
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

Best Practices: Improving Batch Scheduling Performance at Scale Using MCAD and KWOK

Vishakha Ramani & Sara Kokkila-Schumacher, IBM Research

Improving Batch Scheduling Performance at Scale Using MCAD and KWOK

Talk Outline

Motivation and existing challenges

Potential Solution: KWOK (Kubernetes WithOut Kubelet)

Getting Started with KWOK

Things to look out for

Example Use Cases


KubeCon | CloudNativeCon

North America 2023

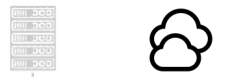
Motivation and Existing Challenges

Foundation Model end-to-end lifecycle

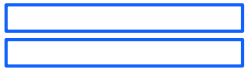
Model creation ... ... deployment

Data preparation

Workflow of steps (e.g. remove hate and profanity, deduplicate, etc)

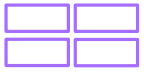


Distributed training



Long-running job on massive infrastructure

Model adaptation



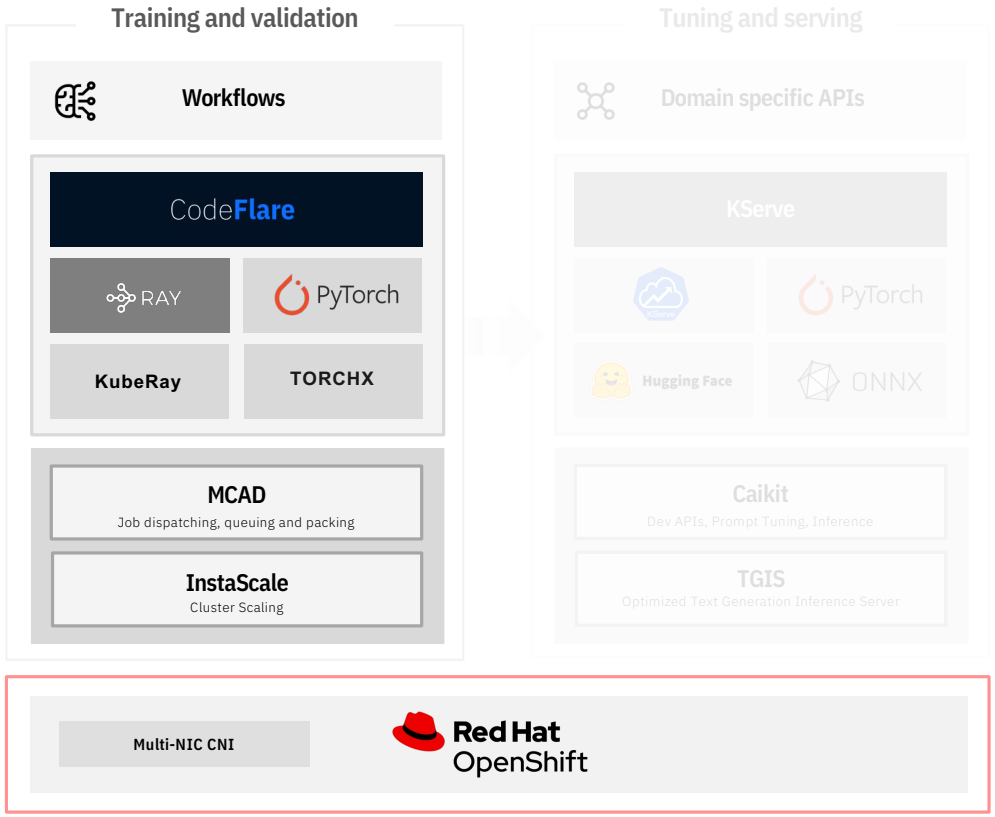
Model tuning with custom data set for downstream tasks

Inference

May have sensitivity to latency, throughput, power, etc.

