

Vac capacity publishing to GOCDB

Vac accounting and APEL

Andrew McNab University of Manchester LHCb and GridPP

Two talks in one set of slides

Start with:

Vac publishing to GOCDB

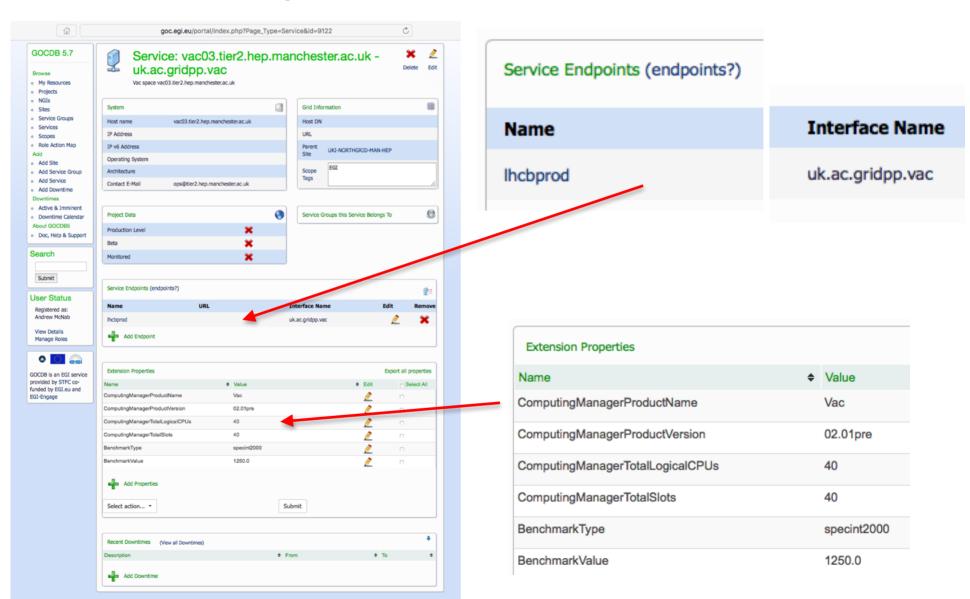
Vac and Vcycle

- Vac: worker nodes as autonomous VM-factories
- Vcycle: manages VMs on OpenStack etc
- Both use the Vacuum Model, and same VM definitions
- VMs are started up and pull work in:
 - DIRAC payloads, HTCondor jobs etc
- In terms of publishing, they both have a view of the whole "space" of VMs
 - Vcycle from cloud API VM listing
 - Vac using VacQuery UDP protocol to poll other VM factories in the same space

Publishing to GOCDB

- Preliminary tests done with curl + shell script
 - Used vac03.tier2.hep.manchester.ac.uk service
 - Used a DN added for API access
- Able to use the new API to insert Extension Properties for the service itself
- Able to add Endpoints within the service by simulating submitting an HTML form
- Able to add Extension Properties to Endpoints using the new API
- Not able to add or edit services using API DN

vac03.tier2.hep.manchester in GOCDB



Storing GLUE2 in GOCDB extensions

- Properties of the vac03 service
- Concatenated names ("full path")

Extension Properties		
Name	\$	Value
ComputingManagerProductName		Vac
ComputingManagerProductVersion		02.01pre
ComputingManagerTotalLogicalCPUs		40
ComputingManagerTotalSlots		40
BenchmarkType		specint2000
BenchmarkValue		1250.0

Storing GLUE2 in GOCDB extensions (2)

Properties of the lhcbprod endpoint

Extension Properties	
Name	♦ Value
ComputingShareMaxWallTime	86400
ExecutionEnvironmentOSFamily	Linux
ExecutionEnvironmentOSName	CERN Virtual Machine
ExecutionEnvironmentPlatform	x86_64
ExecutionEnvironmentVirtualMachine	True
MappingPolicyRule	VO:lhcb,VOMS:/lhcb/Role=NULL/Capability=NULL
MappingPolicyScheme	org.glite.standard

Next steps in publishing to GOCDB

- These are sufficient for LHCb discovery of sites and importing service parameters into DIRAC CS
- Can readily be extended for other requirements
- Plan is to automate this publishing within Vac and Vcycle, using each system's knowledge of capacity, HS06 values etc
 - Started coding this in common vacutils library
- Should be straightforward for CE/Batch systems too
- Interim solution would be general tool to read Resource LDAP server on CE and republish to GOCDB
 - Needs to be run by site due to permissions model

Vac accounting and APEL

Vac and Vcycle

- Vac: worker nodes as autonomous VM-factories
- Vcycle: manages VMs on OpenStack etc
- Both use the Vacuum Model, and same VM definitions
- VMs are started up and pull work in:
 - DIRAC payloads, HTCondor jobs etc
- Same accounting model and library used for both
 - So what I say for Vac applies to Vcycle
 - But Vcycle isn't managing any resources which need APEL accounting yet, so APEL support not used in production

