



# Vcycle and Vac update

**Andrew McNab**

University of Manchester

GridPP, DUNE, LHCb



# Vcycle and Vac

- Vac update
- Digital Asset for IRIS
  - Vcycle
  - Library of VMs



# Vac update

- Vac development is essentially “done”
  - Objective of running complete sites without the usual grid middleware stack has been achieved
    - Vac-only lightweight sites are up and running
    - Including Birmingham with storage
- Vac shares the same “vacuum platform” API for the VMs as Vcycle
- Most of the changes to Vac since 3.0 have been tracking additions to the API for Vcycle
- Some work may be done on Vac as a way of “running OpenStack VMs without OpenStack”



# Vcycle recap

- Vcycle is a small (few thousand lines of Python) agent which instantiates VMs on OpenStack, EC2, Google Cloud, ...
- Experiments 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 experiments
  - Same VM definitions for Vac and Vcycle
- VM definitions exist for ALICE, ATLAS, DUNE, LHCb, GridPP DIRAC (SKA, LSST, LZ, ...), VMCondor simple batch

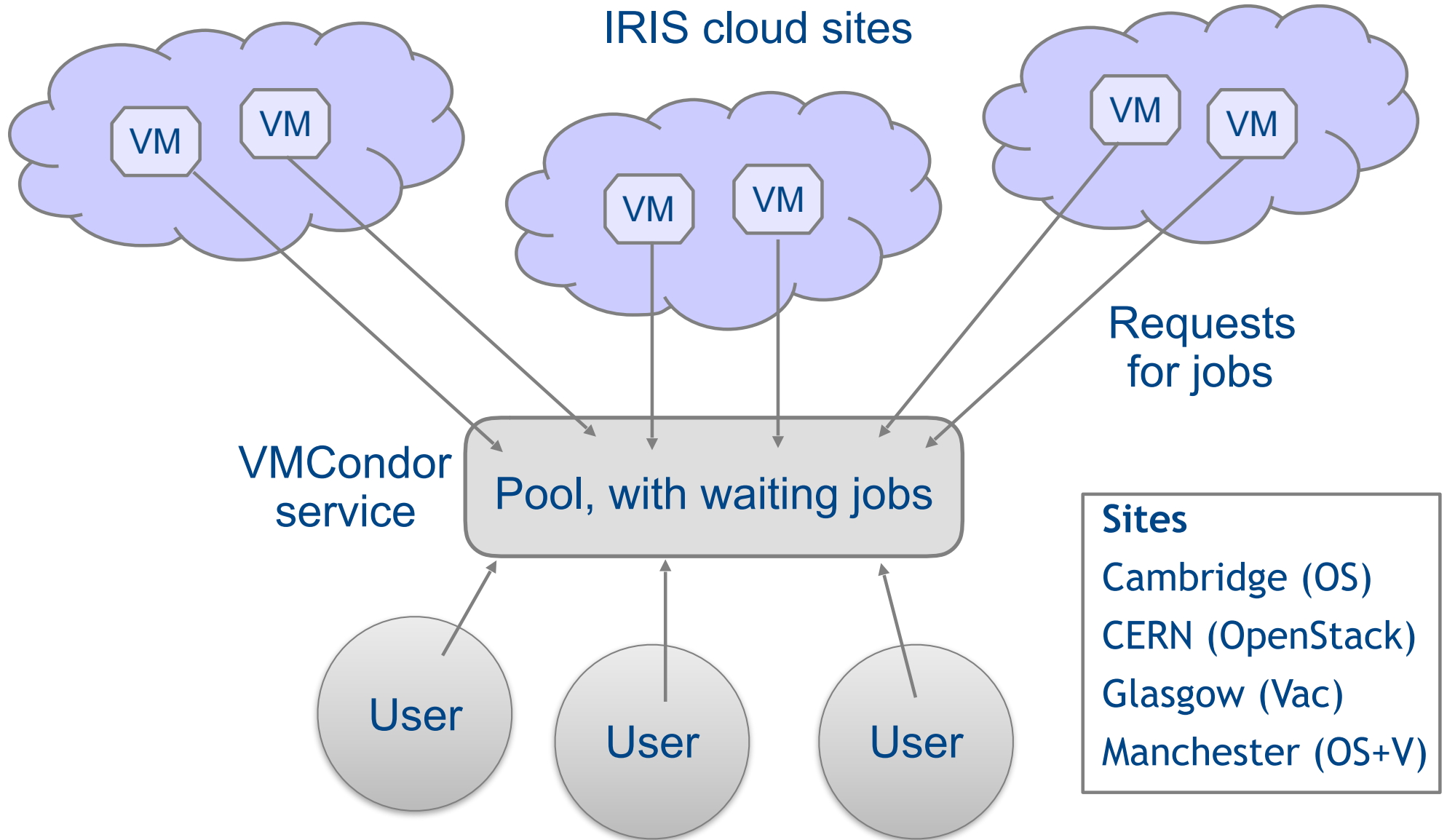
# IRIS context

- IRIS has funded OpenStack in several ways:
  - StackHPC has created “Scientific OpenStack”
    - OpenStack components in Docker containers
    - Make it considerably easier to install
    - Consistency across IRIS sites => peer support
    - It may now be feasible for larger sites to install OpenStack without a huge commitment of effort
  - Vcycle Digital Asset with 3 work packages
  - Hardware explicitly for OpenStack services at Cambridge, Imperial and Manchester
  - Other IRIS sites may provide some resources with an OpenStack service

# WP1 Library of VM definitions

- CernVM 4 (~CentOS7) updates for existing VMs (GridPP, LHCb, ATLAS; ALICE next)
- SKA and LSST requirements in GridPP DIRAC VMs (mainly multiprocessor job support)
- DUNE VM created, contacts GlideinWMS pool at FNAL for jobs
- EUCLID VM ongoing: requires that VMs join a slurm batch system in Edinburgh over a VPN
- VMCondor now an end-to-end “simple batch” system
  - Users do not need X.509 certificates
  - VMCondor wrappers of HTCondor commands
  - VMs at multiple sites join an HTCondor pool
  - <https://github.com/iris-ac-uk/iris-vmcondor/wiki>

