Integrating Kubernetes batch queues using Harvester

Fernando Barreiro Megino
University of Texas at Arlington

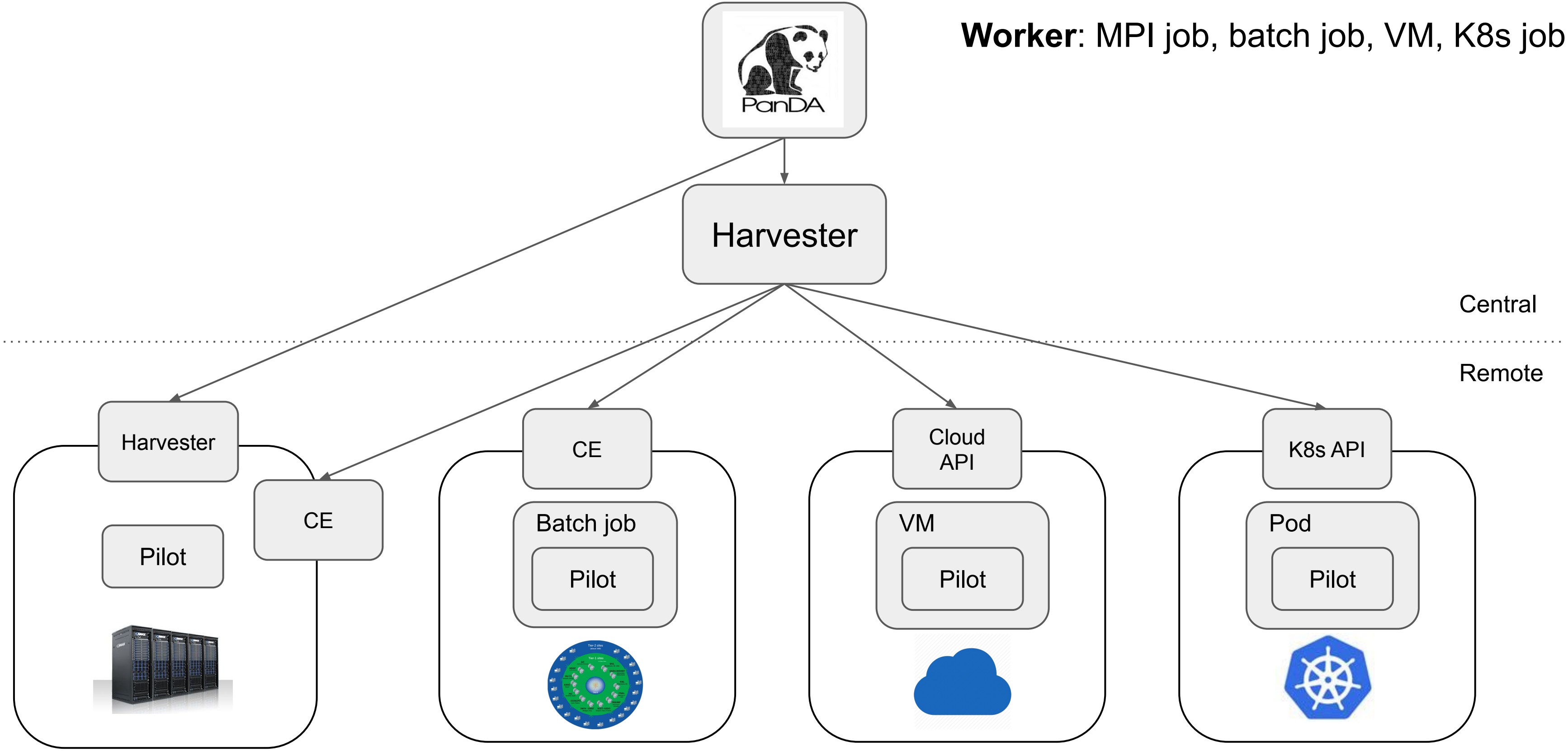
GDB, 8 July 2020

Credits

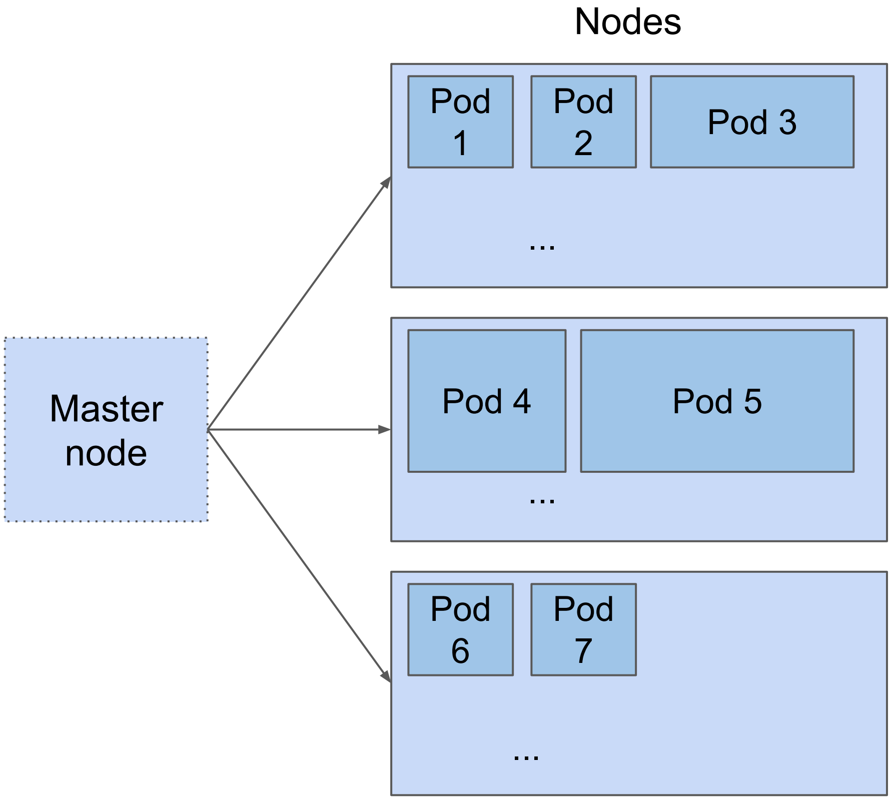
- Harvester team: FaHui Lin (UTA), Tadashi Maeno (BNL), Han-Sheng Peng (ASGC), Mandy Yang (ASGC)
- Rucio team: Mario Lassnig (CERN), Cedric Serfon (BNL), Tobias Wegner (UWuppertal)
- Sites: Ricardo Rocha (CERN), Lincoln Bryant (UChicago), Danika McDonnell (UVic), Ryan Taylor (UVic)

