

Towards a Kubernetes-native T2 at UVic

Ryan Taylor



**University
of Victoria**

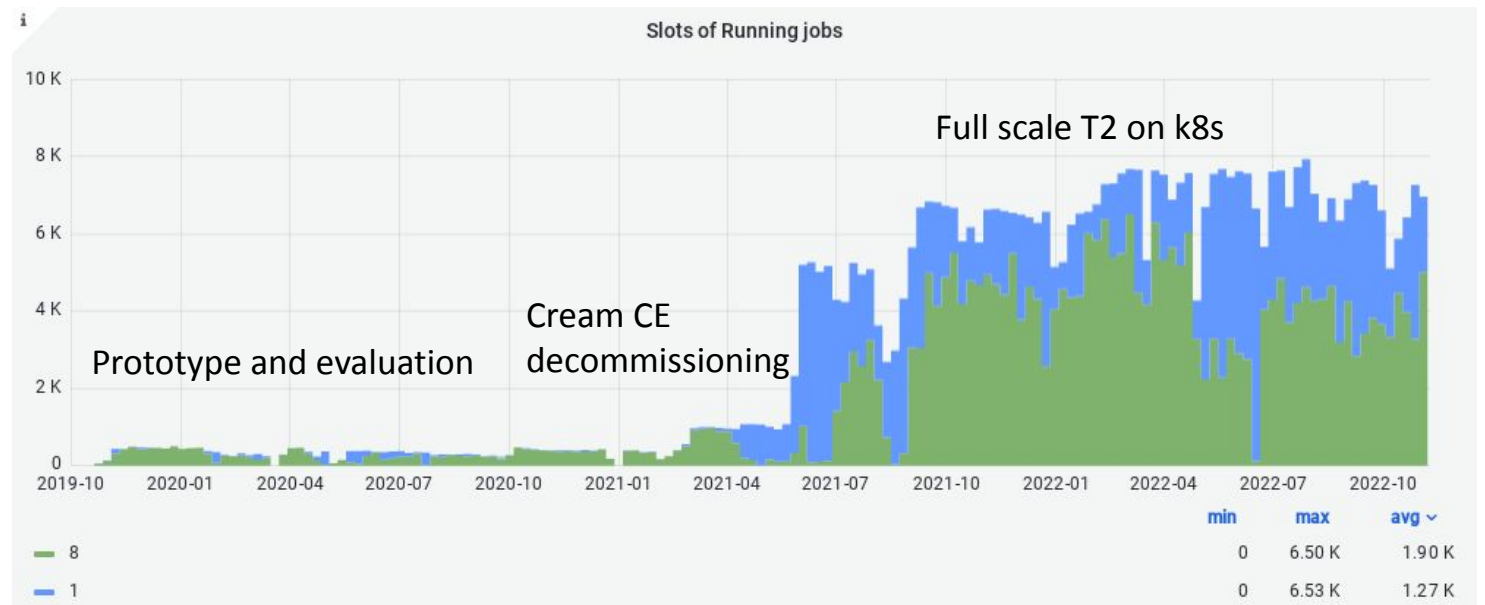


Background

CA-VICTORIA-WESTGRID-T2 uses k8s for container-native batch computing. Harvester submits ATLAS grid jobs to k8s API, which runs them as pods. No traditional batch system or CE.

Prior talks on UVic k8s T2

- 2019 Nov [CHEP](#)
- 2019 Dec [pre-GDB](#)
- 2020 Dec [k8s HEP meetup](#)
- 2020 Dec [WFM SW TIM](#)
- 2021 May [ADC TCB](#)
- 2022 June [pre-GDB](#)



Why cloud and why Kubernetes?

