cloudScheduler V2: Distributed Cloud Computing

Randall Sobie

F. Berghaus, K. Casteels, C. Driemel, M. Ebert, F. F. Galindo C. Leavett-Brown,
D. MacDonell, M. Paterson, R. Seuster, S. Tolkamp, J.Weldon
University of Victoria
TRIUMF

cloudscheduler

Designed in 2009 to provision virtual machines (VMs) on clouds
(dedicated/pledged and opportunistic, private and commercial clouds)

ATLAS and Belle II experiments
CANFar - Canadian Advanced Network For Astronomy Research

Typical HEP workloads 5000 cores (peaks up to 10,000 cores)

Clouds in North America and Europe
(sometimes in Australia)

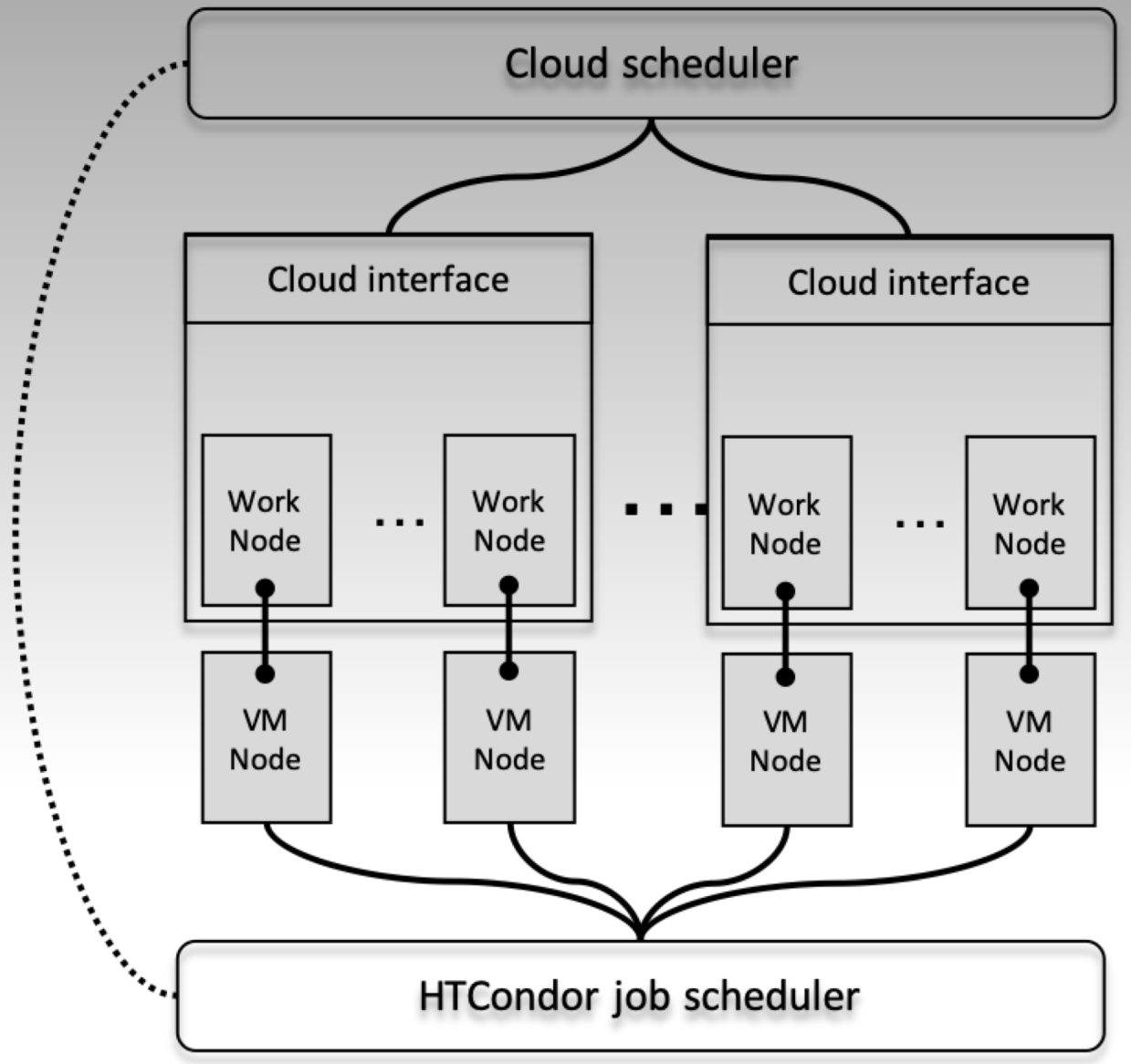
Openstack, Open Nebula, Amazon EC2, Google GCE, Microsoft Azure

Dynafed data federator

Belle II Canadian Raw Data Centre (Tier-1)

cloudscheduler Version 1
(HTCondor and CS run at Victoria)