Foundation Model Stack for Training and Validation

KubeCon CloudNativeCon North America 2023



Midstream delivery in Open Data Hub

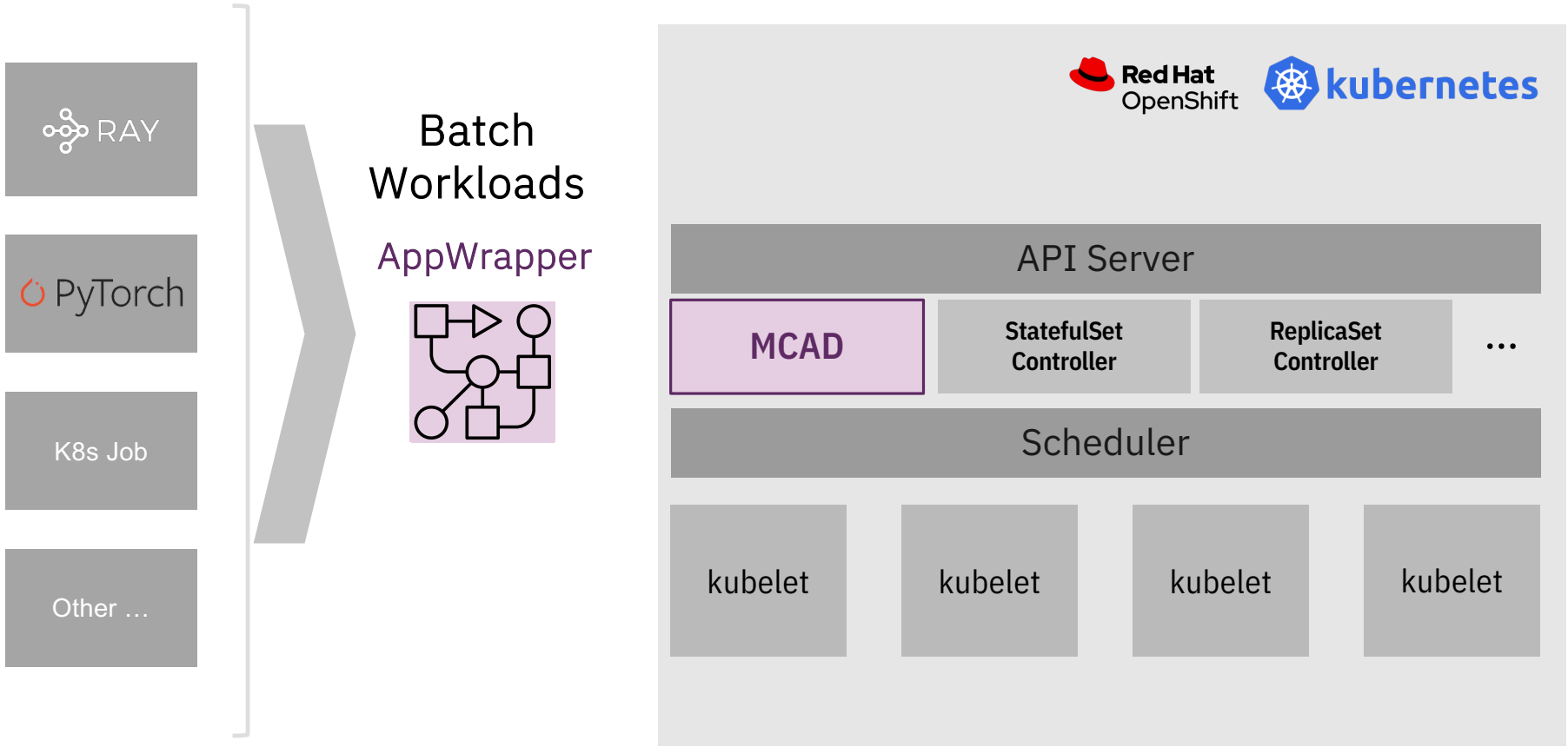
self-managed, supported platform

OPEN DATA HUB
AI Platform powered by Open Source

OpenShift AI

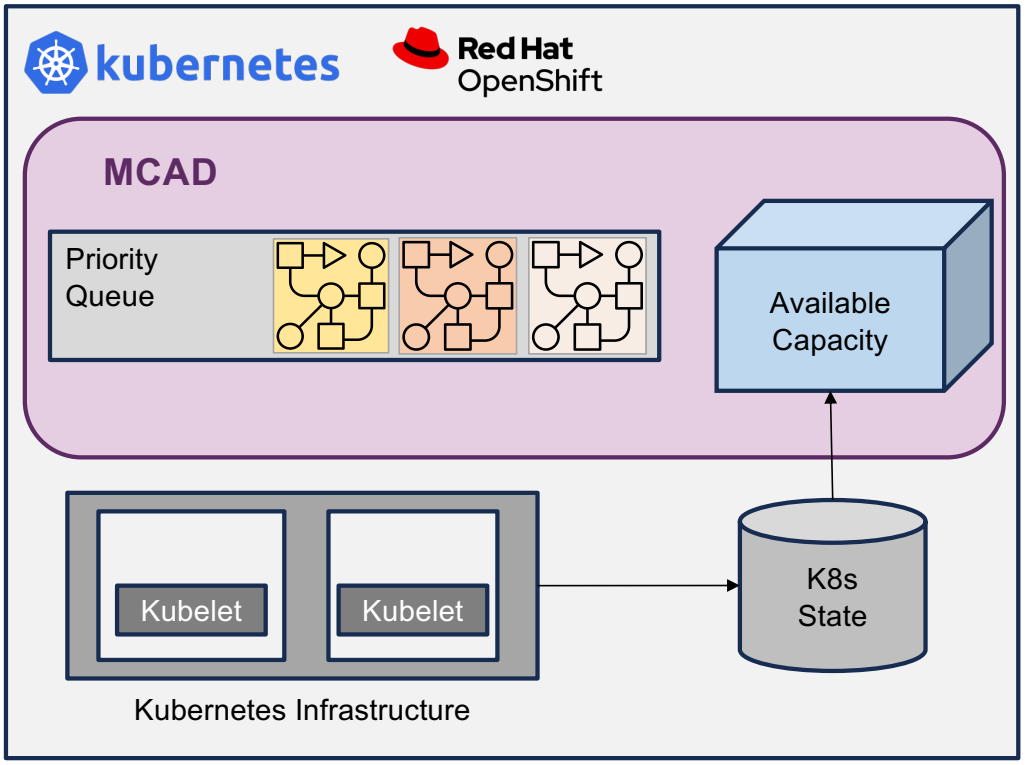
https://research.ibm.com/blog/openshift-foundation-model-stack