Harvester: universal worker submission

Worker: MPI job, batch job, VM, K8s job



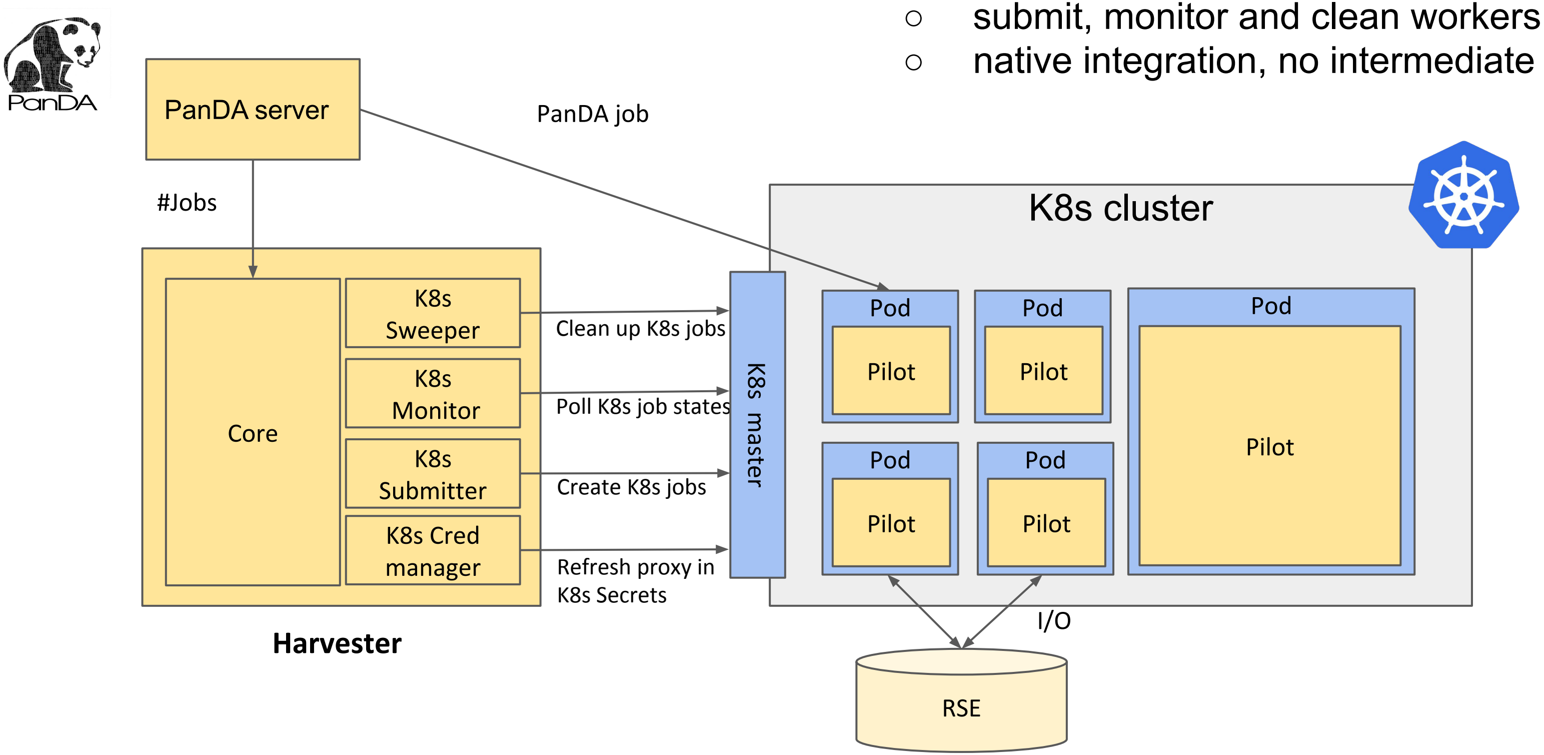
K8s basics



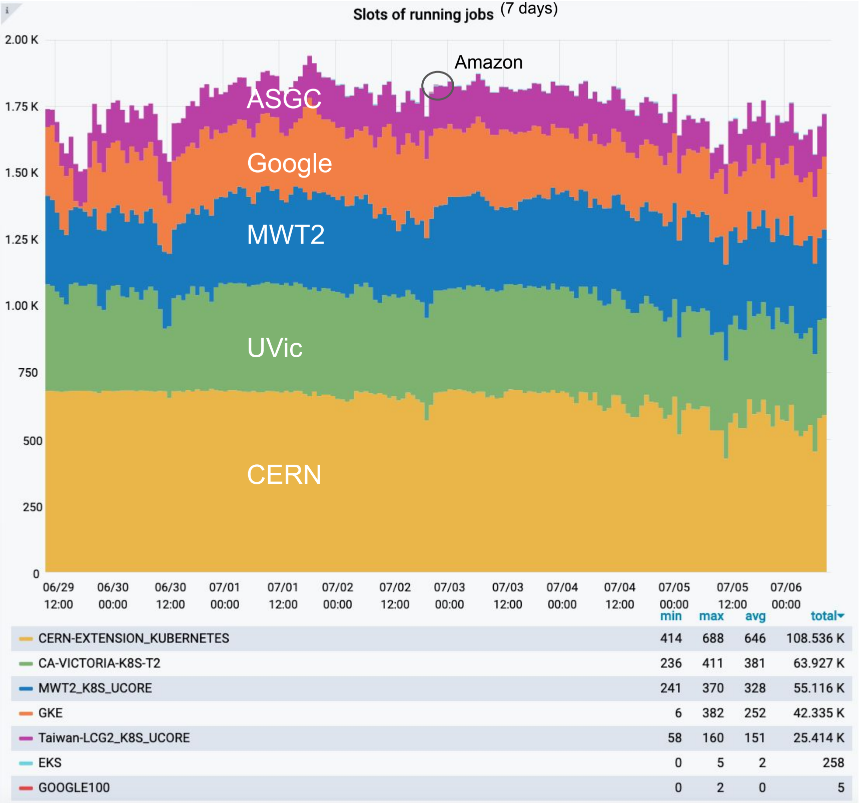
- **Cluster**: group of nodes
- **Node**: VM or physical machine
- **Pod**: scheduling unit, can contain one or more containers
  - Can run a job, a service...
  - CPU and memory reqs define pod QoS
    - No reqs: best effort
    - Reqs: burstable or guaranteed
- **Kubernetes**: schedules and manages pods across the cluster
- **Controllers**: rules pod scheduling/lifecycle, e.g.
  - Job: execute and repeat n times until finished (e.g. the ATLAS job)
  - Daemon set: one pod copy per node (e.g. the CVMFS CSI driver)
  - Replica set: n pod copies anywhere
  - …
- Many storage and network features that go beyond our usage

Harvester K8s integration