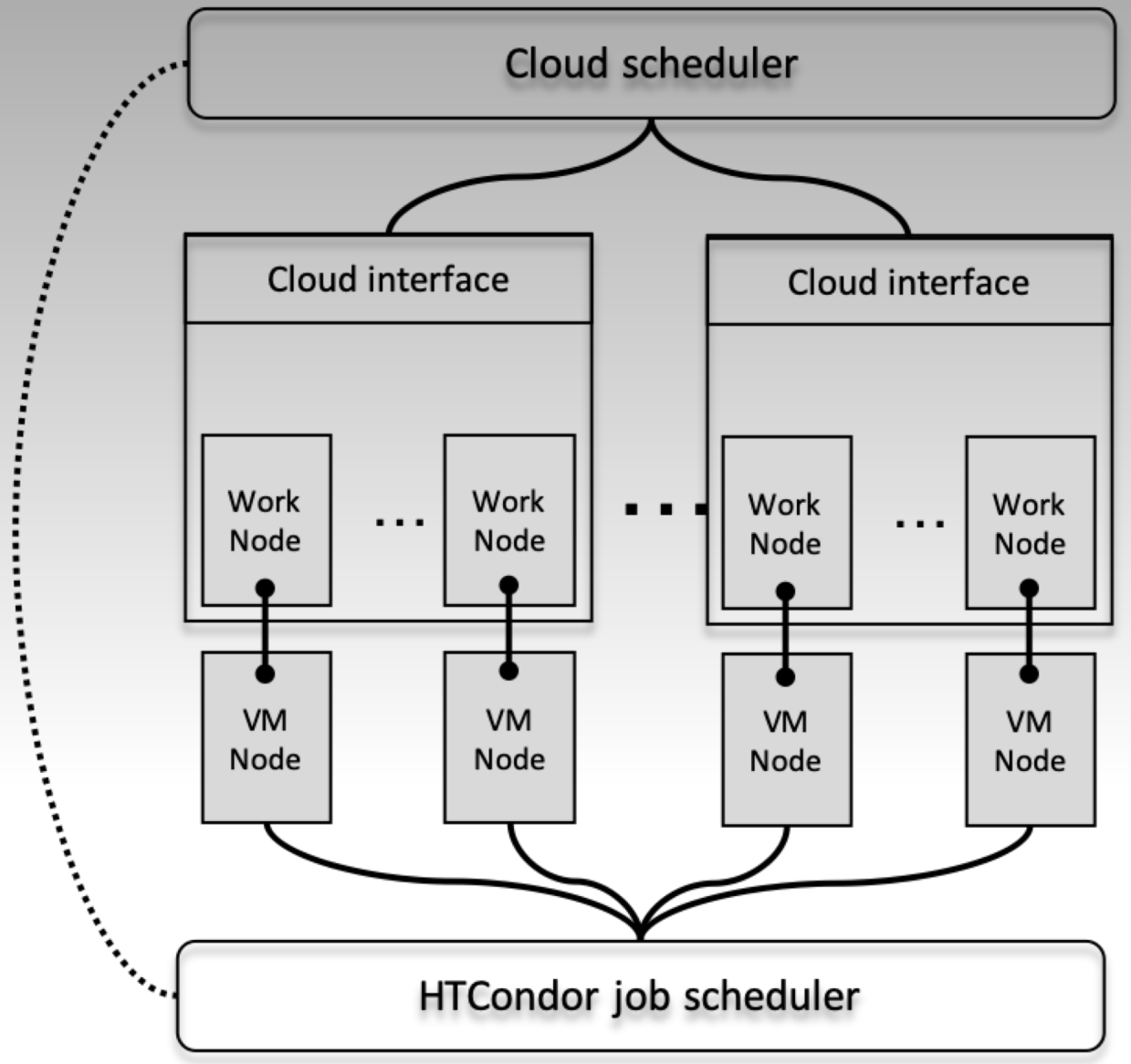
1. User submits job to HTCondor
   - Idle job and no resources
2. CS queries HTCondor for job requirements
3. CS looks for cloud that meets job requirements
   - If a match, requests creation of VM instance
4. Boot and contextualize a CERN micro-VM (CernVM)
   - Use CVMFS for OS/software
   - VM joins HTCondor pool
5. User job has resources and job executes on VM
6. Job completes
   - Other jobs in queue can run on VM
   - Otherwise, CS retires VM instance
7. CS destroys VM when there are no more jobs



cloudscheduler Version 1
(HTCondor and CS run at Victoria)