Multi-Cluster App Dispatcher (MCAD)



https://github.com/project-codeflare/multi-cluster-app-dispatcher

MCAD and AppWrappers



AppWrapper pseudocode

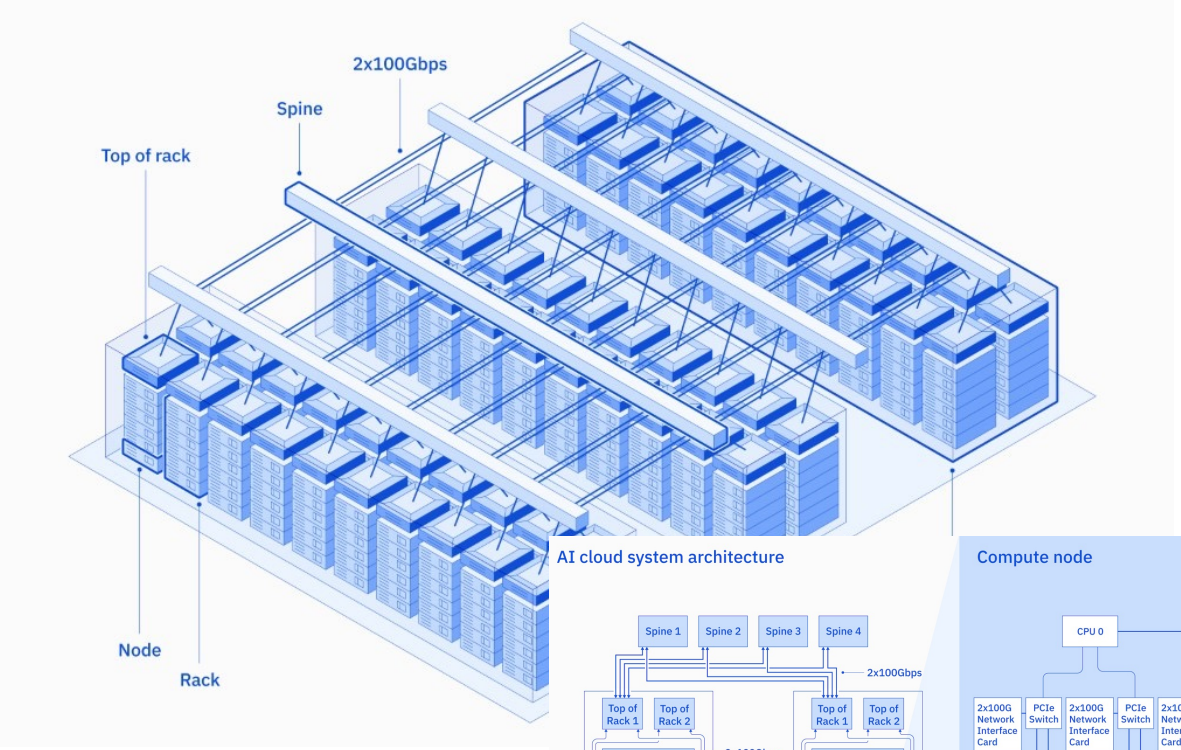
```
apiVersion: workload.codeflare.dev/v1beta1
kind: AppWrapper
metadata:
  name: my-example-appwrapper
  namespace: default
spec:
  resources:
    GenericItems:
      PodGroup
      batch/v1
      Job
```

