Towards a Kubernetes-native T2 at UVic

Ryan Taylor

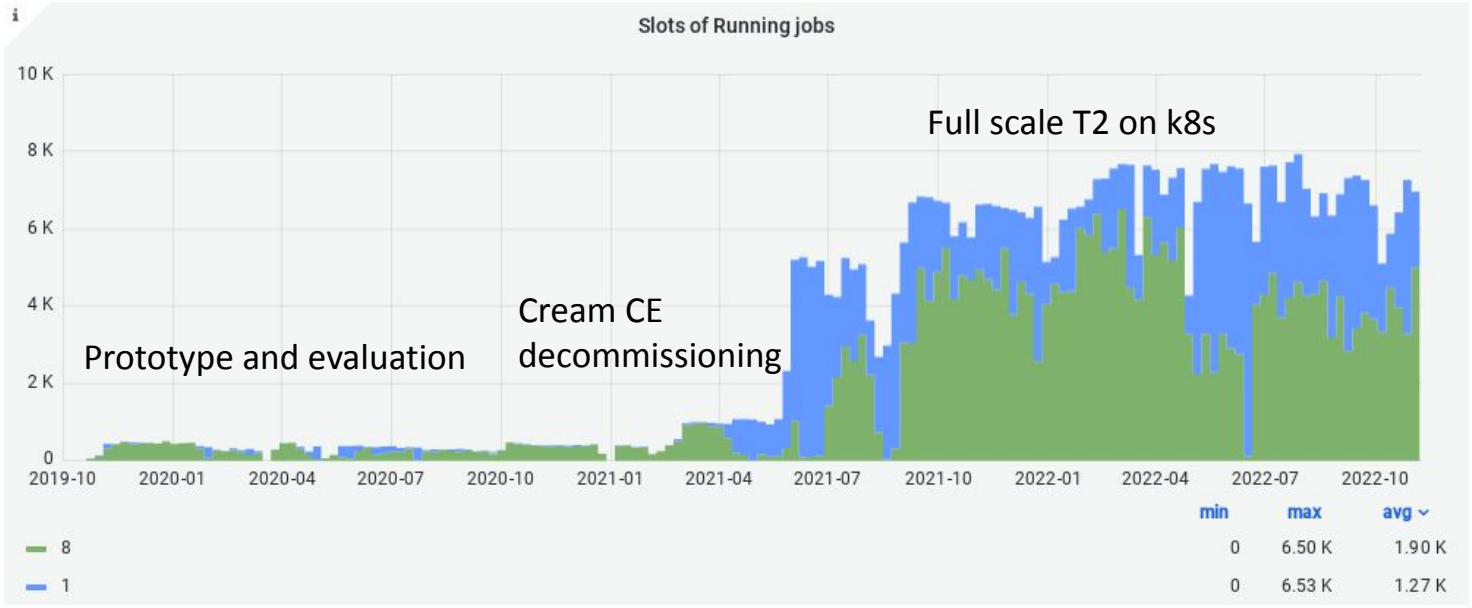
University of Victoria

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
- 2022 Nov WLCG workshop



Arbutus Cloud

- General-purpose science cloud
- ~3K users, 1K projects
- 44,000 vCPUs
- 416 vGPUs
- 17 PB (usable) Ceph
  - RBD, object, CephFS

openstack™

ceph

Why cloud and why Kubernetes?

UVic site background

	Physical		Virtual

batch	• Bare metal batch cluster and WLCG ATLAS T2 commissioned
	  ◦ Serial & parallel partitions, GbE & IB networks
		2010	• cloud technology experimentation
			  ◦ Nimbus, OpenNebula, Oracle cloud

	• Serial cluster expansion	2011	• Synnefo: Nimbus cloud
		2012	

cloud	• Cloud funding, dedicated hardware
	  ◦ first national cloud service offering
	  ◦ major national cloud site
		2014
		2016
			• virtualized batch cluster in cloud
			• Nephos/West: OpenStack cloud
			• Arbutus: OpenStack cloud site

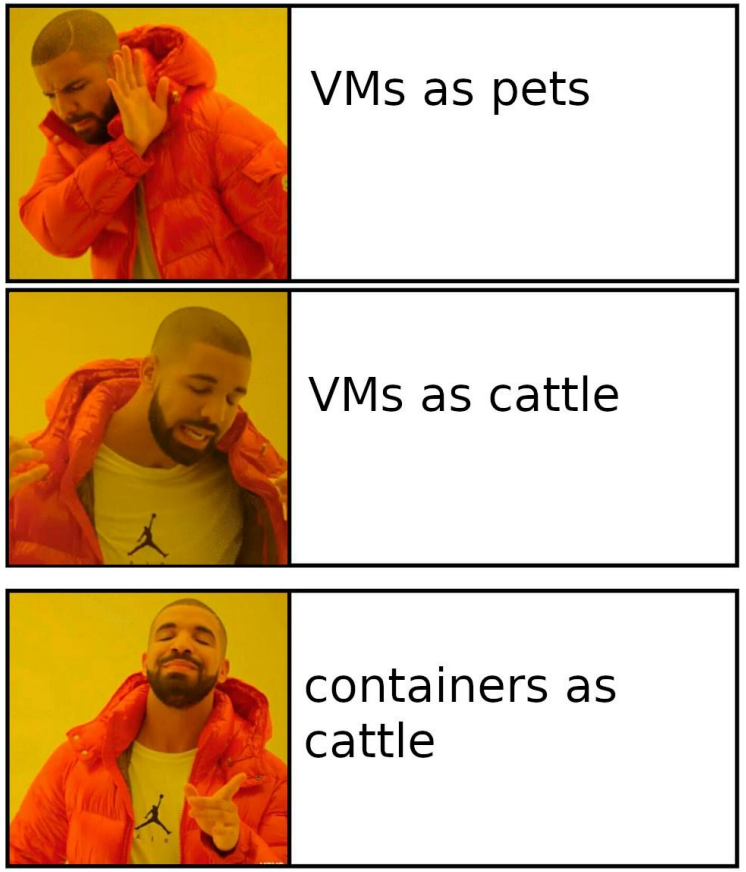
	• Cloud hardware expansion	2018	• Kubernetes experimentation
		2019	• CA-VICTORIA-K8S-T2 in production ◦ gaining k8s experience for ATLAS
	• Cloud hardware expansion	2020	• T2 compute entirely k8s-native ◦ CREAM CE decommissioning
		2021	

