An implementation of cloud-based grid CE and SE for ATLAS and Belle II

Jonathan Woithe [1] Martin Sevior [2] Paul Jackson [1] David Dossett [2]
Marcus Ebert [3]

[1]University of Adelaide, Australia
[2]University of Melbourne, Australia
[3]University of Victoria, Canada

CHEP2024, Kraków, Poland
October 2024

Outline

Motivation

- Research Computing groups at our institutions now provide and maintain hardware and access through cloud platforms

  ➜ Need to fit in with what our universities provide

- Industry standard interfaces

  ➜ No esoteric filesystems are exposed to cloud users

- Exploit economies of scale in commerical cloud resource providers for Grid computing.

- Compute and storage can be easily increased as funding allows and demand grows.

Servers (storage, compute)

- Melbourne Research Cloud (MRC) VMs
- Orchestration by OpenStack
- Server configuration managed by Ansible, tracked in git

Storage

- 750 TB of S3 compatible object store from MRC
  - Not a traditional filesystem
  - Each “file” is an object in a database
  - The object’s “key” is interpreted as its filesystem path
  - No explicit objects for filesystem directories

Storage

- Currently use a single bucket for flexibility
- Belle II and ATLAS have separate key namespaces
  - Gives illusion of separate top-level directories
- Transports: root, davs, https
- Enabled by the `xrootd-s3` work at SLAC (`https://cds.cern.ch/record/2857626/files/ATL-SOFT-SLIDE-2023-125.pdf`)

Storage

- Xrootd redirector VM
  - Authenticates incoming requests
  - Generates access token
  - Redirects requests to one or more proxy servers
- Xrootd proxy server VM
  - Validates access token
  - Serves requested resource
  - Currently have 1 proxy server
  - Can deploy more when bandwidth requirements increase

Storage

- Storage Resource Reporting (SRR) json file
  - Defines Belle II and ATLAS storage shares, space usage and capacities
  - Generated hourly by python script on primary proxy server
  - Boto3 library used for S3 access
- Adler32 checksums
  - Managed on primary xrootd proxy server
  - Maintained with python script using boto3 library for S3 access
  - Calculated on first request, stored as metadata attribute on S3 object

Storage

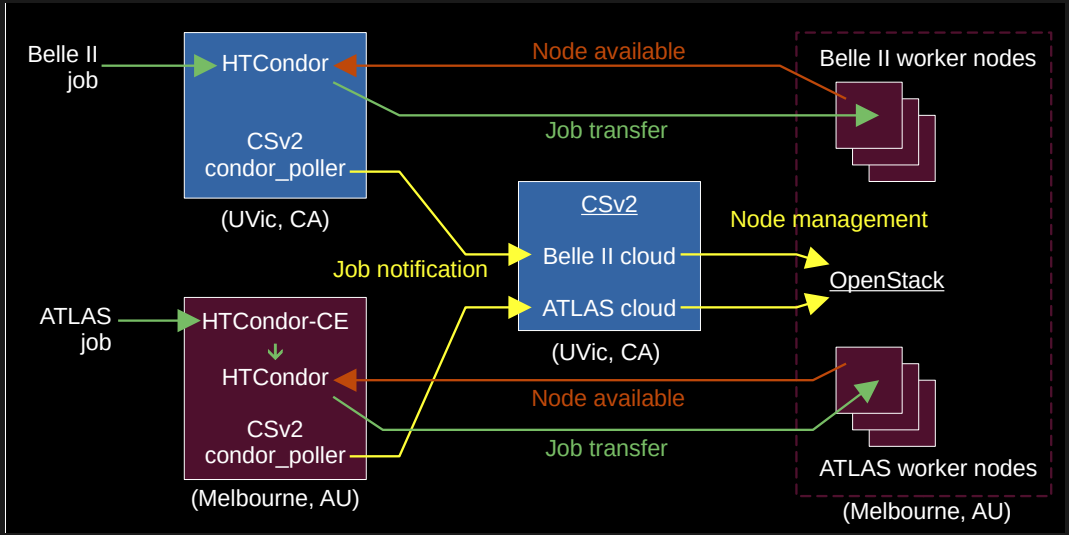
- Third Party Copy
  - Executed on xrootd proxy servers
  - Uses bash shell script to interface with xrootd
  - `root://` transfers: xrdcp streams content from source, s3cmd sends content into S3 object
  - `davs://` transfers: handled by libXrdHttpTPC.so:

    ```
    http.exthandler xrdtpc libXrdHttpTPC.so
    ```

Compute

- Slightly different architectures used for Belle II and ATLAS
- Cloud resources managed by Cloud Scheduler v2 (CSv2) instance at UVic (`https://csv2.heprc.uvic.ca`)

Compute


Belle II job
HTCondor
CSv2 condor_poller
(UVic, CA)
Node available
Job transfer
Belle II worker nodes
CSv2
Belle II cloud
ATLAS cloud
(UVic, CA)
Node management
OpenStack
Job notification
ATLAS job
HTCondor-CE
HTCondor
CSv2 condor_poller
(Melbourne, AU)
Node available
Job transfer
ATLAS worker nodes
(Melbourne, AU)

Compute

Belle II

- HTCondor host VM at UVic
- Jobs submitted to HTCcondor host via local DIRAC site-director
  (Belle-II still uses GSI, no HTCondor-CE is involved)
- CSv2 monitors HTCondor, starts HTCondor worker VM in MRC if needed
- Worker node set up via cloud-init as configured in CSv2
- Worker node registers with HTCondor when ready
- HTCcondor runs job on appropriate VM
- CSv2 shuts down worker VMs that remain idle for too long