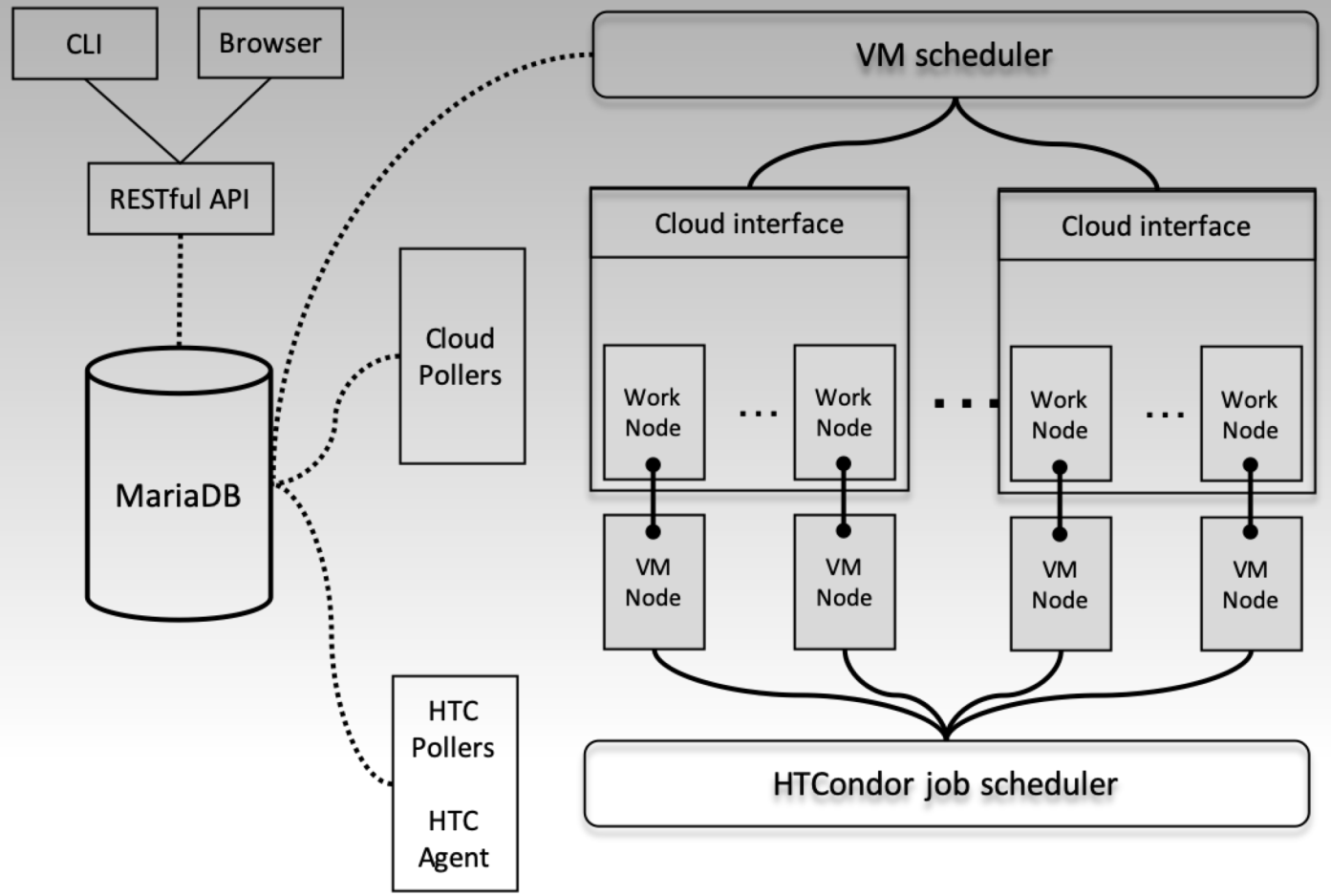
1. User submits job to HTCondor
   - Idle job and no resources
2. CS queries HTCondor for job requirements
3. CS looks for cloud that meets job requirements
   - If a match, requests creation of VM instance
4. Boot and contextualize a CERN micro-VM (CernVM)
   - Use CVMFS for OS/software
   - VM joins HTCondor pool
5. User job has resources and job executes on VM
6. Job completes
   - Other jobs in queue can run on VM
   - Otherwise, CS retires VM instance
7. CS destroys VM when there are no more jobs

Time for a rewrite after 10 years
Python 3, redesigned architecture, GUI, ..
New functionality, opportunistic use of clouds, ..