https://github.com/project-codeflare/multi-cluster-app-dispatcher

Deploying at Scale

System specifications

- Nodes with 8 x A100 GPUs (80GB)
- GPUs interconnected with NVLink, NVSwitch
- Cascade Lake CPUs, 1.5TB of DRAM,
- Four 3.2TB NVMe drives
- Redundant connections between nodes, TORs and spines
- 2 x 100G NICs from each node – NCCL benchmarks show we drive close to line rate

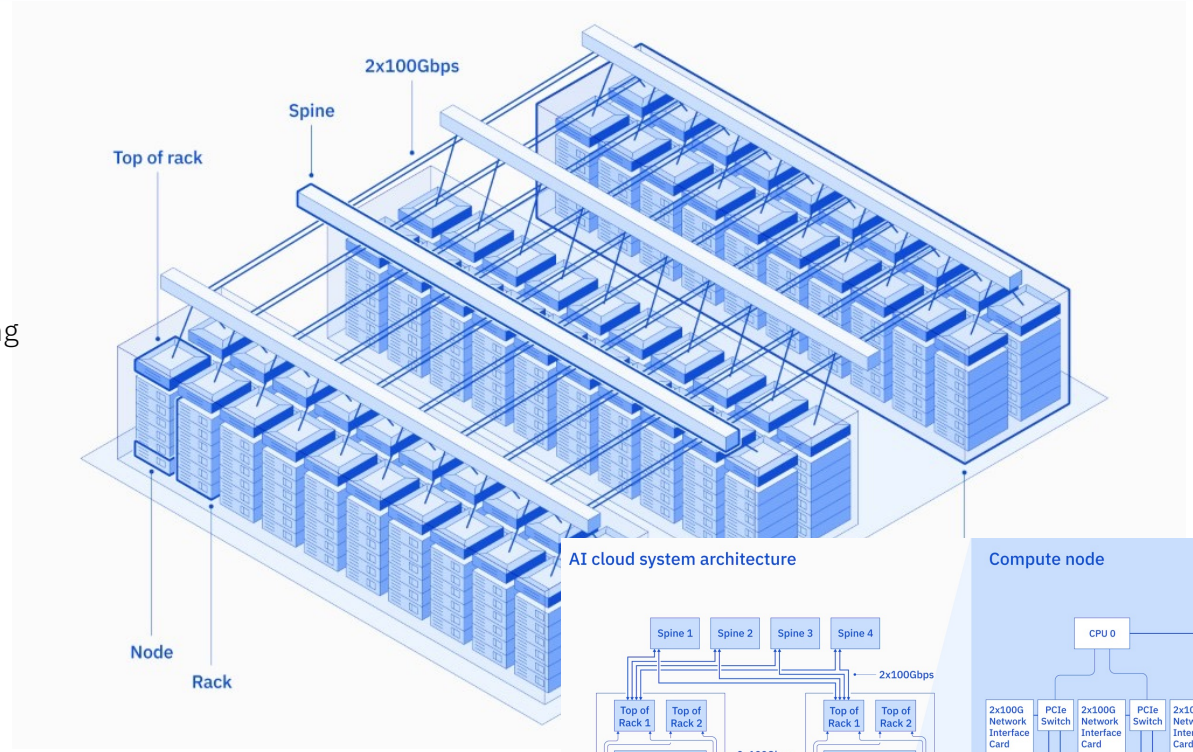


https://research.ibm.com/blog/AI-supercomputer-Vela-GPU-cluster

Resource Management Challenges

Managing large training jobs

- Gang scheduling
- Queue large training jobs and dispatch as resources become available, including just-in-time object creation
- Priority, pre-emption, and retry support
- Quota management



https://research.ibm.com/blog/AI-supercomputer-Vela-GPU-cluster


KubeCon | CloudNativeCon

North America 2023

Potential Solution: KWOK

KWOK: Kubernetes WithOut Kubelet

- Toolkit to create and simulate pods and nodes to behave like real ones with low resource footprint.
- Key Tools in KWOK
  - **kwok**: Simulates lifecycle of fake nodes and pods, and other Kubernetes API resources.
    - kwok in cluster - Installing kwok in a cluster
    - kwok out of cluster - Run kwok out of your cluster
  - **kwokctl**: CLI tool for cluster creation
    - kwokctl clusters - Create/Delete a cluster where all nodes are managed by kwok
    - kwokctl snapshots cluster - Save/Restore the Etcd data of a cluster created by kwokctl

https://github.com/kubernetes-sigs/kwok

KWOK vs. Kubemark

- *Kubemark*: A kubelet without running containers, but memory-intensive for large simulations.
  - use kubemark to simulate an N-nodes cluster without real kubelets.
  - with N standalone binaries
  - Each hollow-node consumes ~60M laptop
  - Spinning up a 1k-node cluster would consume 60G.

- *KWOK*: Only simulates node behavior
  - designs a central component to watch and ensure nodes' liveness with a central manager
  - slim tool that can reliably maintain 1k nodes and 100k pods locally


KubeCon | CloudNativeCon

North America 2023

Getting Started with KWOK

Create a Simulated Node

