Cloudscheduler (V2)


University of Victoria
Cloudscheduler

Provision CernVM to execute batch workloads on cloud infrastructure

ATLAS - HEP Experiment at CERN
  ● Cloudscheduler: Integration of cloud resources
  ● Sim@P1: Opportunistic use of high level trigger resources

Belle-II - HEP Experiment at KEK

CANFar - Canadian Advanced Network For Astronomy Research
  ● Using cloudscheduler and testing CernVM
Cloudscheduler V2

Version 1 released in 2009 - time for a rewrite

- Python 3, architecture redesign, graphical user interface
- https://github.com/hep-gc/cloudscheduler

Version 2

- Driving ATLAS cloud queues
- Future: Belle-II cloud resources, container workflows
Workflow

0. Empty queue, no workers
   - Condor & CS run at UVic
1. User submits job
2. Idle job and no resources
3. Create instance on any cloud with matching resources
   - CernVM batch version
4. Instances connect to Condor
5. Jobs execute
6. No idle jobs in queue
7. CS retires instances
   - Delete on job completion
Reading from services

- Monitor
- Condor Jobs
- Condor Resources
- OpenStack Instance
- OpenStack Images
- Cloudscheduler status

Cloudscheduler

Instructions to clouds
Monitoring

Tables of operational variables

Time series plots!
Image distribution

Image distribution between clouds

Usually CernVM images :)

Lead transition to SL7/CentOS7 with CernVM4 for ATLAS
Cloud configuration is info from resources file

Adding support for more cloud infrastructures and configurations

Running already on many clouds (OpenStack and EC2)
Cloudscheduler in ATLAS

Used in commissioning & integration:

- Singularity, containers, pilot2, HTTP/WebDAV,
Cloudscheduler in Belle-II

Provides Canadian computing pledge
Uses Dynafed as data access layer (gfalfs)

M Ebert et al in proceedings CHEP2018
Other noteable CernVM & CVMFS related work
**Shoal**

(Squid) proxy discovery

- shoal.heprc.uvic.ca

Used for CVMFS and frontier on clouds

1. Agent on cache advertises to server
2. Server performs function test on cache
3. Server registers cache
4. Client requests cache(s)
5. Server calculates best match(es)
   a. ACL, GeoIP, and load
6. Server responds to client

Gable et al arXiv:1311.0058 [cs.DC]
Sim@P1

Opportunistic use of ATLAS HLT farm

$O(100k)$ cores

Virtualization provides isolation

Boot with libvirt tools

Configuration with config disk

Three (3) dedicated squids

Boots 2,500 instances in 2 minutes
Conclusion

Cloudscheduler orchestrates CernVM instances

CernVM provides stable system that allows us to test and develop

Container testing ongoing

- Pilot calls singularity
- Kubernetes manages containers running Pilot