UVic site background

Physical

Virtual

batch

cloud




- Bare metal batch cluster and WLCG ATLAS T2 commissioned
 - Serial & parallel partitions, GbE & IB networks2010
- Serial cluster expansion 2011
- Cloud funding, dedicated hardware 2012
- first national cloud service offering 2014
- major national cloud site 2016
- Cloud hardware expansion 2018
- Cloud hardware expansion 2019
- Cloud hardware expansion 2020
- Cloud hardware expansion 2021

- cloud technology experimentation
 - Nimbus, OpenNebula, Oracle cloud
- Synnefo: Nimbus cloud
- virtualized batch cluster in cloud
- Nephos/West: OpenStack cloud
- Arbutus: OpenStack cloud site
- Kubernetes experimentation
- CA-VICTORIA-K8S-T2 in production
 - gaining k8s experience for ATLAS
- T2 compute entirely k8s-native
 - CREAM CE decommissioning



Why cloud and why Kubernetes?

- Running compute jobs is “easy”
- Running long-lived services is hard
 - Ongoing management, updates, configuration changes
- Doing it robustly is harder
 - Availability, redundancy
- Cloud + k8s provides:
 - Flexible & dynamic infrastructure
 - Resilience and automated remediation
 - Rapid application deployment
 - Application lifecycle management
 - Horizontal scalability

	VMs as pets	Openstack
	VMs as cattle	Openstack + ???
	containers as cattle	Openstack + k8s

The eventual goal: a fully k8s-native T2

Installable with Helm

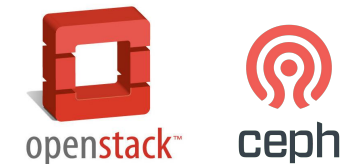
- Helm: ~~package~~ application manager for Kubernetes
 - One command to install/upgrade everything
 - Comprehensive configuration via one YAML file
- **helm install T2Site**
 - (K)APEL accounting DONE 2021
 - frontier-squid DONE Sep 2022
 - EOS SE further work needed
 - compute (security rules, Harvester setup) done (but not via Helm yet)
 - CVMFS-CSI optional
 - ~~Compute Element~~ built-in
 - ~~Batch system~~ built-in

Considerations for EOS SE on k8s with CephFS



Currently: dCache (2.7 PiB) bare metal storage servers. Why change?

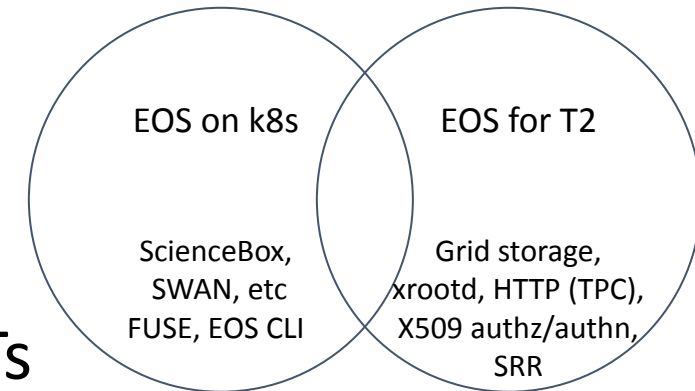
- Physical consolidation: all storage on Ceph
- Logical consolidation: services on k8s
- EOS can be installed on k8s via Helm chart
 - reproducible, single step deployment
 - easier to manage and maintain
 - easy to set up another instance, e.g. for dev
- EOS + CephFS is an established solution
- Opportunity: [direct data access for jobs](#) on CephFS