- A VM in MRC OpenStack hosts HTCondor and HTCondor-CE instances
  - Host is running AlmaLinux9
  - Token authentication is supported
- Jobs submitted to HTcondor-CE on the HTCondor host
- After authorisation, jobs passed onto HTCondor by HTCondor-CE on same host
- CSv2 processes proceed as for Belle II

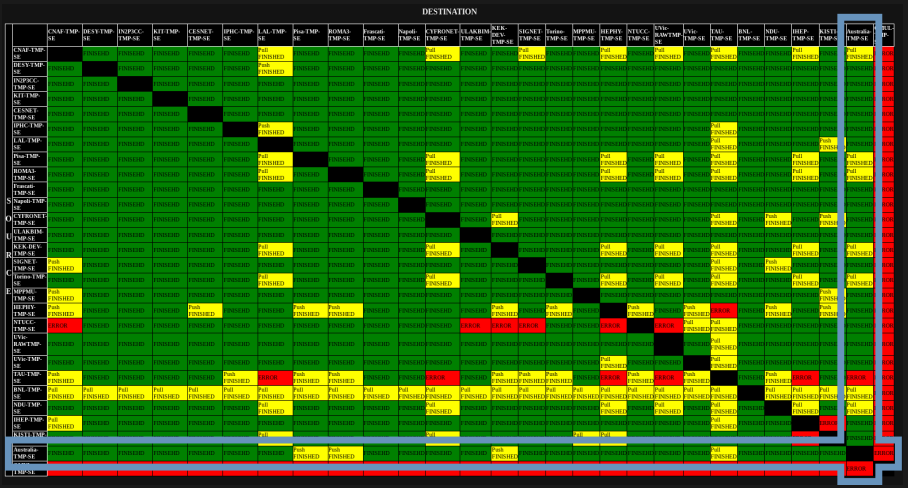
Current status

Belle II

- Belle II storage is operational (400 TB)
- Belle II compute is operational (900 vCPUs)

Current status

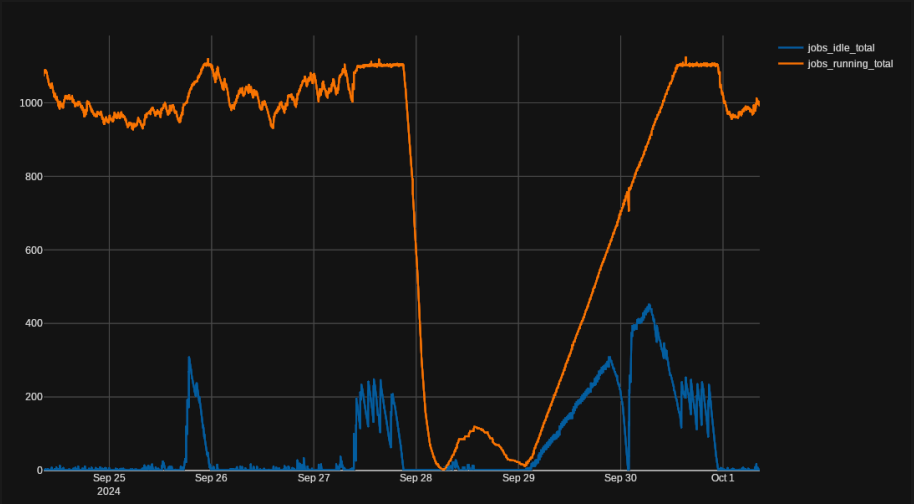
Belle II - TPC matrix



(from https://people.na.infn.it/~spardi/tpc-davs-latest.html)

Current status

Belle II - Compute jobs, running and idle



Current status

ATLAS

- ATLAS storage is ready for production testing (350 TB)
- ATLAS compute is being finalised (200 vCPUs initially)
  - HTCondor-CE accepts local job submissions and passes remote token-based access test at `https://novastore.farm.particle.cz/cgi-bin/condor.cgi`
  - All CSv2 processes work
  - Jobs are run by VMs as required
  - Remote HTCondor-CE access is being debugged
  - Looking to add additional vCPUs to ATLAS pool

Benchmarks

	Within cloud	In Australia
davs:// read	108 MB/s	40 MB/s
davs:// write	123 MB/s	74 MB/s
Checksum calc	3.2 s	3.4 s
Checksum fetch	0.72 s	0.98 s
s3 read	213 MB/s	n/a
s3 write	165 MB/s	n/a
root:// read	6.6 MB/s	5.9 MB/s
root:// write	132 MB/s	70 MB/s

Read/write tests used gfal-copy, checksum tests used gfal-sum. s3 tests on xrootd proxy server. Results are the average of 5 tests, each using a 1 GB test file.

Challenges

- Invisible application firewalls
- Slow `root://` read
- Read/write speed variability, particularly outside Australia
- AlmaLinux 9 environment

Future plans

- Resolve remaining issues with ATLAS compute infrastructure
- Bring ATLAS SE and CE into production together (the approach preferred by ATLAS)
- Monitor production transfers for Belle-II and ATLAS, add extra proxy servers as needed
- Increase storage and compute resources as funding allows.
  - Tentatively planning for an additional 1 PB in 2025, mostly directed towards ATLAS
  - Add 1000 vCPUs to ATLAS pool

Conclusions

- A grid site using cloud storage and compute is feasible
- The “Melbourne” site is in production for Belle-II (CE and SE)
- The “Melbourne” site is expected to also provide CE and SE resources for ATLAS soon