- Add a specific annotation for the node to be managed by KWOK controller
- Add taints to avoid scheduling actual running pods to fake nodes
- Users can customize the node resources

https://kwok.sigs.k8s.io/docs/user/kwok-manage-nodes-and-pods/

```yaml
apiVersion: v1
kind: Node
metadata:
  annotations:
    node.alpha.kubernetes.io/ttl: "0"
    kwok.x-k8s.io/node: fake
  labels:
    beta.kubernetes.io/arch: amd64
    beta.kubernetes.io/os: linux
    kubernetes.io/arch: amd64
    kubernetes.io/hostname: kwok-node-0
    kubernetes.io/os: linux
    kubernetes.io/role: agent
    node-role.kubernetes.io/agent: ""
    type: kwok
  name: kwok-node-0
spec:
  taints: # Avoid scheduling actual running pods to fake Node
    - effect: NoSchedule
      key: kwok.x-k8s.io/node
      value: fake
status:
  allocatable:
    cpu: 5000m
    nvidia.com/gpu: 8
    memory: 256Gi
    pods: 110
  capacity:
```

Create a Simulated Pod

- Simulated pods should land on simulated nodes
  - Add node affinity
  - Add pod toleration

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: fake-pod
  namespace: default
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: type
            operator: In
            values:
            - kwok
  tolerations:
    - key: "kwok.x-k8s.io/node"
      effect: "NoSchedule"
      value: fake
  containers:
    - name: fake-container
      image: fake-image
      resources:
        limits:
          cpu: 10m
          memory: 110Gi
          nvidia.com/gpu: "8"
```

https://kwok.sigs.k8s.io/docs/user/kwok-manage-nodes-and-pods/

Simulating the Lifecycle of Kubernetes Resources

- Stage API
  - KWOK configuration
  - Simulate different stages in the lifecycle of nodes and pods
  - Stack stage resource to simulate complete lifecycle

Simulating the Lifecycle of Kubernetes Resources

- KWOK Configuration
  - Stage API
  - Simulate different stages in the lifecycle of nodes and pods
  - Stack stage APIs to simulate complete lifecycle

kind: stage
spec:

Simulating the Lifecycle of Kubernetes Resources: Stage API Essentials

KubeCon CloudNativeCon North America 2023

Resource Kind and Version

resourceRef:

