Vcycle Digital Assets for IRIS

Andrew McNab

University of Manchester

Vcycle recap

- Vcycle is a small (few thousand lines of Python) agent which instantiates VMs on OpenStack, EC2, Google Cloud
- Projects publish VM definitions as JSON files
  - Max lifetime, memory, URL of disk image etc
- Vcycle attempts to create VMs for projects and then watches whether they find work to do
- Uses feedback to decide whether to create more VMs of a particular, or to back off for a while and try again later
- Need suitable VM definitions for the projects
- VM definitions exist for ALICE, ATLAS, LHCb, capital-I DIRAC, and generic one for HTCondor

Vcycle digital assets project

- Three work packages
  - WP1 Library of VM, container and bare metal definitions
  - WP2 Vcycle hardening and scalability
  - WP3 Vcycle Dashboard (starts in FY19)

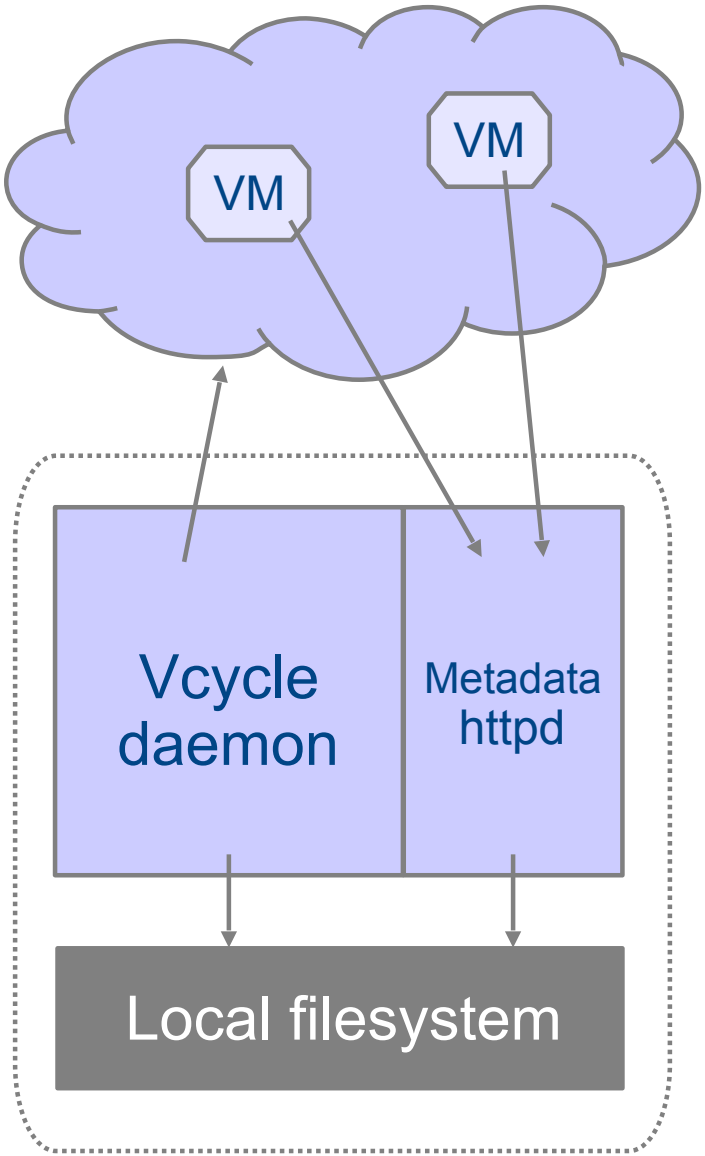
WP1 Library of VM definitions

- 1.1 VM definitions work started in FY18
  - for EUCLID (started, ready to test with a EUCLID slurm headnode)
  - SKA/AENEAS on capital-I DIRAC (multiprocessor job support mostly done, will also help LSST)
  - DUNE (procedure agreed last month, following F2F meetings at WLCG19, VM with required software produced, integration next)
  - Bonus: HTCondor VMs to provide batch-on-cloud (started)
- 1.2 Support for IRIS AAI in Vcycle - needs clarification of what to target
- 1.3 Docker container and bare metal support - not yet started. It involves adding support to Vcycle for these logical machine types in OpenStack. No use cases identified. Should be straightforward once we have a user community to try this with.
- Tasks 1.4 and 1.5 start in FY19 and expand on the above with more user communities and generalising the solutions to provide a library of off-the-shelf VM/container definitions.