Challenges deploying EOS SE on k8s with CephFS

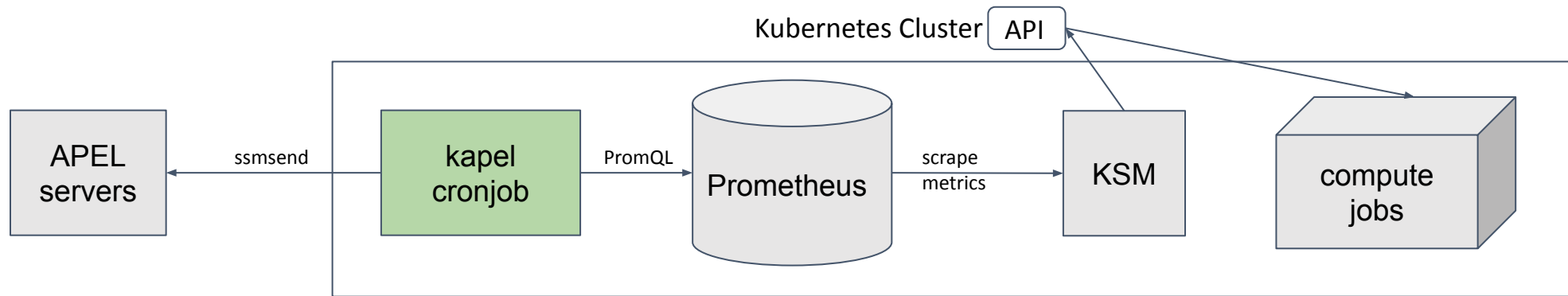


- Expansion/development of Helm chart needed
 - e.g. for X509 authz/authn [#74](#) [#75](#)
- CephFS bug encountered: [55090](#)
 - Ceph fixes: [#46902](#) [#46905](#)
- Would be nice to have 1 Ceph share used by all FSTs
 - Easier to scale up storage
 - FST unification?
- PureLB: integrate with Calico BGP to scale bandwidth > 1 NIC
- Performance to be evaluated/benchmarked



KAPEL

Container-native APEL accounting for Kubernetes



- Standard k8s add-ons do most of the work
 - kube-state-metrics (KSM) instead of batch log parser
 - Prometheus instead of MySQL DB for data collection and storage
 - PromQL for data querying, analytics
 - k8s cron job instead of APEL node
 - Only needed to write ~200 lines of python (and some YAML)
- Recently added support to easily [publish manual corrections](#)

Frontier-squid

Deployed on Kubernetes

- Chose ScienceBox frontier-squid [Helm chart](#)
 - Simple, lightweight, container-native approach
 - Trivial to scale to N instances with automatic load-balancing and failover
- UVic contributed enhancements
 - Run as unprivileged squid user [#61](#)
 - Allow configuration of service details [#63](#)
 - Support for priorityClass and pod resource requests/limits [#64](#)
 - Send access logs to stdout [#69](#)
 - Configurable ACL activation [#72](#)
 - Harmonize configuration with upstream package [#73](#)
 - Add backup readiness probe URL for redundancy [#74](#)
 - Update ACLs for Frontier servers [#78](#)
 - Expand list of safe ports [#81](#)
- TODO: support for CVMFS proxy sharding [#97](#)
- But difficult/impossible to use WLCG SNMP monitoring
 - Hardcoded to use only port 3401
 - Multiple load-balanced squids behind one IP

Questions/discussion

Ingress and LBaaS

- Initial basic approach used keepalived and nginx-ingress to receive traffic from outside world into clusters
- Migrated to PureLB and Traefik
 - More maintainable/manageable, via Helm charts
 - Cohesive access to dashboards etc across all clusters
- PureLB: like MetalLB but simpler, lightweight
 - relies on Linux network stack of host
 - Programmable (LB -> LBaaS)
- Traefik Ingress controller
 - Widely used, full featured, nice web UI, CRDs
 - Better TCP and UDP support



Misc. improvements

- Switch from CentOS8 to AlmaLinux 8 (needed GPT partition table)
- Switch from Docker to containerd
- Install metrics-server for node/pod resource monitoring (`kubectl top`)
- Using full-node VMs (80 cores, 360 GB RAM)
 - fewer VMs, better disk IO
- Define priorityClass for everything to avoid resource contention
 - also resource requests/limits as much as possible
- Calico scalability/efficiency
 - use Typha to reduce load on API servers for large clusters
 - disable IPIP encapsulation within cluster to reduce network overhead