- **Core**: implements most of the Harvester intelligence
- **Plugins**: resource integration
    - submit, monitor and clean workers
    - native integration, no intermediate layers

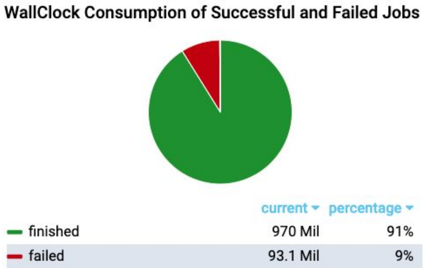


Current ATLAS K8s resources



	min	max	avg	total
CERN-EXTENSION_KUBERNETES	414	688	646	108.536 K
CA-VICTORIA-K8S-T2	236	411	381	63.927 K
MWT2_K8S_UCORE	241	370	328	55.116 K
GKE	6	382	252	42.335 K
Taiwan-LCG2_K8S_UCORE	58	160	151	25.414 K
EKS	0	5	2	258
GOOGLE100	0	2	0	5

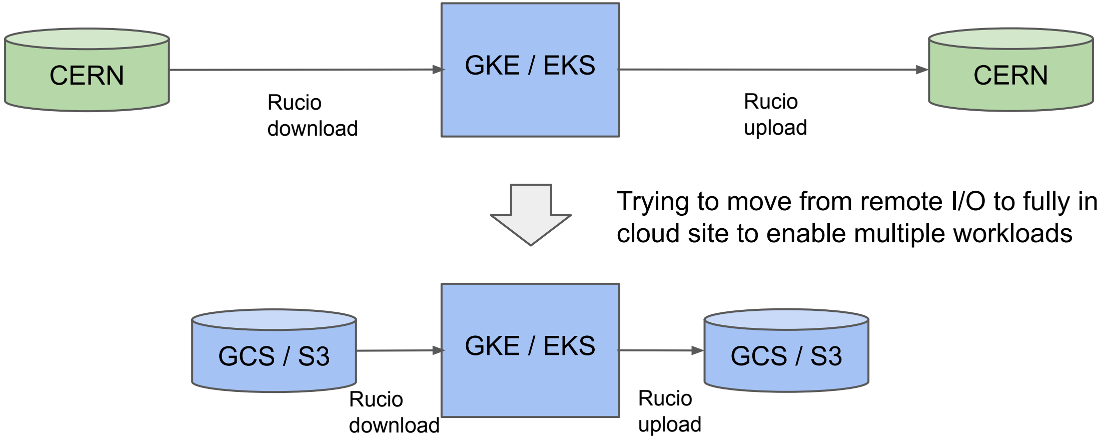
- Same integration for own clusters as for clusters provided by major cloud providers
- Various reasons to build K8s cluster
  - Simpler compute setup
  - R&D cluster to host various services
  - ATLAS R&D quota
  - Resources that need to be integrated at institutional clouds
  - Projects with cloud providers



