Kubernetes as an ATLAS Computing Site

Pre-GDB - Kubernetes Many Faces

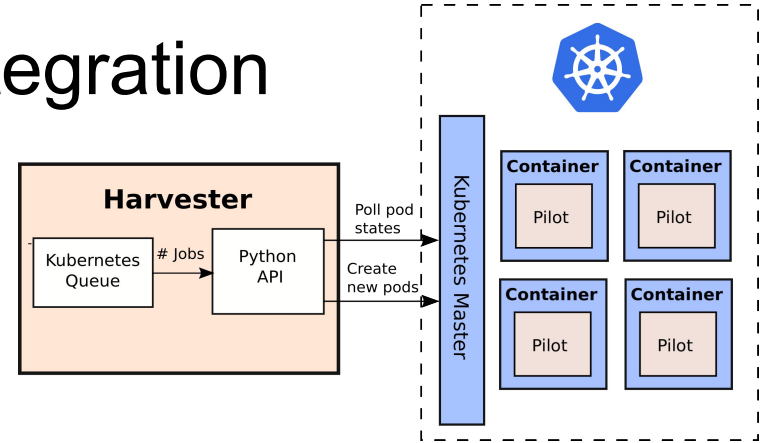
Dec. 10, 2019

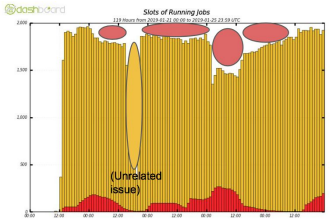
Fernando Barreiro Megino[1], Frank Berghaus, Danika MacDonell, Rolf Seuster[2], Ryan Taylor on behalf of the ATLAS experiment

[1] Technical Supervisor
[2] Local Supervisor

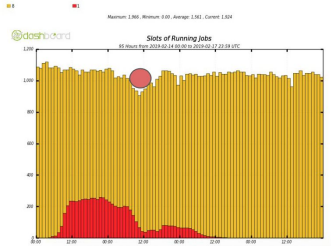
Background: Harvester k8s Integration

- Kubernetes (k8s) is a popular container orchestration tool.
- Using the k8s python API, Harvester functionality added by FaHui Lin and Mandy Yang (Academia Sinica) to manage a k8s container cluster as a computing site.
- Dedicated containers are launched to run jobs.





With default K8s scheduler (round robin load balance)



With policy tuning to pack nodes

- Tested at scale at CERN with ~1000 cores (early 2019)
- Conclusions:
  - Custom scheduling needed
  - Need to improve/automate node management.

Kubernetes Site Testing (CA-IAAS-T3-K8S)

- **Goal:** Test/improve the functionality and scalability of remote k8s cluster as an ATLAS computing site.

- Dedicated harvester node at CERN
- k8s cluster comprised of 20 8-core VMs on Compute Canada Arbutus cloud (UVic) successfully ran production jobs (mostly) stably for ~2 months.
- Automated k8s cluster setup with terraform + kubespray + custom ansible scripts.
  - Few prerequisites for kubespray setup → advantageous for remote sites
- CVMFS currently mounted via host
  - Have successfully run with parrot on a baremetal cluster (Rolf)

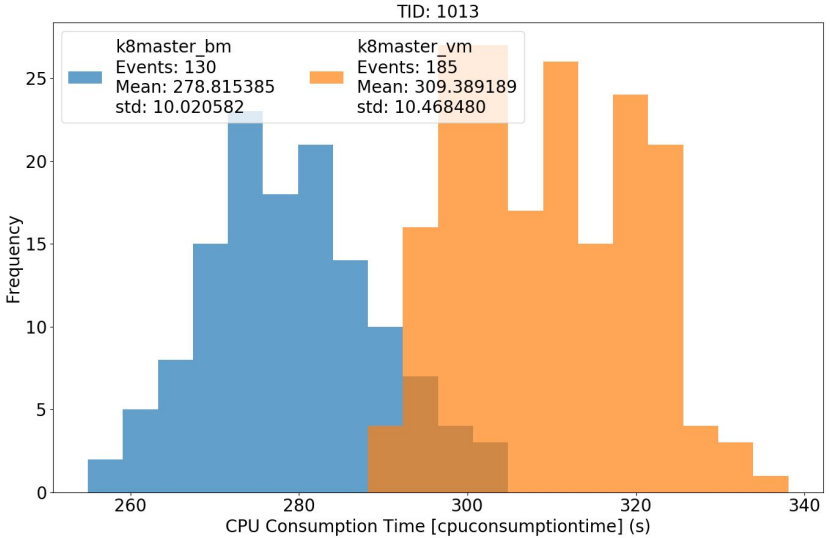
Automated Kubespray Build

- Construction of multi-node kubernetes cluster automated with kubespray
- Terraform script automates provisioning of openstack nodes for k8s cluster.
- Two settings: build cluster or scale it up.