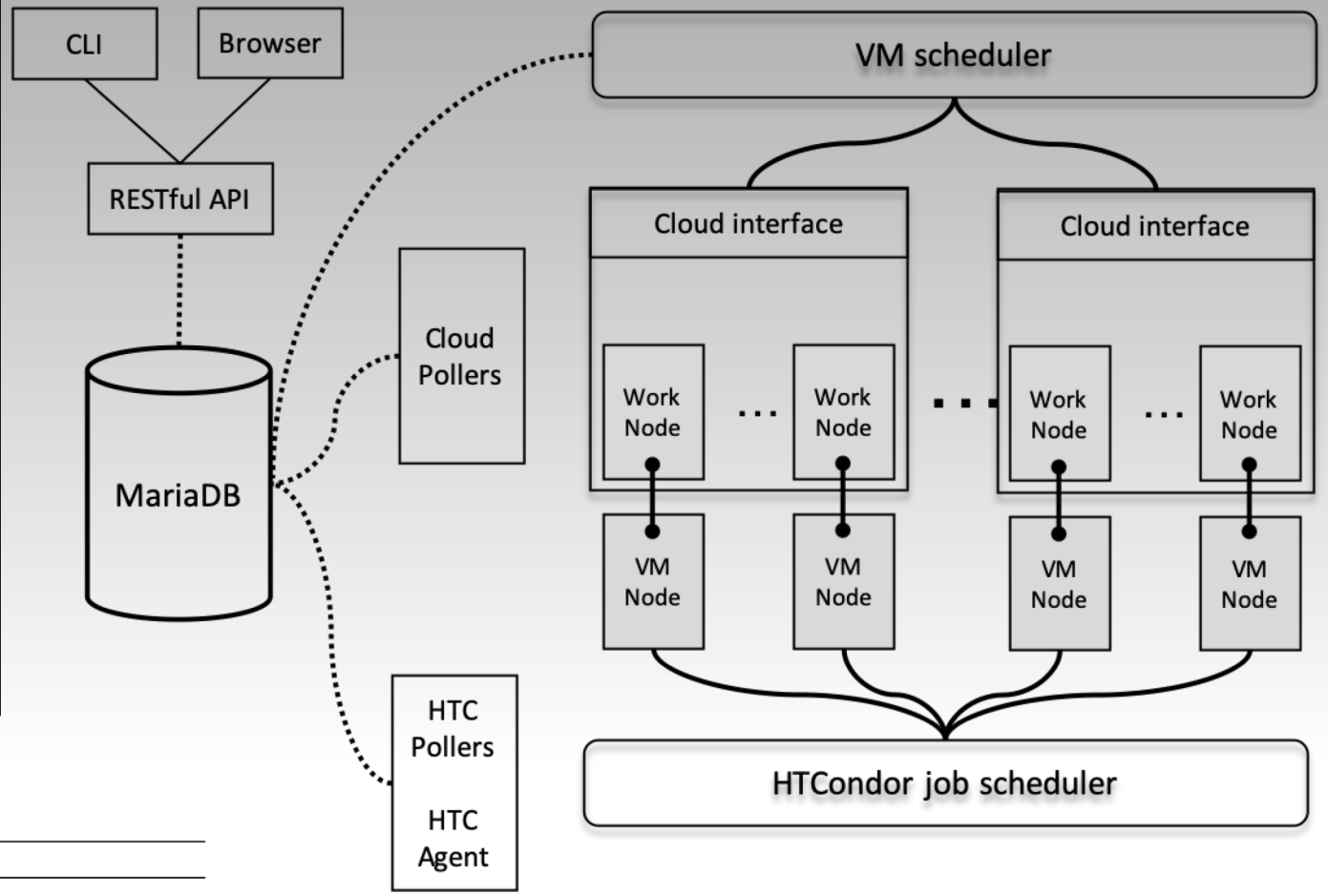
Cloudscheduler Version 2

- In-memory data of CSV1 are replaced with a MariaDB database
  - Keep track of system state in MariaDB
  - More responsive with MariaDB
  - Resilient to outages and easier to maintain
- Independent scheduling, polling, and user interface processes that track the state of the clouds and the HTCondor pool
  - Improved VM and job scheduling
- RESTful web user interface for
  - Improved administration and management
  - Monitoring of the clouds and jobs
- Expanded functionality
  - Multiple projects (experiments)
  - Multiple HTCondor instances
  - Opportunistic sharing in a cloud



Cloudscheduler Version 2 Workflow
(Similar to CSV1)

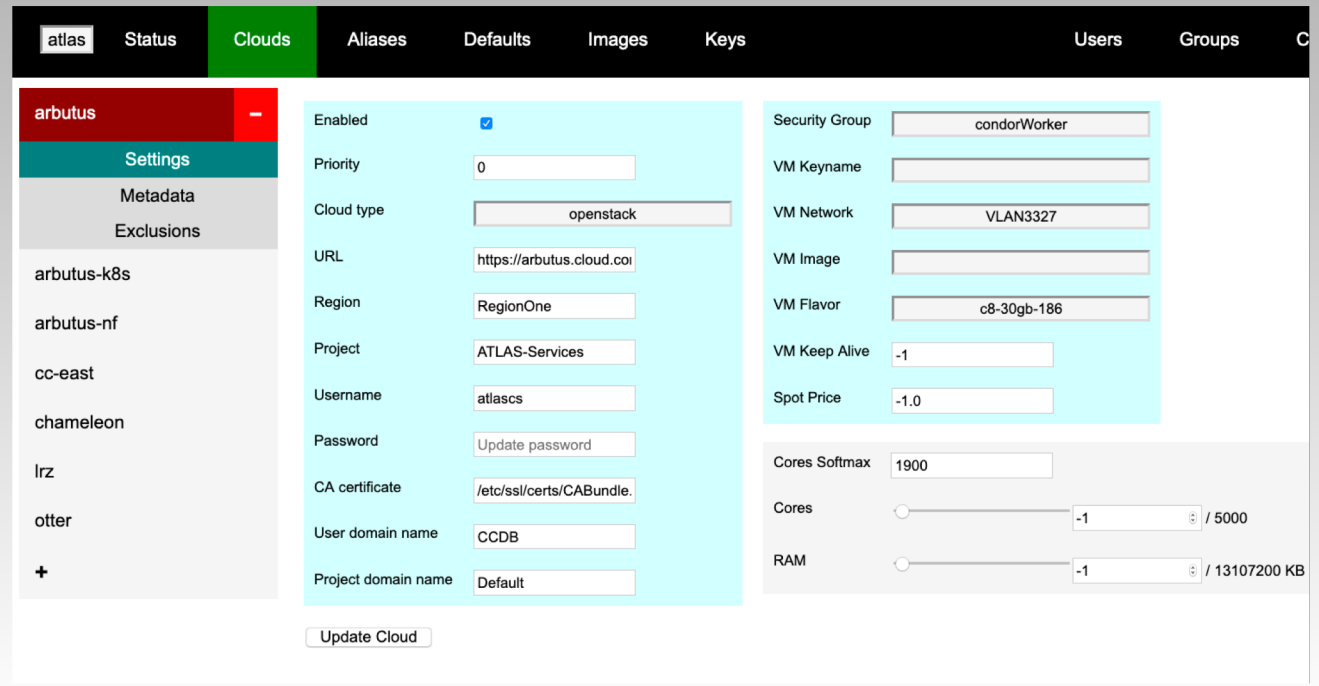
- Determined by configuration of the system, clouds, job information and the state of VMs
- Mandatory contextualization
  - metadata passed to “cloud-init”
- Additional contextualization
  - Project configuration
  - Benchmark VM
  - Activate ES reporting by the VM
  - Apel accounting