	current	percentage
finished	970 Mil	91%
failed	93.1 Mil	9%

Google & Amazon exercises

- **Google**: running 240 core simulation cluster. Cost: ~100 USD/day
  - Preemptible: nodes can live up to 24h, 80% cheaper
    - Restricting queue to <5h jobs
  - Autoscaled: cluster ramps up/down depending on #submitted jobs
- **Amazon**: demonstrated basic integration with HC test jobs
  - First discussions for US ATLAS T3 project
- Working together with Rucio team: integrate cloud storage to enable I/O intensive workloads

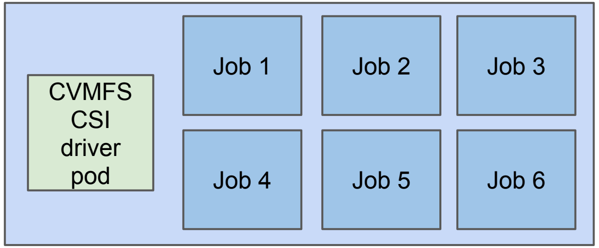


Trying to move from remote I/O to fully integrated cloud site to enable multiple workloads

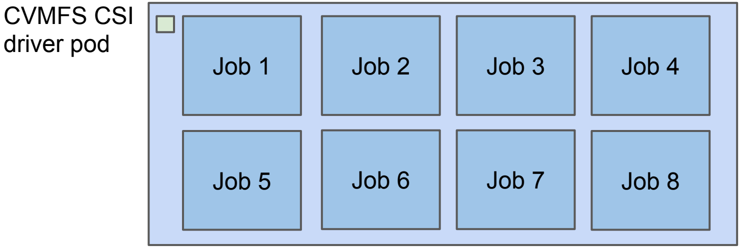
CVMFS installation options on K8s

- Direct installation on nodes
  - Easiest and most stable installation
  - Not possible when no control over nodes
- CVMFS CSI driver (Ricardo/CERN IT)
  - CSI: Container Storage Interface
    - One CVMFS node cache. One bind mount per pod
    - Requires implementation of driver interface (golang) and some auxiliary pods
      - Relatively complex
  - Some operational issues. Learning how to deal with it

CVMFS driver submitted without CPU/mem requirements



Not fully packed: CSI gets resources and runs OK



Fully packed: CSI gets killed (node OOM) or throttled (node CPU full)

CVMFS on K8s (cont.)

- prp-osg-cvmfs (Igor Sfiligoi)
  - Heard about during GDB preparation for the first time
  - Similar concept as CERN IT's solution, but does not require any CSI driver implementation
    - Much simpler, while also providing good efficiency through shared CVMFS node cache
    - Definitely to be evaluated
- Bottomline: ATLAS depends on CVMFS for any SW distribution and an officially supported solution would be greatly appreciated

Miscellaneous

- Harvester Helm chart: easy Harvester installation on K8S
    - First pre-prod installation
        - Evaluate stability and operational experience
    - US ATLAS HPC managers going to evaluate/adapt/extend for satellite installations
- APEL Accounting: colleague working on central APEL feeder, but we need some guidance

Conclusions

- K8s provides a simple, industry-wide accepted solution
- Resources grew significantly in the last months
  - Making service more robust and improving operations
- Pioneering sites like the model
- Standard integration of major cloud providers for compute
  - Cloud storage integration in Rucio & FTS also becoming a reality
  - Ironing out last details for fully native cloud site
- Whole world of possibilities for native user container submission (still to be evaluated)
- Some CE features (accounting, fairshares) need to be worked on