- Additional shell and ansible scripts to:
  - Install dependencies
  - Prevent scheduling on master nodes (found that scheduling+running jobs can crash node)
  - Install cvmfs on each node
  - Set up local docker registry → container image to run jobs downloaded only once over public network, then distributed to other nodes via private network
  - Enable ssh connection from Harvester
    - ssh tunnel currently used as quick solution for connecting to VM master with floating IP

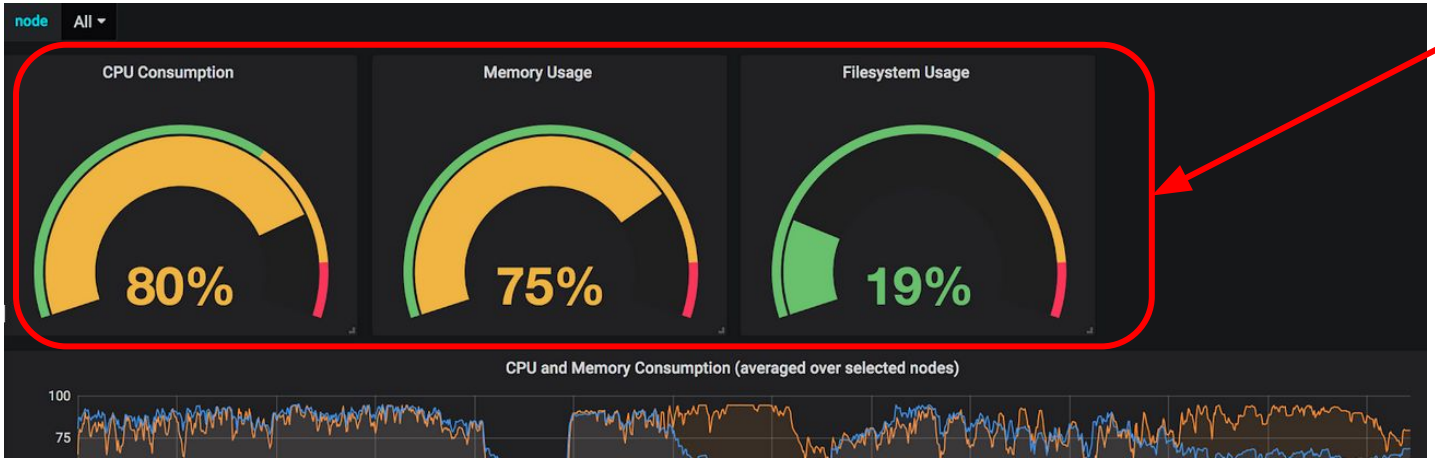
VM vs. Baremetal Comparison

- First project: compare performance of jobs running on baremetal vs. VM k8s node.
- Collected PanDA job metrics for jobs running on a bar metal 16-core IBM, vs. 16-core VM launched on the same machine.
- ~11% performance loss in cpuconsumptiontime seen for jobs running on VM compared with baremetal
- Results motivate future investigations into baremetal alternatives to VM cluster (eg. Ironic).



Internal Cluster Resource Monitoring & Storage

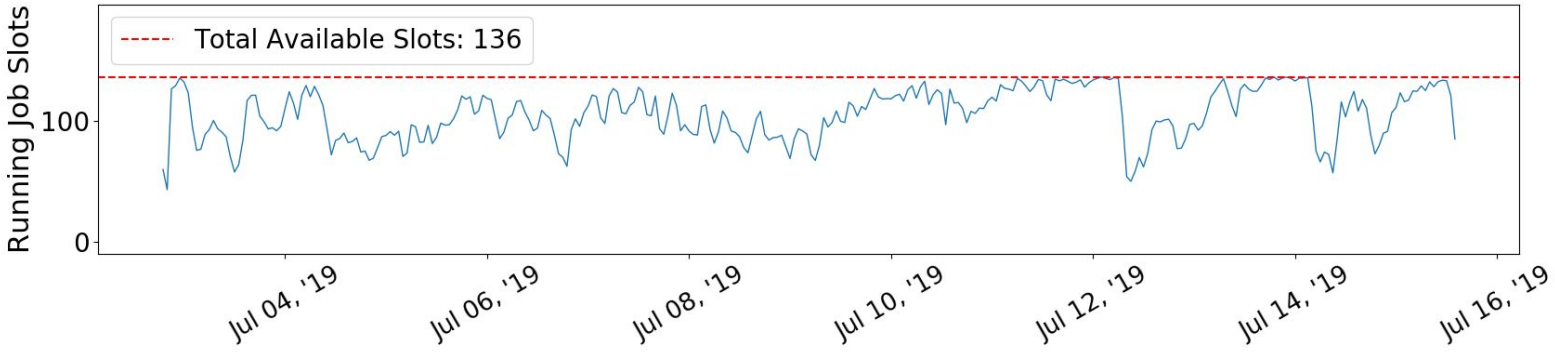
- Adapted a metricbeat setup to send k8s cluster resource metrics to a local elasticsearch database for long-term storage and monitoring.
  - CPU consumption
  - Memory usage
  - Network and filesystem usage
- Metrics visualized in real time with Grafana.