VM state	Description
starting	VM is booting/contextualizing
unregistered	VM is running and has not registered in HTCondor pool
idle	VM is running, registered in HTCondor pool and not running jobs
running	VM is running, registered in HTCondor pool and running jobs
retiring	VM is running, retired in HTCondor pool and will complete running jobs
manual	VM is flagged as being manually used and will be ignored by the *VM Scheduler*
error	VM in error state according to the cloud information

Cloudscheduler Version 2 GUI/CLI
(CLI and GUI have nearly identical functionality)

- CSV2 has users and groups (group = expt)
  - Access rights configurable
- Clouds configuration
  - Quotas, instance types
  - Opportunistic sharing
- VM images and SSH keys
- Distribute VM images to Openstack clouds
  - (formerly “glint” service)

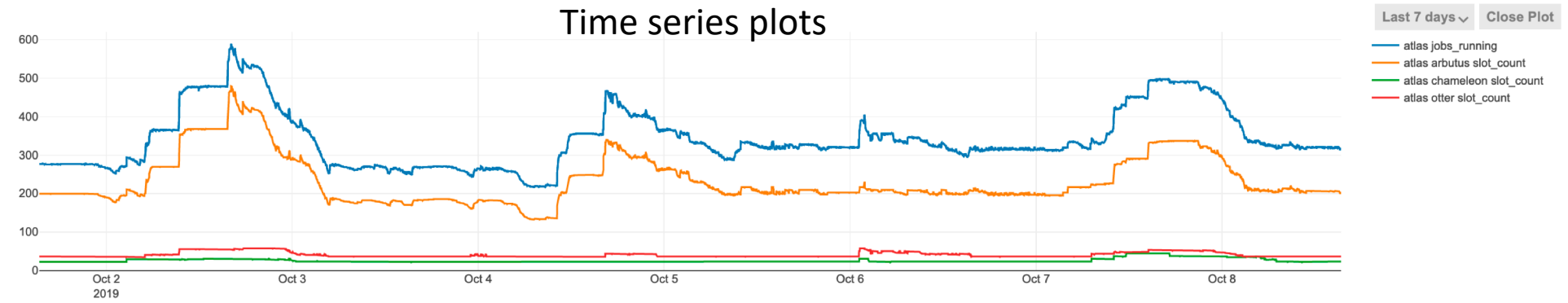


Monitoring

Jobs
VMs

<u>Also</u>
HTCondor slots
Core usage
Memory
Central services

Time series plots



Group	Jobs	Idle	Running	Completed	Held	Other
atlas	510	151	319	40	0	0

Foreign
0

Condor FQDN	Condor Status	Agent Status	Condor Cert	Worker Cert
csv2a.heprc.uvic.ca	✔	✔	✔	✔

