Info publishing to GOCDB for Vac and Vcycle

Andrew McNab
University of Manchester
GridPP, LHCb

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”
- 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
- VMs are self-contained black boxes defined by experiments
  - Know how to pull in jobs to run from experiment HQ

Previous talks at InfoSys TF

- "Publishing Vcycle and Vac state in GLUE2/JSON/HTTPS" 12 Nov 2015
  - Using GLUE2 JSON schema, with file put on a web server with a pointer from GOCDB

- Then the TF "weather" changed …
- "Vac capacity publishing to GOCDB" 2 Feb 2017
  - Proof of concept using new GOCDB write API to put info into GOCBD itself
  - Using GLUE2 inspired property names

GOCDB write API

- Since last year GOCDB has provided an API for writing information programmatically
- Sites can specify X.509 DNs able to write
- Only allows creating/updating property extensions to services and to endpoints of services
  - Can’t currently add endpoints programmatically but this is coming too
- Services are things like CEs
  - Or Vac/Vcycle “spaces”, like an OpenStack project
- Endpoints could be things like queues
  - Or Vac/Vcycle machinetypes (a type of VM defined by one experiment)

Vac/Vcycle implementation

- Vac 3.0 (uk.ac.gridpp.vac) and Vcycle 1.0 (uk.ac.gridpp.vcycle) can now publish to GOCDB automatically, based on their own knowledge
  - GLUE2 inspired names, with capacity and VO information
  - Only service extension properties are created for now
- 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

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

Delete Edit

Development Vac-in-a-box space

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

Next steps

- 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
  - What's published is enough for LHCb
- Or do we go back to publishing JSON with a pointer from GOCDB?