Overall resource consumption over given period for selected nodes (all)

Node-Packing Scheduler

- Node occupancy was varying erratically and often far below cluster capacity.



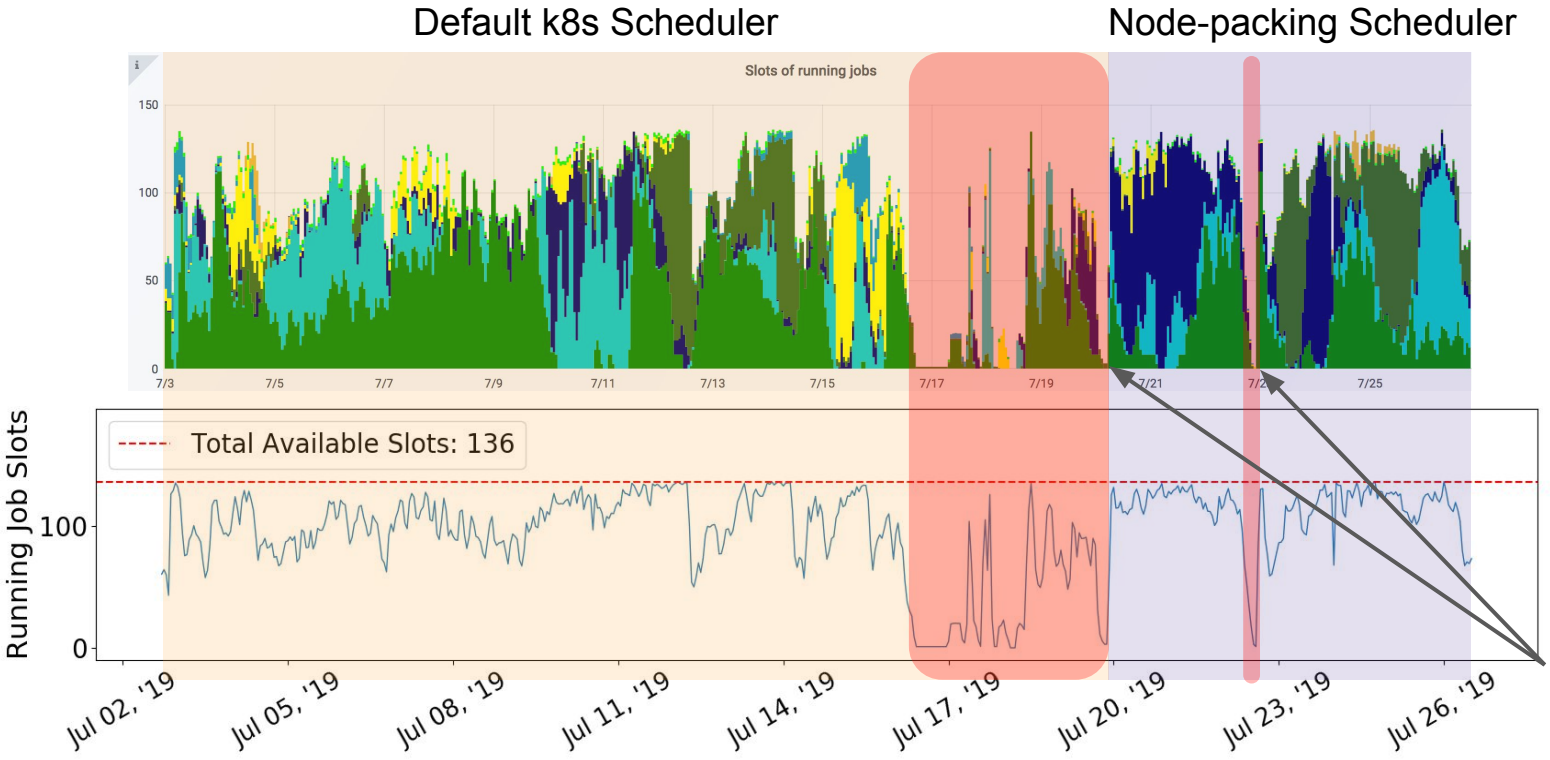
Node-Packing Scheduler

- Fernando suggested switching from default k8s scheduler to custom node-packing scheduler documented in the harvester-k8s docs to improve node occupancy.