```yaml
metadata:
  name: pod-ready
spec:
  resourceRef:
    apiGroup: v1
    kind: Pod
```

Defining When to Execute

selector:

```yaml
selector:
  matchExpressions:
    - key: '.metadata.deletionTimestamp'
      operator: 'DoesNotExist'
    - key: '.status.phase'
      operator: 'In'
      values:
        - 'Pending'
    - key: '.status.conditions.[] | select( .type == "Initialized" ) | .status'
      operator: 'In'
      values:
        - 'True'
    - key: '.status.conditions.[] | select( .type == "ContainersReady" ) | .status'
      operator: 'NotIn'
      values:
        - 'True'
```

Simulating the Lifecycle of Kubernetes Resources: Stage API Essentials

Modify the Delay before the Stage is Applied

delay:

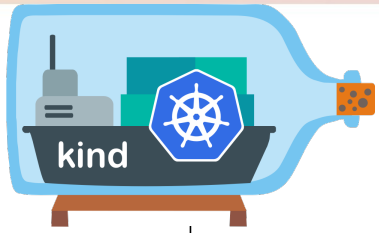
```yaml
delay:
  durationMilliseconds: 10000
  jitterDurationMilliseconds: 11000
```

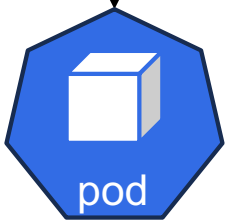
Define New State of the Resource

next:

```yaml
containerStatuses:
  {{ range $index, $item := .spec.containers }}
  {{ $origin := index $root.status.containerStatuses $index }}
  - image: {{ $item.image }}
    name: {{ $item.name }}
    ready: true
    restartCount: 0
    started: true
    state:
      running:
        startedAt: '{{ $now }}'
  {{ end }}
phase: Running
```

Real Job in Kind

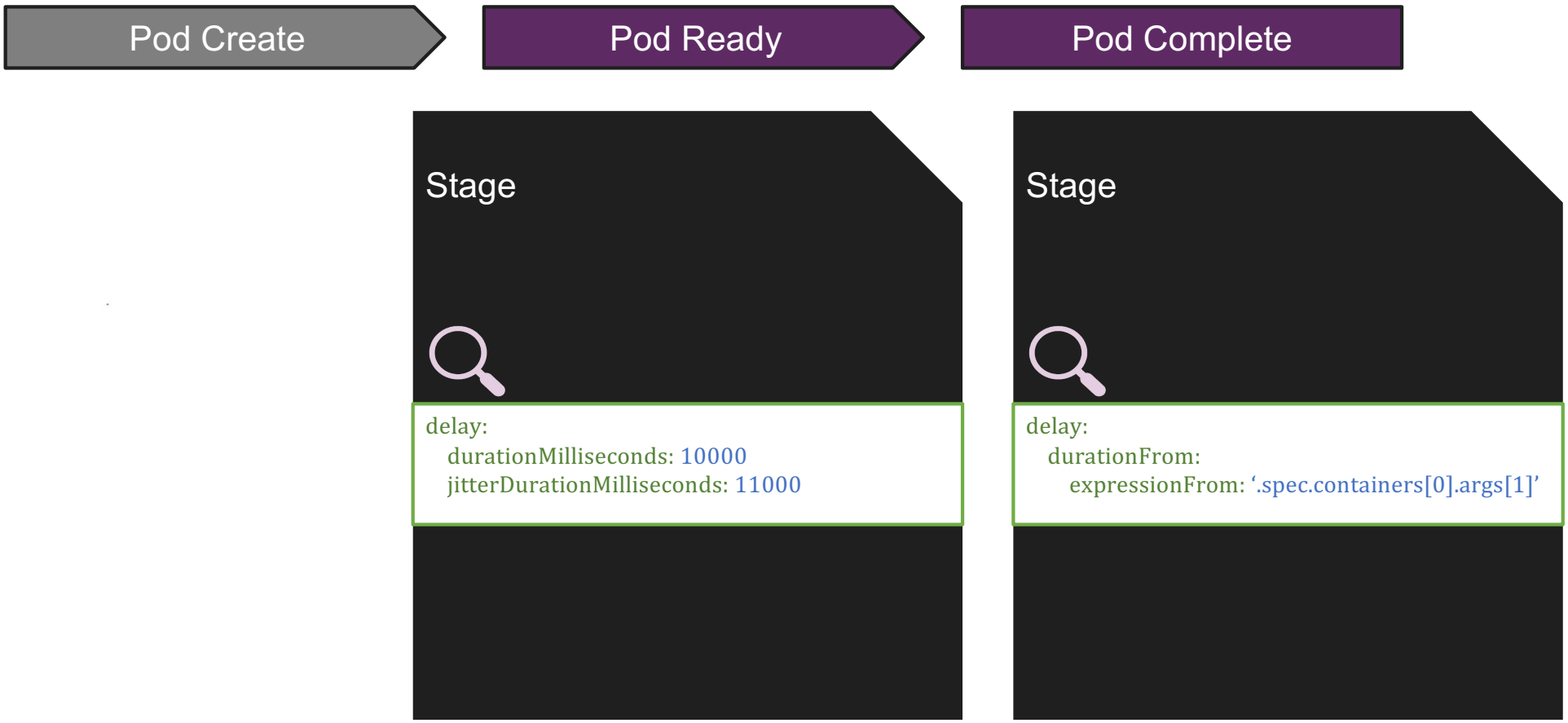





