Native Kubernetes-ATLAS integration through Harvester

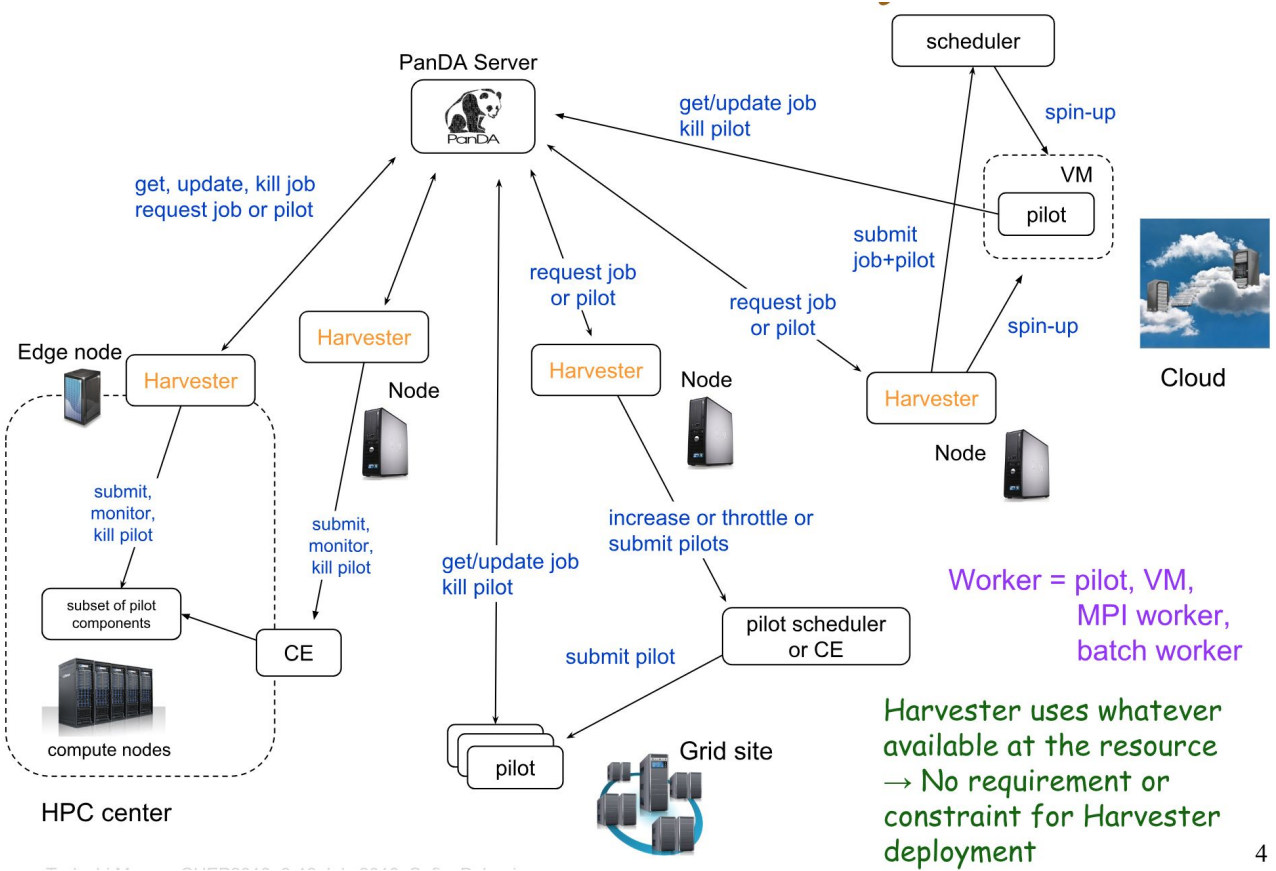
Fernando Barreiro Megino

Pre-GDB on Kubernetes, CERN 10 Dec 2019

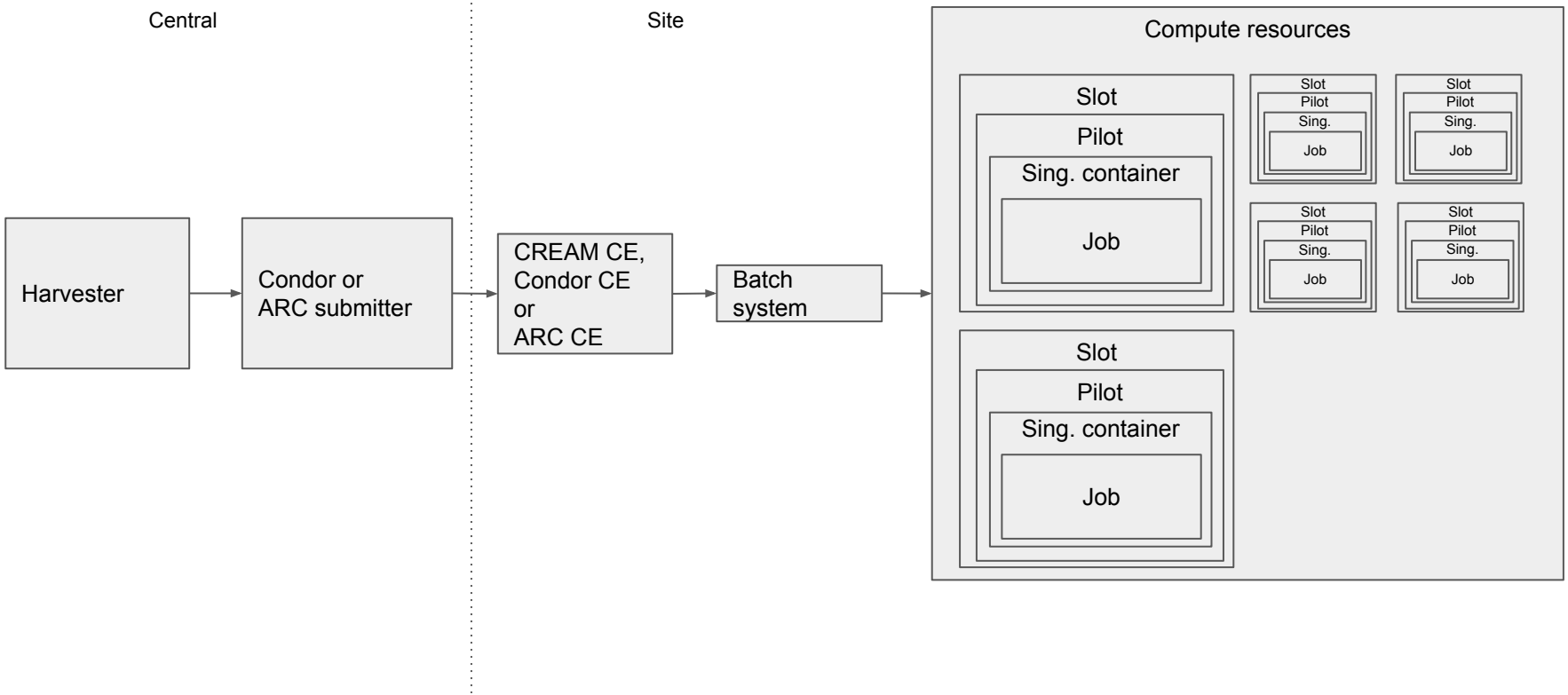
Introduction

- Idea born during ATLAS-Google Data Ocean project
  - Containers as lightweight alternative to VMs in GCE (or other clouds)
  - But also Kubernetes as potential open-source replacement for WLCG CE+batch infrastructure
    - No HEP-specific middleware
    - Container-native environment
- ATLAS investigating in exploratory capacity
  - Harvester interfacing experiment WMS and Kubernetes
  - Native, minimalistic setup. Only keep necessary layers
  - Production ATLAS queues ran successfully at CERN and UVic
- Initial working model available and ideas to improve the system