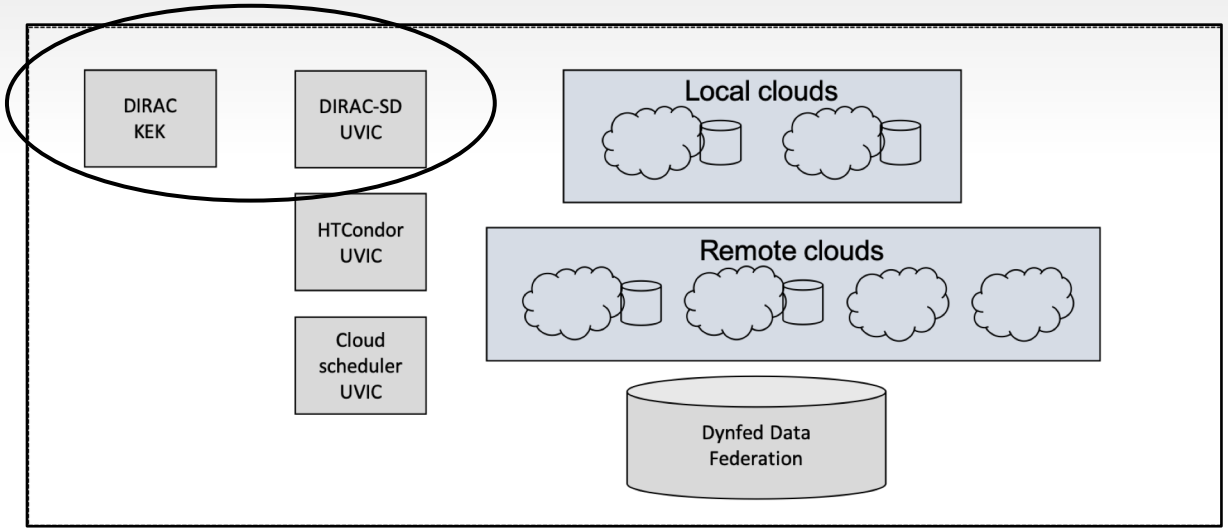
Jobs

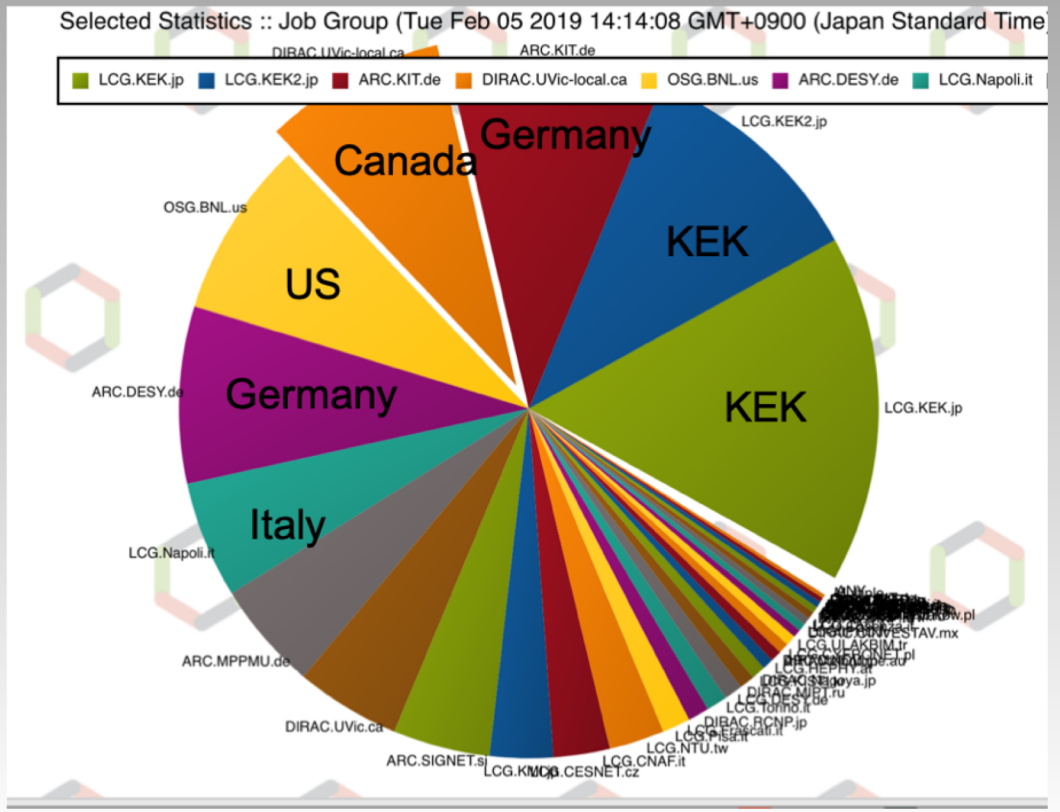
Current status

Group	Clouds	RT (ms)	VMs	Starting	Unreg.	Idle	Running	Retiring	Manual	Error
atlas	arbutus	269	200	0	0	0	200	0	0	0
	arbutus-k8s	116	0	0	0	0	0	0	0	0
	arbutus-nf	269	18	0	0	0	18	0	0	0
	cc-east	162	17	0	0	0	17	0	0	0
	chameleon	141	24	0	0	0	24	0	0	0
	lrz	110	0	0	0	0	0	0	0	0
	otter	101	37	0	0	0	37	0	0	0
	Totals		296	0	0	0	296	0	0	0

Belle II

- Uses the DIRAC workload management system
  - 3 DIRAC “Site Directors” (SD) in Victoria
  - SD submits pilot jobs to HTCondor based on queues in KEK
- CS system shared with ATLAS (separate HTCondor systems)
- Storage element (grid storage), Ceph (object storage)
- Dynafed federation for managing remote cloud storage



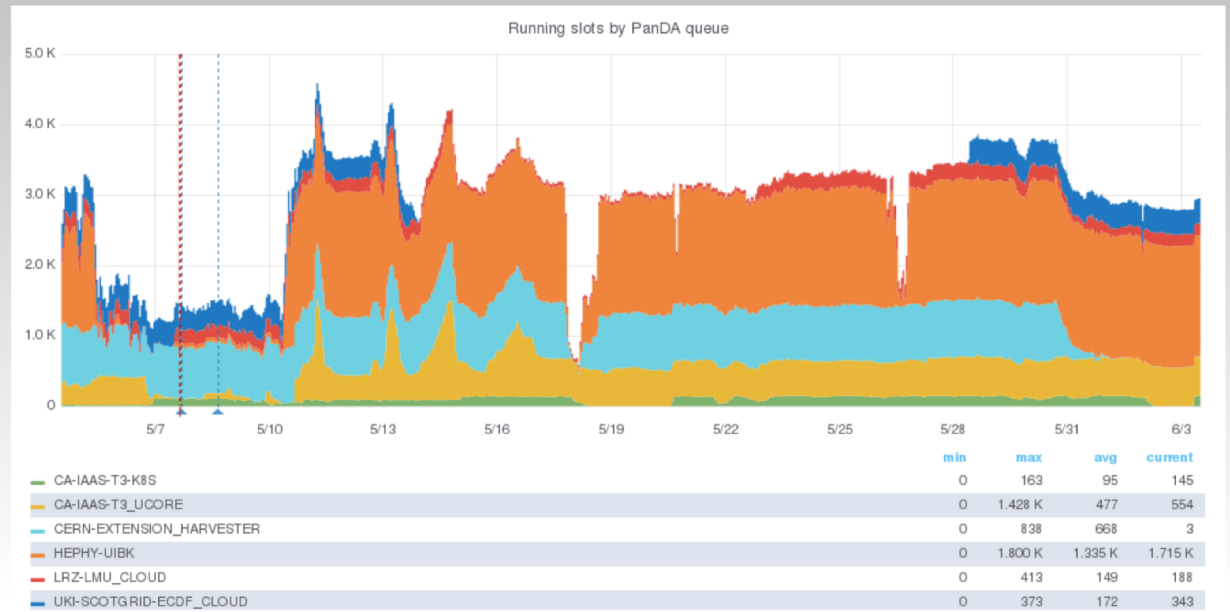


Distributed cloud provides 10% of B2 computing

Canadian Raw Data Centre will store 15% of the 2nd copy of the raw data in 2021 (using the “cloud” infrastructure)