WESTGRID

University of Victoria

Why cloud and why Kubernetes?

- Running compute jobs is “easy”
- Harder: statefulness (storage), long-running services
  - Ongoing management, updates, configuration changes
- Doing it robustly is even harder
- Cloud + k8s provides:
  - Flexible & dynamic infrastructure
  - Resilience and automated remediation
  - Rapid application deployment
  - Application lifecycle management
  - Horizontal scalability



Openstack

Openstack + ???

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

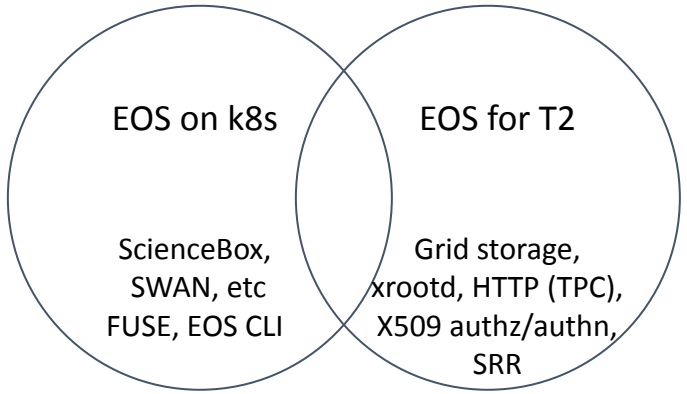
Considerations for EOS SE on k8s with CephFS

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

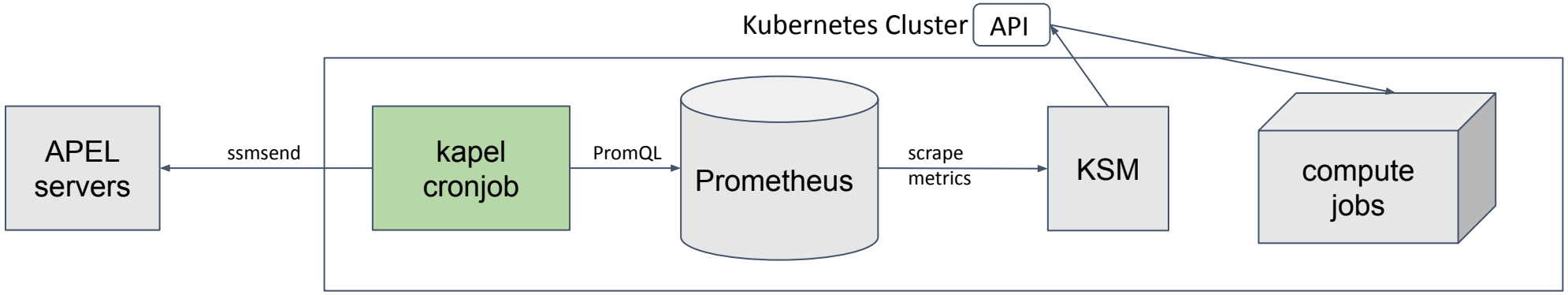
- Expansion/development of Helm chart needed
  - e.g. for X509 authz/authn #74 #75
- CephFS bug encountered: 55090
  - Ceph fixes: #46902 #46905
- Need to scale cluster ingress bandwidth > 1 NIC
  - integrate PureLB with Calico BGP instead of ARP mode
- Benchmarking and performance tuning



https://github.com/rptaylor/kapel
available via Helm chart

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)

Many thanks to Enrico Bocchi !

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
- But difficult/impossible to use WLCG SNMP monitoring
  - Hardcoded to use only port 3401
  - Multiple load-balanced squids behind one IP - need different ports

CVMFS proxy sharding with k8s Squids

- New feature in CVMFS v2.10 to improve cache hit rates
- CVMFS understands round-robin DNS
  - dereferences multiple A records
- Solution using k8s Services: headless ClusterIP

```
service:
  clusterIP: None
```

- Should decrease CVMFS_DNS_MIN_TTL to a small value
  - CVMFS default is 1 min
  - K8s deployment upgrade could be < 1 min (and DNS TTL is 5 s)
  - Details: #97

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