- Node-packing scheduler avoids spreading 1-core jobs across nodes, which can prevent scheduling of 8-core jobs.

- Implemented as custom kube-scheduler pod, as documented here.

Node-Packing Scheduler



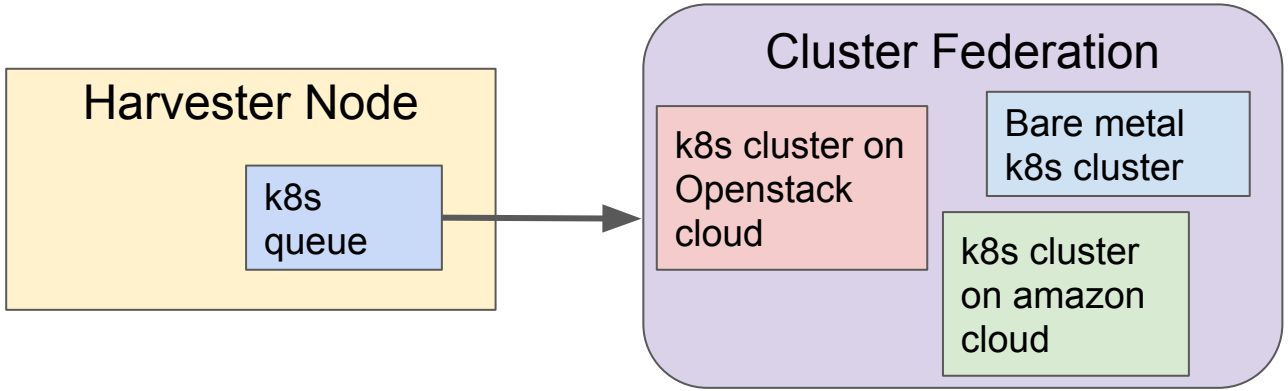
Unrelated Harvester issues

Cluster Federation

Motivation: Centrally manage submission of production jobs to multiple k8s clusters

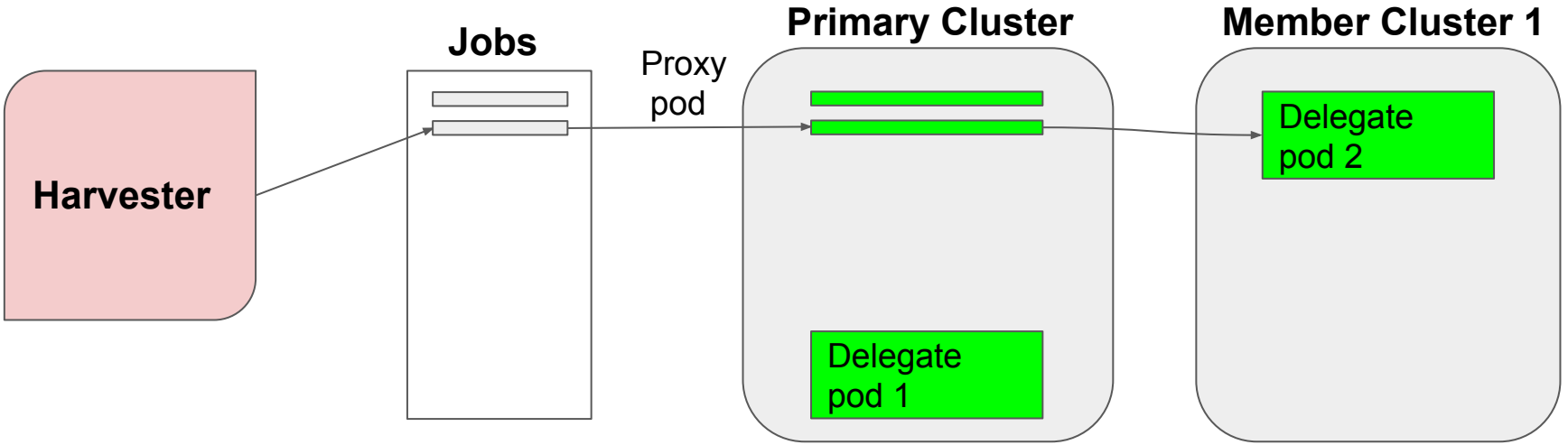
- Would allow to combine job submission to eg. multiple small k8s clusters rather than managing 1 queue per cluster
- Federated k8s clusters could be on completely different networks and clouds

Idea: Harvester “sees” k8s resources as a single queue, but queue gives access to resources in multiple federated k8s clusters.