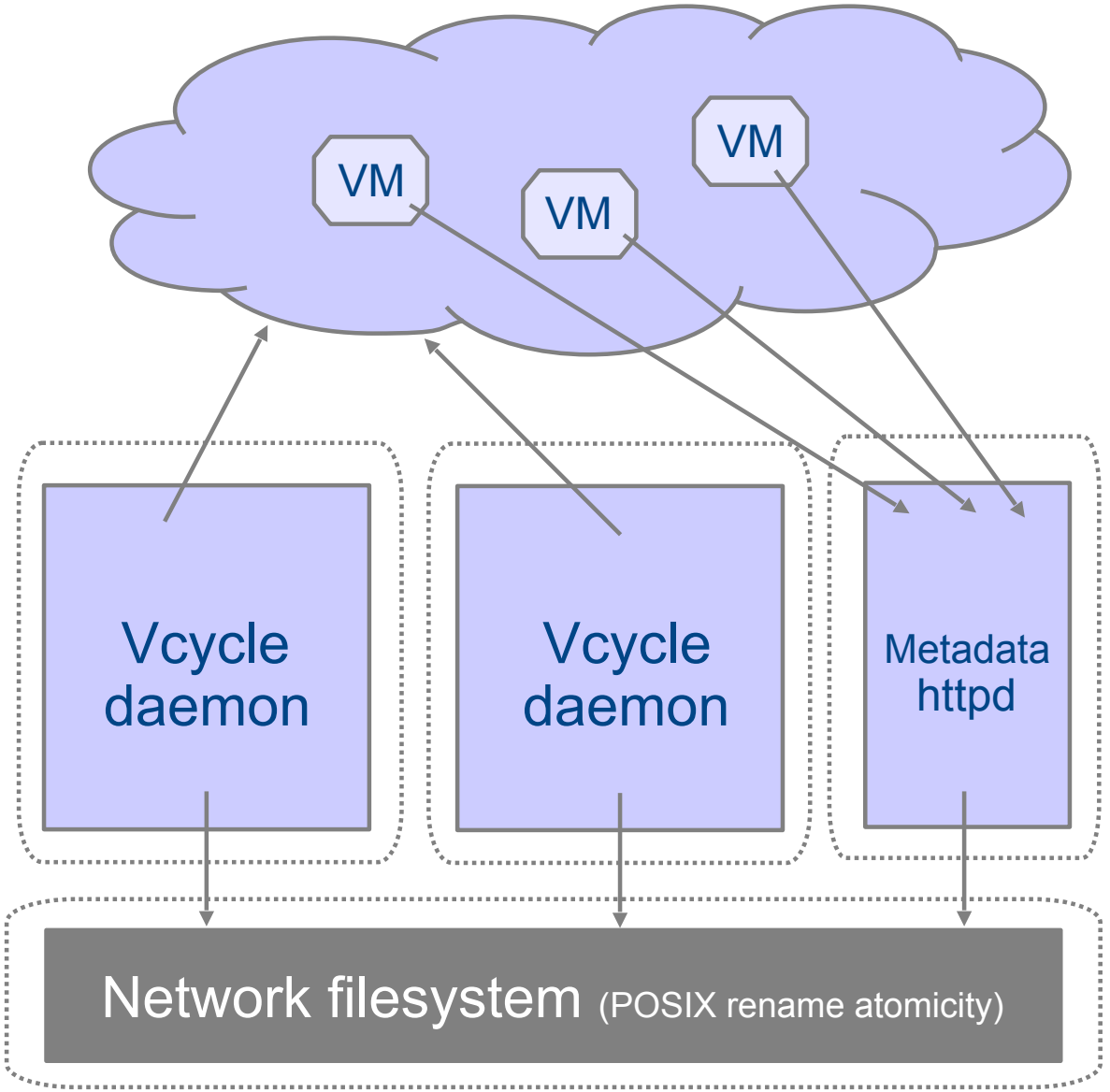
WP2 Vcycle hardening and scalability

- 2.1 Hardening and documentation. Ongoing. Work done so far already in production.
- 2.2 Support for multiple co-operating Vcycle instances
  - Scalability: going beyond ~4000 VMs. V1 done.
  - Live upgrades: take one Vcycle instance down at a time for upgrades etc. V1 done
  - Failover: one Vcycle instance dies, can the VMs it created be managed by another instance? Not started yet.
  - Ongoing work to fix bugs, race conditions, as this scales up.
  - So far using shared filesystem as backend: will investigate DB
- 2.4, 2.5 and 2.6 in FY19 to support multiple co-operating VMs (eg Apache Spark); pre-emption of VMs by higher priority workloads; and further work based on experiences in FY18/19

Single instance Vcycle



Multi instance Vcycle



Summary and next steps

- VM/Container library work is underway on multiple fronts
  - Some just starting (DUNE VMs)
  - Others almost ready for release (multiprocessor DIRAC VMs)
- Initial Vcycle scalability work is already in production
  - Two of the three multi-instance scenarios accommodated
- Next steps:
  - Complete existing VMs
  - Start the generalised library of VM definitions work
  - Vcycle Dashboard (WP3) and failover support
  - Support the IRIS AAI solution once agreed