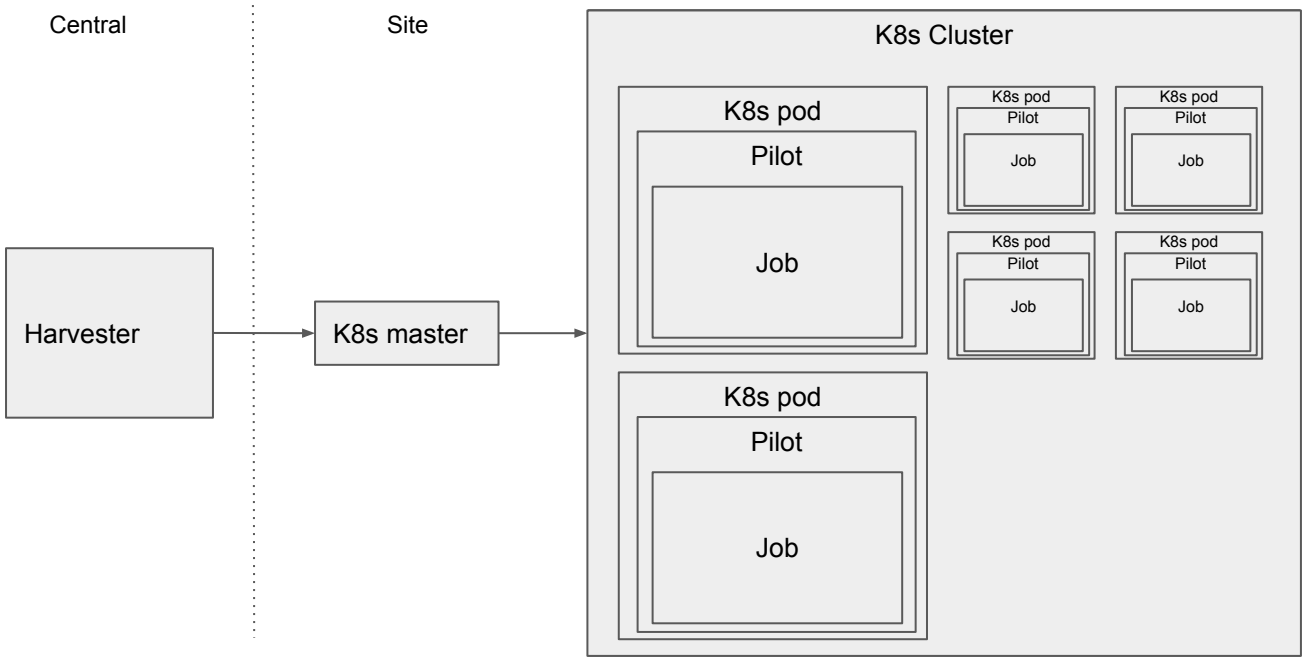
Harvester: universal resource interface



Layers for ATLAS grid/batch setup

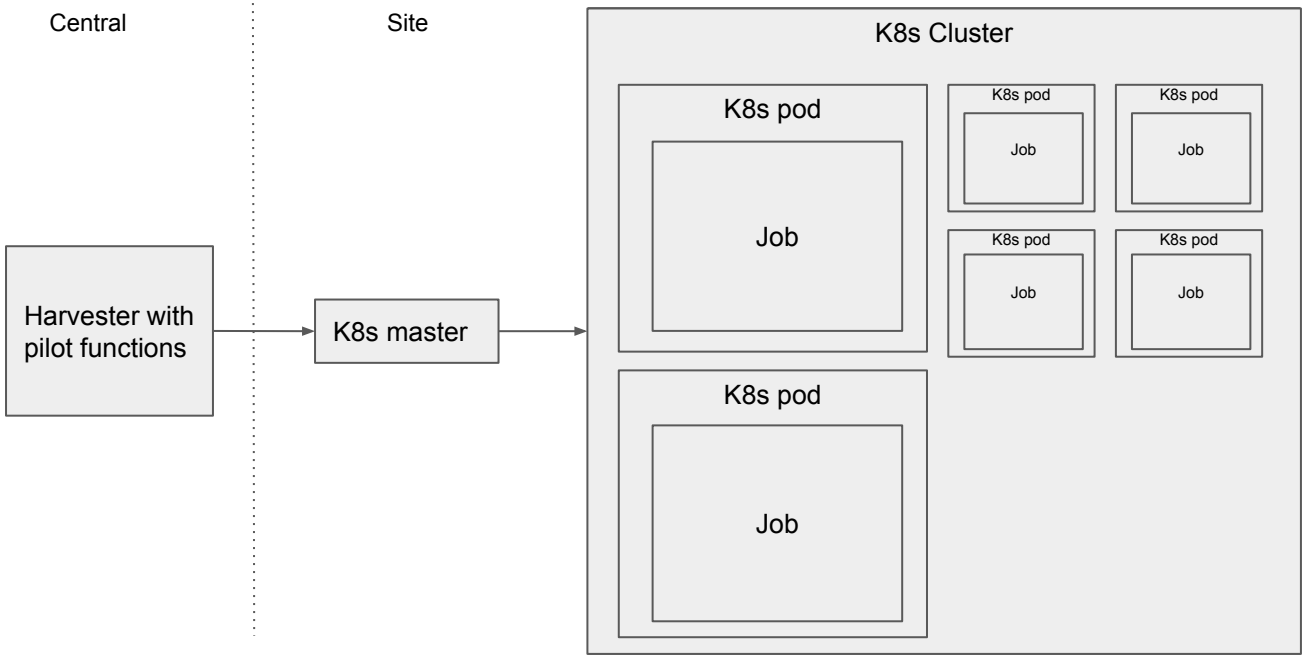


Current ATLAS K8s setup



(Currently with limitations, e.g. on container images)

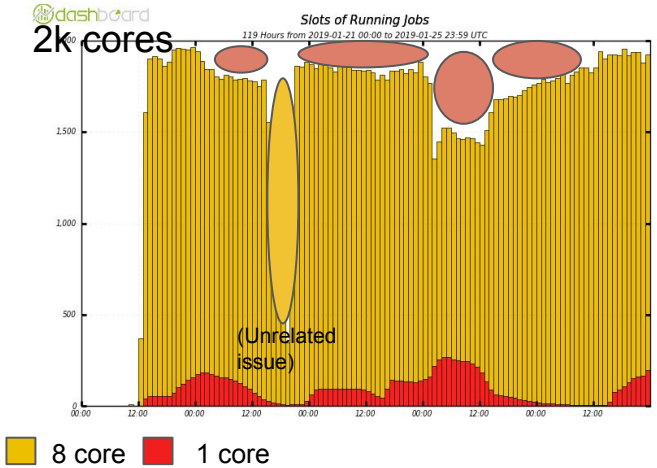
Ideal K8s setup



Production at CERN: Jan-Feb 2019

Temporary ~1 month exercise on loaned HW. Standard production (no test jobs) queues

Beginning of the exercise: default K8s scheduling
21-25 Jan 2019



Default scheduling: spreads 1 core pods across nodes (8 cores), blocking out 8 core jobs

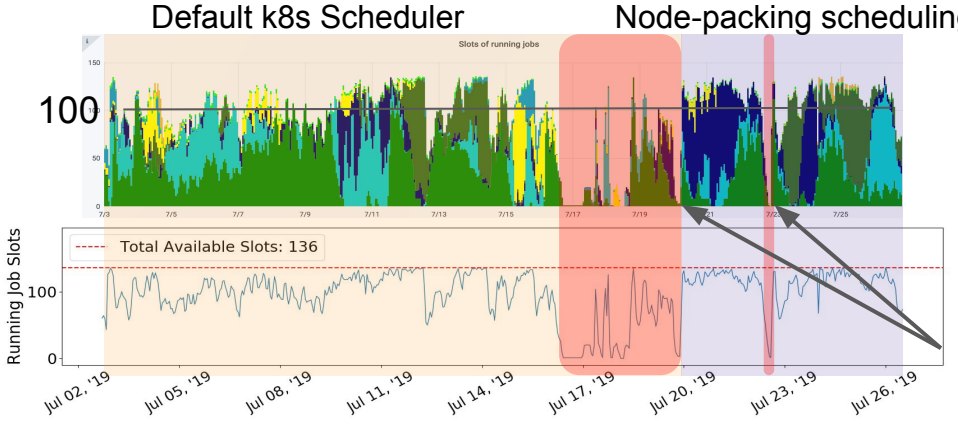
Later in the exercise: policy tuning to pack nodes
14-17 Feb 2019

Scheduling policy: packs multiple 1 core pods together. No significant loss, except for node draining (same as in batch systems)

Throughout the exercise we kept losing nodes for multiple infrastructure reasons that should have been solved by now

Production results at UVic

See Danika and Ryan's presentations later today



Unrelated issues

UVic T3: test cluster for K8s cluster setup development

UVic T2: this is a permanent production queue for ATLAS now

Conclusions

- Industry standard
- Important simplification and reduction of layers
- Native containerized environment, suitable also for analysis preservation
- Demonstrated ATLAS production queues on pure K8s
  - No efficiency losses or abnormal error rates
- Room for improvement and word of caution
  - Evolve integration model and increase scale
  - Unanswered questions (e.g. fairshares, authentication/authorization model, accounting)
  - Reimplement a decade of work & customization
  - Will we lose control on evolution?