Multicluster Scheduler

- Initially looked into KubeFed v2 for multicluster federation
  - No clear functionality for scheduling jobs across multiple clusters
- Have been investigating [“Multicluster Scheduler”] developed by admiralty as alternative
  - Jobs submitted to ‘primary’ cluster can be scheduled as ‘delegate pods’ on any of the member clusters



Multicluster Scheduler

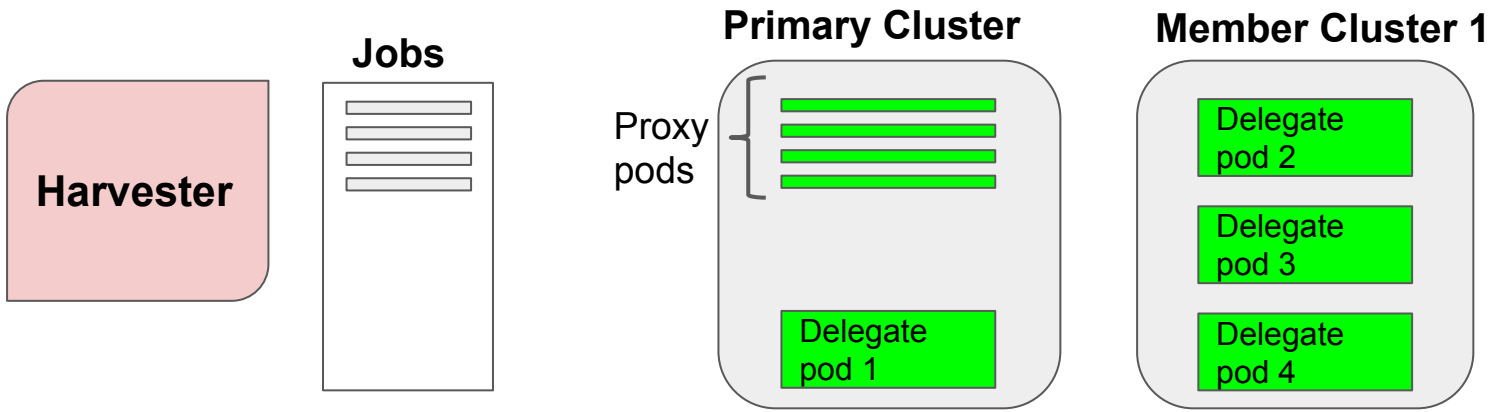
- Initially looked into KubeFed v2 for multicluster federation
  - No clear functionality for scheduling jobs across multiple clusters
- Have been investigating "Multicluster Scheduler" developed by admiralty as alternative
  - Jobs submitted to 'primary' cluster can be scheduled as 'delegate pods' on any of the member clusters
  - Only change needed on Harvester side is an annotation added to the job's yaml file
  - Integrates seamlessly with custom node-packing scheduling on member clusters
- Tested multicluster scheduler setup on two openstack k8s clusters
  - 2 8-core worker nodes per cluster ⇒ **32 cores** total
  - Harvester submitting production jobs to one queue shared by the two k8s clusters

Multicluster Scheduler: Minor Issues

- Need cron job on ‘primary’ k8s cluster to regularly distribute k8s secret containing grid proxy to other ‘member’ clusters
- Some modifications needed to ‘out-of-the-box’ multicluster scheduler deployments to:
  - Schedule deployment pods on master nodes rather than worker nodes (deployment pod cpu requests could prevent scheduling of 8-core jobs on workers)
  - Increase memory requests (some deployment pods were failing due to insufficient memory)
    - Should be much cleaner to specify memory in the latest release of multicluster scheduler that came out a few days ago

