000
PILOT_DN_GridPP	/C=UK/O=eScience/OU=Manchester/L=HEP/CN=gridpp-vm.tier2.hep.manchester.ac.uk
ComputingManagerCreationTime	2018-04-06T12:44:01Z
PolicyRule	VOMS:/gridpp/Role=NULL/Capability=NULL,VOMS:/alice/Role=NULL/Capability=NULL,VOMS:/lhcb/Role=NU
PolicyScheme	org.glite.standard
ComputingManagerOtherInfo	Share=lhcb:33,Share=alice:33,Share=gridpp:33
ComputingManagerTotalPhysicalCPUs	9

Squid proxy management in Vcycle

- To use cvmfs we need suitable Squid proxy caches, ideally at the same site as the VMs
- At sites without these, LHCb has been manually setting up Squid VMs on OpenStack: “squids as pets”
- But Vcycle is good at managing VMs: “squids as cattle!”
- In 1.0, you can use cvmfs_proxy_machinetype to specify which machinetype defines the pool of squid VMs
- All squid VMs with a recent heartbeat are included
  - A user_data file is provided which only generates heartbeats when the Squid daemon in that VM is working
- The worker VMs are provided with a list of squid caches, in the format the cvmfs configuration needs

SAM/ETF for Vac/Vcycle VMs

- We've ticked off APEL, GOCDB and now capacity publishing
- What's left is the SAM testing (now "ETF" in WLCG)
- LHCb has been running the ETF probes in its VMs since 2017
  - Code added to DIRAC pilots to run the probes before payloads
  - Results go all the way through the chain to the dashboards and WLCG monthly A&R reports
- Same approach could be done for ATLAS and ALICE
  - But probably more practical to have ETF jobs submitted to the HTCondor pools that the VMs are already pulling pilot jobs from
  - Discussions started with ETF team on this

Summary and next steps

- Vac 3.0 and Vcycle 1.0 released
  - Multipart vacuum pipes
  - GOCDB capacity publishing
  - CentOS 7 support
- Vcycle squid proxy management
- Next steps
  - ETF tests for ATLAS and ALICE VMs
  - IPv6 support in Vac …
  - … and the VM definitions