Acknowledgements

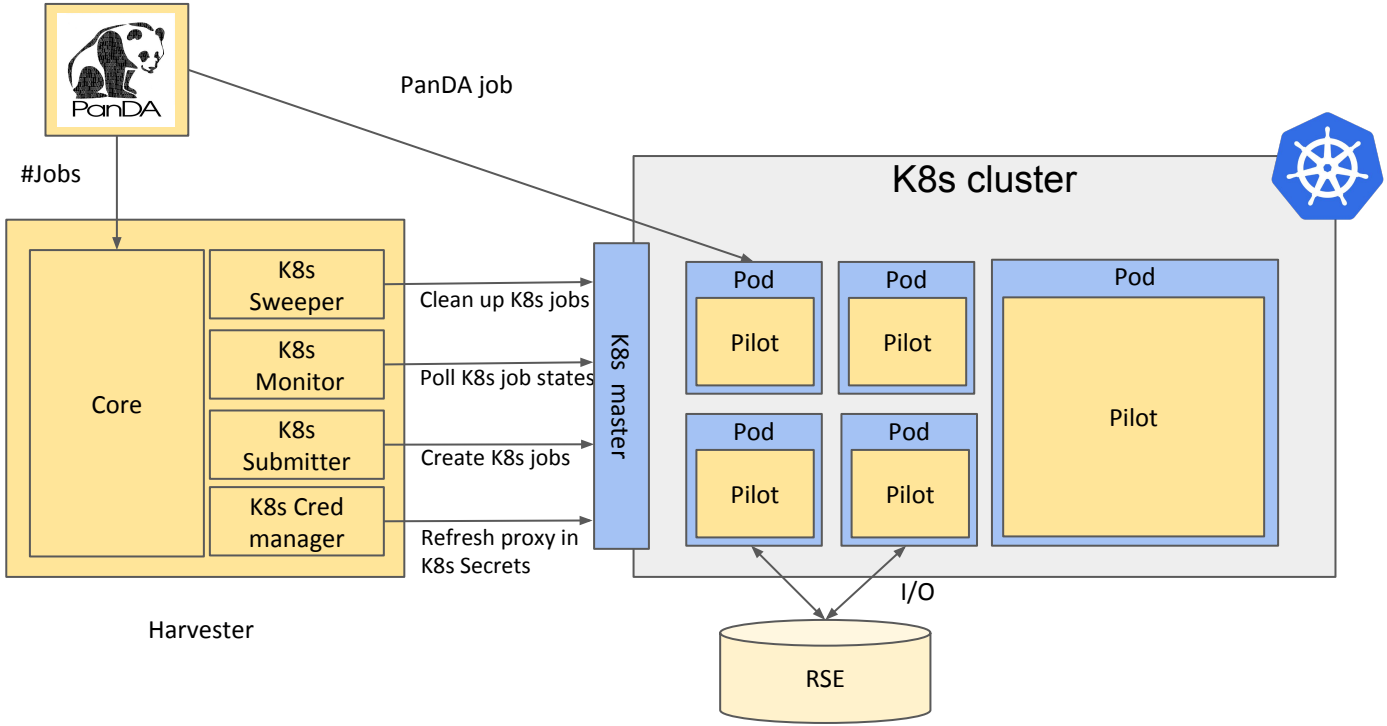
Harvester development: FaHui Lin, Tadashi Maeno, Mandy Yang

CERN cluster: Ricardo Rocha

UVic clusters: Danika MacDonnell, Rolf Seuster, Ryan Taylor

Backup slides

Harvester - Kubernetes integration



Container image configured per queue (for ATLAS: CentOS7 image). Needs to be improved

Concrete work items

- Image management: automated generation and distribution
- Benchmarking efficiency penalties from containers
- Cluster setup recipes: kubespray, ansible, magnum
  - Including monitoring
- APEL accounting
- Pilot-Harvester
  - Extract container image and executable, fill out executable template
  - Stage-in and stage-out model
  - Logfile management
  - Error code and message
  - Memory monitoring
  - Loop detection
  - Improved usage of K8s API
- Authentication/authorization model
- Federation
- Helm chart for Harvester and PanDA servers