# VMCondor architecture

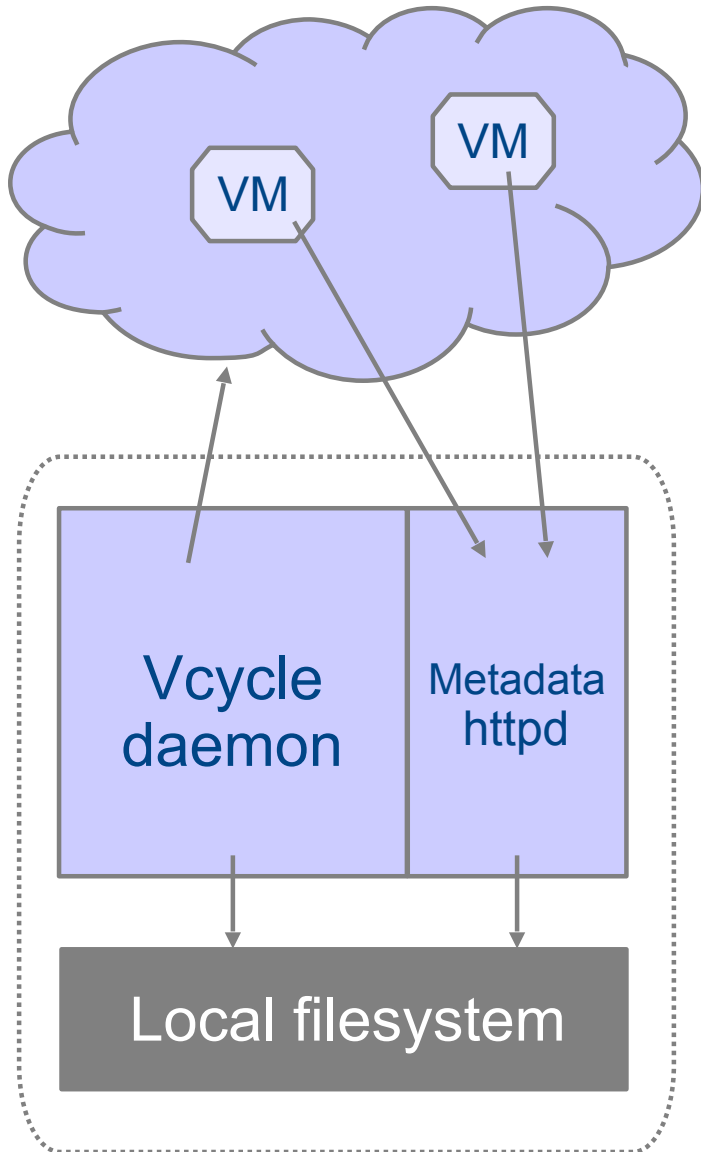


## WP2 Vcycle hardening and scalability

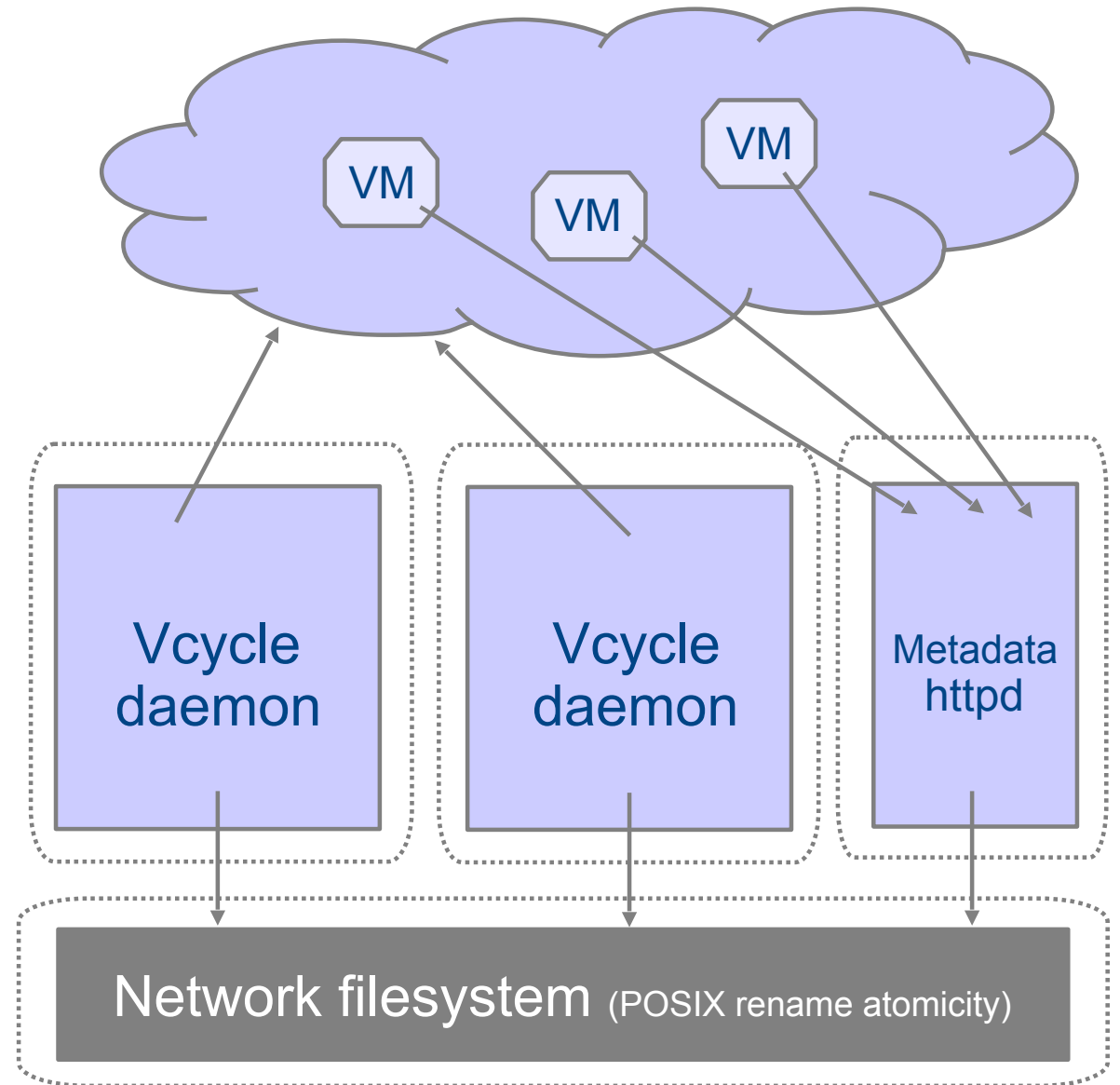
- Hardening and documentation
  - LHCb Vcycle instance at CERN used to identify some race conditions and improve robustness
- Support added for multiple co-operating Vcycle instances
  - Scalability: going beyond ~4000 VMs.
  - Live upgrades: take one Vcycle instance down at a time for upgrades etc.
  - Failover: one Vcycle instance dies, the VMs it created become managed by another instance
  - Ongoing work to fix bugs, race conditions, as all this scales up.
- Underpinned by shared filesystem between instances rather than a database



# Single instance Vcycle



# Multi instance Vcycle





# Summary and next steps

- Vac is in a stable state, just with maintenance and shadowing Vcycle changes to the VM API
- VM library is producing results
  - Multiprocessor DIRAC VM (GridPP for SKA/LSST and LHCb)
  - New DUNE VM
  - New VMCondor simple batch VM
- Multi instance Vcycle work is in production
  - Scalability, live upgrades, failover
- Next steps:
  - More VM types
  - Provide these VM-based workloads to more IRIS sites