Vac and APEL

- Vac has had native APEL support since 2015
 - Before that wrote out PBS/BLAHP format log files that could be parsed and fed into site APEL service
- When a VM finishes, Vac writes out an APEL accounting record file in /var/lib/vac/apel-outgoing and /var/lib/ vac/apel-archive
- APEL's ssmsend client runs every ~hour to process record files in apel-outgoing
 - When file is successfully received by message bus, then ssmsend deletes it
 - ssmsend is also run when Vac shuts down to flush any records still waiting

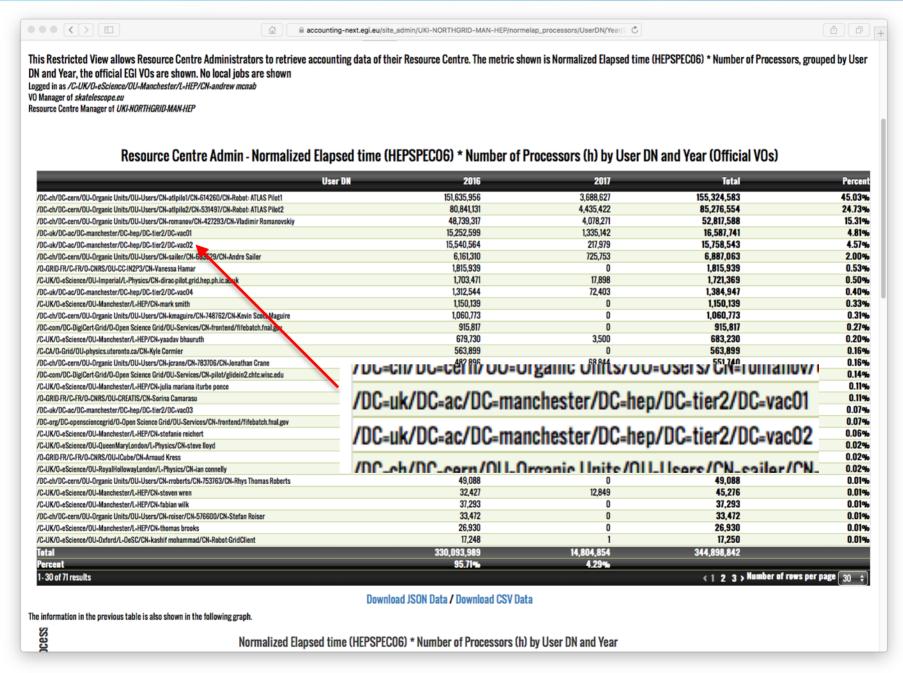
APEL record file

```
APEL-individual-iob-message: v0.3
Site: UKT-NORTHGRTD-MAN-HFP
SubmitHost:
  vac01.tier2.hep.manchester.ac.uk/vac-wn1905190.tier2.hep.manchester.ac.uk
LocalJobId: 3ec25221-4c8f-4d91-a5f2-df6966b7870f
LocalUserId: wn1905190.tier2.hep.manchester.ac.uk
Oueue: atlasprod
GlobalUserName: /DC=uk/DC=ac/DC=manchester/DC=hep/DC=tier2/DC=vac01
FQAN: /atlas/Role=NULL/Capability=NULL
WallDuration: 9953
CpuDuration: 9368
Processors: 1
NodeCount: 1
InfrastructureDescription: APEL-VAC
InfrastructureType: grid
StartTime: 1484691667
EndTime: 1484701620
MemoryReal: 2621440
MemoryVirtual: 2621440
ServiceLevelType: HEPSPEC
ServiceLevel: 8.8
%%
```

Mapping VM attributes to Job records

- Some attributes are what you would expect
 - Sitename and FQAN
- GlobalUsername represents the space too
 - /DC=uk/DC=ac/DC=manchester/DC=hep/DC=tier2/DC=vac01
 - Means Vac-specific totals can be seen in the portalSubmitHost is specific to the VM factory
 - vac01.tier2.hep.manchester.ac.uk/vac-wn1905190.tier2.hep.manchester.ac.uk
 - vac01 is the Vac space (equivalent to CE name)
 - wn1905190 is the VM factory
 - Used by APEL-Sync records: means Sync records can be generated on each VM-factory independently





Grid vs Cloud accounting

- Vac tries to look like "just another CE" in GOCDB etc
 - Natural to report it through "Grid" route in APEL
- Vcycle also looks like a CE, but is managing cloud resources.
- This approach with APEL allows Cloud accounting through the normal "Grid" route in APEL too
 - In particular, it requires no reconfiguration or modification of the underlying cloud infrastructure
- Compare EGI Federated Cloud approach
 - Needs separate route into APEL
 - Needs reconfiguration by cloud provider (eg elevated privileges for account reading native accounting DB)