ATLAS

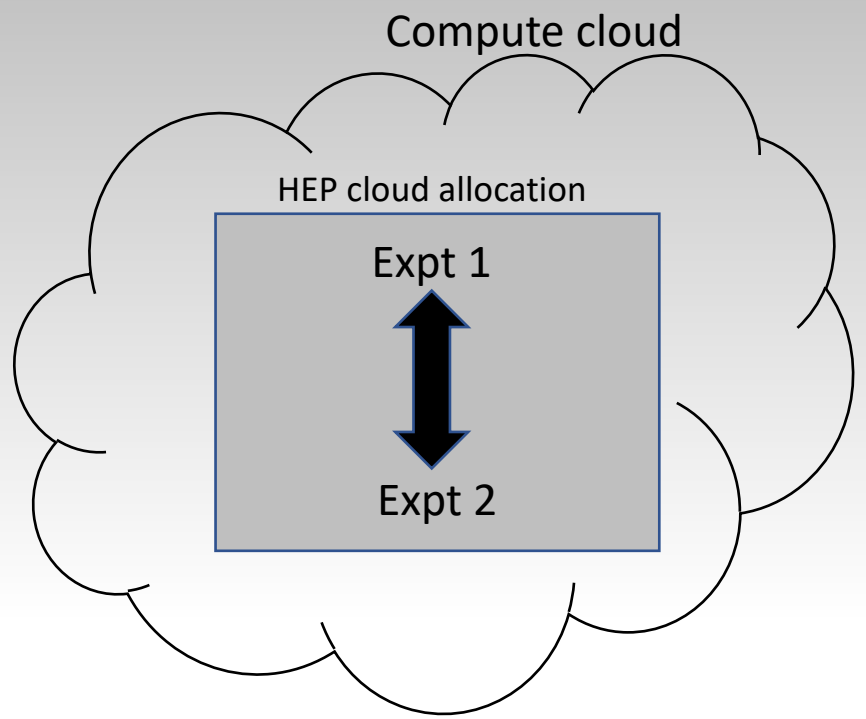
- Two systems:
  - CSV2 in Victoria for Belle II/ATLAS using clouds in North America
  - CSV1 in CERN for ATLAS using clouds in Europe
- Separate PanDa queues for each European cloud
  - Each site gets credit for their resources
- Support clouds in Edinburgh (ECDF), Munich (LRZ), Innsbruck and CERN (around 3000 cores)
- Plan to migrate all clouds to CSV2 in Victoria
  - Belle II HTCondor instance
  - Multiple ATLAS HTCondor instances



Distributed cloud provides 1% of ATLAS resources (comparable to a Canadian Tier-2 site)

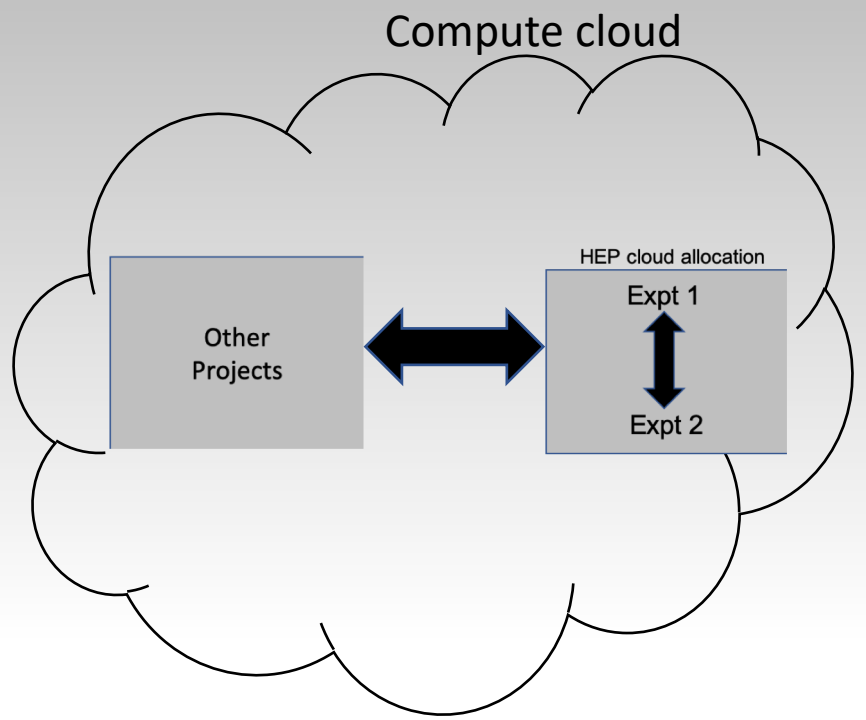
Opportunistic use of idle HEP resources

- We have clouds for ATLAS, Belle II or HEP
- There are times when the workload on one project is low
- If the cloud is open to opportunistic use, then we shift the idle resources to the other experiment
- Once jobs appear, the opportunistic VMs are retired, the jobs are allowed to complete and the VMs destroyed
  - Typical turnover in 6-12 hours
- CSV2 can be configured for opportunistic use within a cloud
  - Can use a configurable fraction of the other resources