Multicluster Scheduler: Proxy Pod Overscheduling

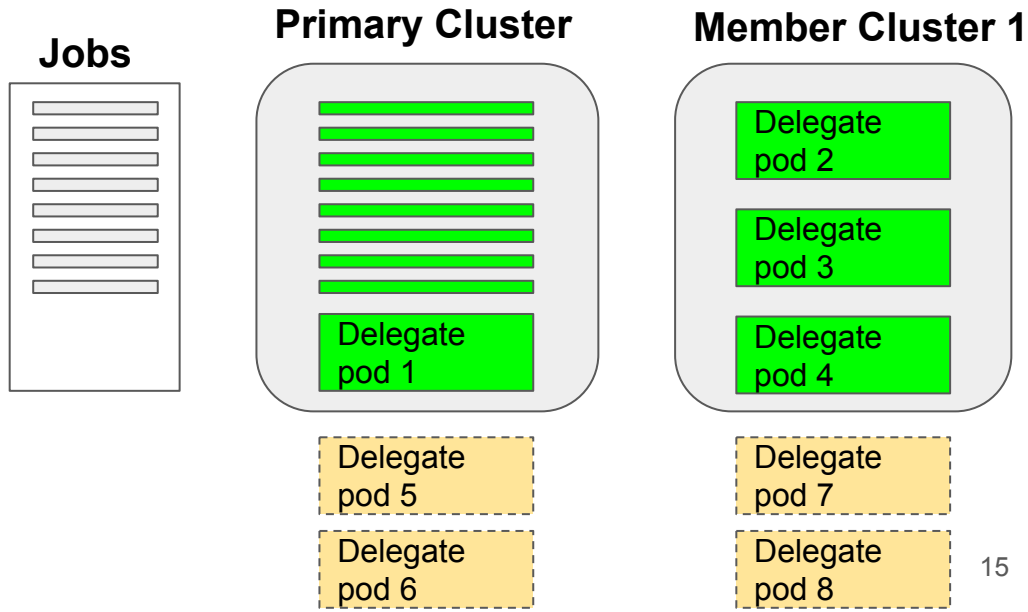
- Multicluster-scheduler replaces the elected pods with ‘proxy pods’ on the primary cluster and deploys ‘delegate pods’ to member clusters to run jobs



Multicluster Scheduler: Proxy Pod Overscheduling

- Multicluster-scheduler replaces the elected pods with ‘proxy pods’ on the primary cluster and deploys ‘delegate pods’ to member clusters to run jobs

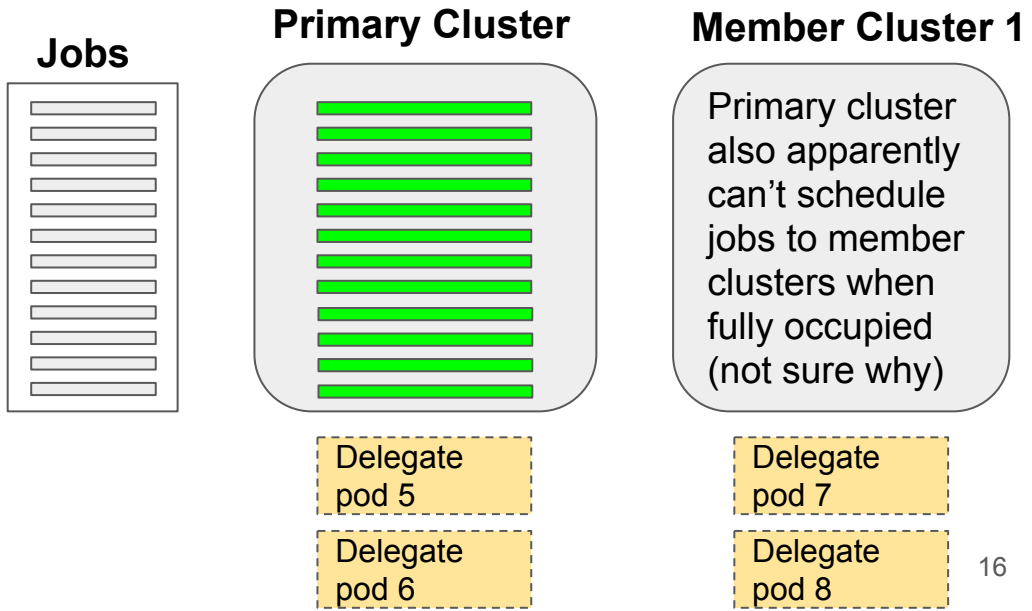
- **Problem:**
  - Proxy pods request no memory or cpu, so start running immediately
  - Harvester sees running proxy pods, so submits more jobs



Multicluster Scheduler: Proxy Pod Overscheduling

- Multicluster-scheduler replaces the elected pods with ‘proxy pods’ on the primary cluster and deploys ‘delegate pods’ to member clusters to run jobs

- **Problem:**
  - Proxy pods request no memory or cpu, so start running immediately
  - Harvester sees running proxy pods, so submits more jobs
  - Primary cluster reaches max allowable running pods per node with proxy pods and can’t schedule delegate pods to run jobs

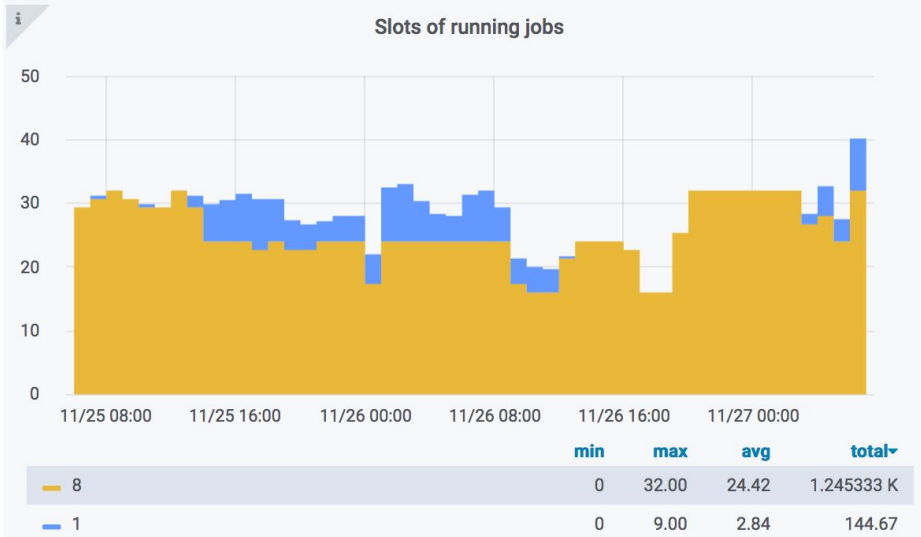


Multicluster Scheduler

- Current solution to proxy pod overscheduling:
  - Modify multicluster scheduler to schedule proxy pods to dedicated nodes (not worker nodes)
  - Proxy pods can now fill dedicated nodes to max pod capacity without affecting delegate pod scheduling on worker nodes

- Test clusters running production jobs near full 32 core capacity with current setup

- Developer is interested in pursuing more general solutions to proxy pod overscheduling using kube-batch or virtual kubelet



Summary and Conclusions

- Performance gains of baremetal k8s worker node over VM node motivate future investigations of baremetal alternatives to VM clusters

- Custom node-packing scheduler found to improve node occupation on test cluster
- Multicluster scheduler shows potential for central management of multiple k8s clusters
- Code and setup instructions for testing clusters documented on gitlab

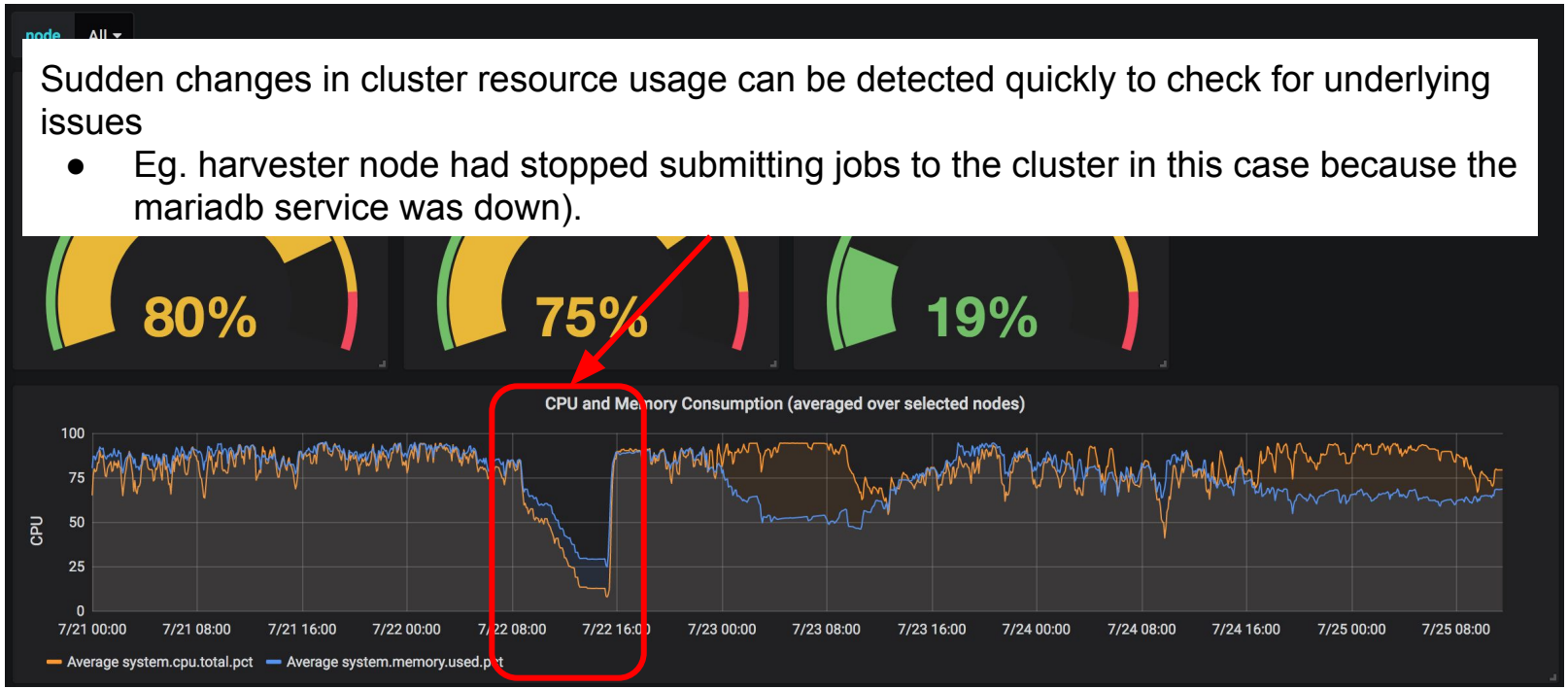
- Ryan Taylor and Jeff Albert (UVic) have launched a k8s cluster at UVic as a T2 production site - setup similar, with additional security+scalability features
  - Running stably with 500 cores