Opportunistic use of idle non-HEP resources

- The goal is to use the idle resources of other projects
- We have no information on the resources allocated to other users
  - Exploring what information is required
  - How to have the cloud-operators pass it to CSV2
  - Automated system rather than manual
- Consider how to manage these resources:
  - Gentle transition (letting jobs finish)
  - Immediate termination
  - Single event generation

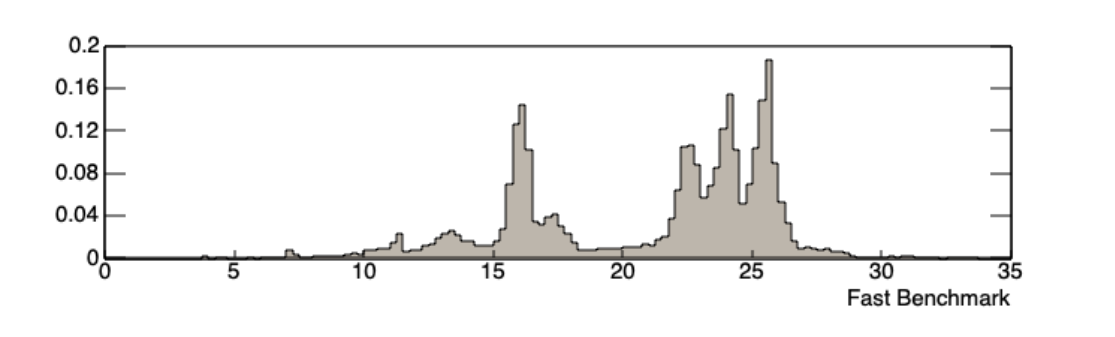


Accounting

Experiments:
- Belle II keeps track within DIRAC using the fast-benchmark (DB12)
- ATLAS resources are tracked with PanDA

Distributed cloud
- Measure the CPU performance of each VM at boot (DB12 with HEPiX normalization)
- Record the CPU/wallclock time and write to ES

Apel Accounting
- Now storing the variables required for Apel accounting
- Written the code to extract and upload the information to MariaDB
- Developing script to upload to Apel database
- Should be operational Nov 2019
- Credit assigned to group, cloud or country



We do not know the underlying hardware

HEPSpec06 takes hours to run (license issues)

Run the fast-benchmark (DB12) at boot on each VM

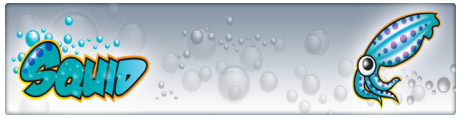
Not an ideal benchmark – waiting for new one

Distributed cloud (using cloudscheduler) is a platform of external and in-house components

Preference is to use external components and develop those that do not exist

External services

CernVM
CVMFS
DB12 and HEPSpec06 Benchmarks
Squids
Dynafed data federator
HTCondor job scheduler
Elasticsearch database
Apel accounting
Ceph object storage
Openstack/EC2/GCE/Azure clouds
MariaDB
Docker/Singularity containers
HTTP/WebDAV
InFluxDB



Summary

The distributed compute cloud using cloudscheduler and HTCondor has been running production jobs for ATLAS and Belle II for many years
(also used for Canadian astronomy workloads)

We have utilized dedicated and opportunistic, research and commercial clouds
All usage to be reported to Apel accounting for each site/region

The system has undergone an extensive revision (cloudscheduler V2, MariaDB, ..)
Significant improvement in reliability, responsiveness and management
Opportunistic sharing of HEP resources (and non-HEP resources soon)

The Canadian Belle II Raw Data Centre will use this technology

<u>Related talks at this conference:</u>
Using Containers for managing infrastructure in ATLAS
Using Dynafed as a distributed storage element
Sim@P1 (ATLAS), a cloud system utilizing the HLT cluster at CERN

Dynafed
The Dynafed data federator as grid site storage element, F.Berghaus, CHEP2019

Using a dynamic data federation for running Belle-II simulation applications in a distributed cloud environment, CHEP2018
https://doi.org/10.1051/epjconf/201921404026

Integrating a dynamic data federation into the ATLAS distributed data management system, CHEP 2018,
https://doi.org/10.1051/epjconf/201921407009

cloudscheduler
High-throughput cloud computing with the cloudscheduler VM provisioning service. Submitted to Software and Computing in Big Science 2019
http://heprcdocs.phys.uvic.ca/papers/cs.pdf

Quasi-online accounting and monitoring system for distributed clouds, CHEP 2018,
https://doi.org/10.1051/epjconf/201921407035

sim@P1
Title: ATLAS Sim@P1 upgrades during long shutdown two, F. Berghaus CHEP 2019

Containers
Using Kubernetes as an ATLAS computing site, CHEP 2019