Backup Slides

Internal Cluster Resource Monitoring & Storage

Monitor resource usage for individual nodes to check for disparities.

Internal Cluster Resource Monitoring & Storage



Sudden changes in cluster resource usage can be detected quickly to check for underlying issues

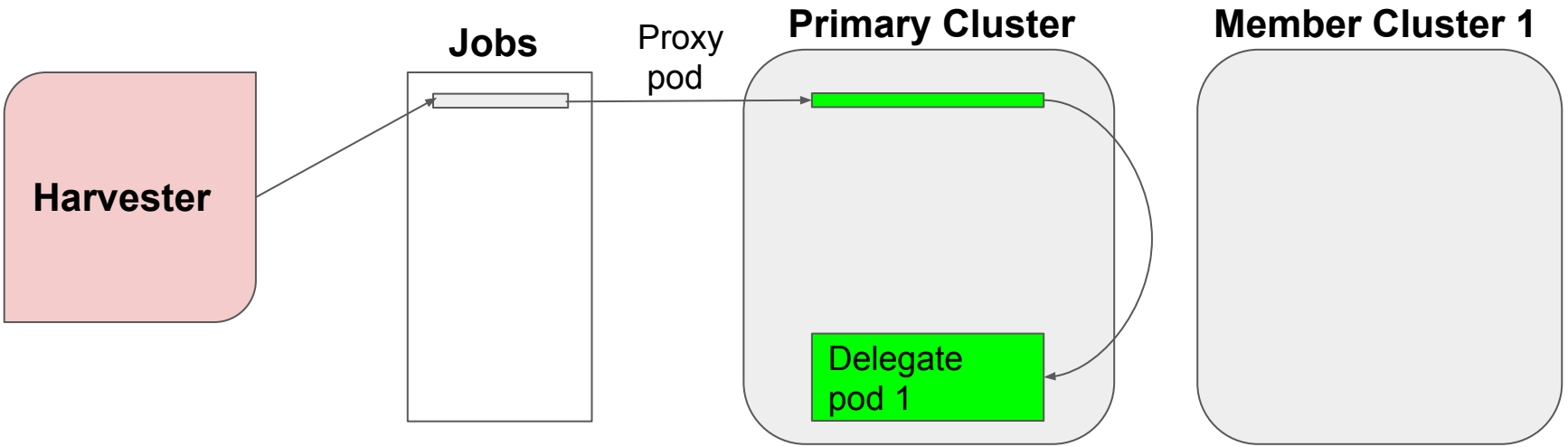
- Eg. harvester node had stopped submitting jobs to the cluster in this case because the mariadb service was down).

Updates on Harvester Side

- Running multiple k8s clusters on the same harvester node
  - Initially found that doing so led to conflicts (i.e. configuration files, etc. mixed up between different k8s queues).
  - Fernando identified the issue causing this behaviour and implemented a fix for it in the harvester code.

Multicluster Scheduler

- Initially looked into KubeFed v2 for multicluster federation
  - No clear functionality for scheduling jobs across multiple clusters
- Have been investigating "Multicluster Scheduler" developed by admiralty as alternative
  - Jobs submitted to 'primary' cluster can be scheduled as 'delegate pods' on any of the member clusters



CA-VICTORIA-K8S-T2

Ryan Taylor and Jeff Albert (UVic computing specialists) have deployed a production k8s cluster with additional features, including:

- Ability to connect directly to cluster API via public IP
  - no need for ssh tunnel
- Enhanced security features
  - X509 cert (issued by k8s cluster CA) required to authenticate
  - Role Based Access Control restricts which API actions are allowed to Harvester
- Containers launched with CVMFS Docker Graph Driver plugin for scalability
  - Docker loads data on demand from CVMFS instead of pulling entire image
  - Significant startup acceleration and bandwidth savings

CA-VICTORIA-K8S-T2

- Started running production jobs for T2 Oct 25
- 500 cores running stably since Nov. 13