```
spec:
  containers:
  - name: baseline-cpu-job-short-0
    image: nginx:1.24.0
    command: ["sleep", "10"]
```

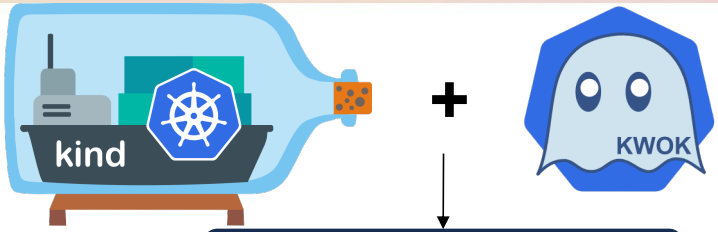
```
% kubectl get pods --output-watch-events --watch
EVENT      NAME                                READY   STATUS    RESTARTS   AGE
ADDED      baseline-cpu-job-short-0-75crs   0/1     Pending   0          0s
MODIFIED   baseline-cpu-job-short-0-75crs   0/1     Pending   0          0s
MODIFIED   baseline-cpu-job-short-0-75crs   0/1     ContainerCreating   0          0s
MODIFIED   baseline-cpu-job-short-0-75crs   1/1     Running             0          10s
MODIFIED   baseline-cpu-job-short-0-75crs   0/1     Completed           0          20s
MODIFIED   baseline-cpu-job-short-0-75crs   0/1     Completed           0          21s
MODIFIED   baseline-cpu-job-short-0-75crs   0/1     Completed           0          22s
```

Custom Pod LifeCycle



Simulated Job in KWOK

KubeCon CloudNativeCon North America 2023



+

Job (batch/v1)


```
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
            - key: type
              operator: In
              values:
                - kwok
  tolerations:
    - key: "kwok.x-k8s.io/node"
      operator: "Exists"
      effect: "NoSchedule"
  containers:
    - args:
      - sleep
      - {sleep_time}
```

```
% kubectl get pods --output-watch-events --watch
EVENT      NAME                                READY   STATUS    RESTARTS   AGE
ADDED      nomcadkwok-cpu-job-short-0-pkkqt    0/1     Pending   0          0s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    0/1     Pending   0          0s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    0/1     Pending   0          1s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    0/1     ContainerCreating   0          1s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    1/1     Running             0          11s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    0/1     Completed           0          22s
MODIFIED   nomcadkwok-cpu-job-short-0-pkkqt    0/1     Completed           0          23s
```

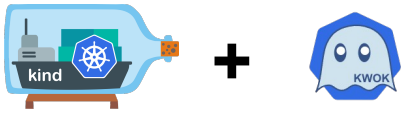

KubeCon | CloudNativeCon

North America 2023

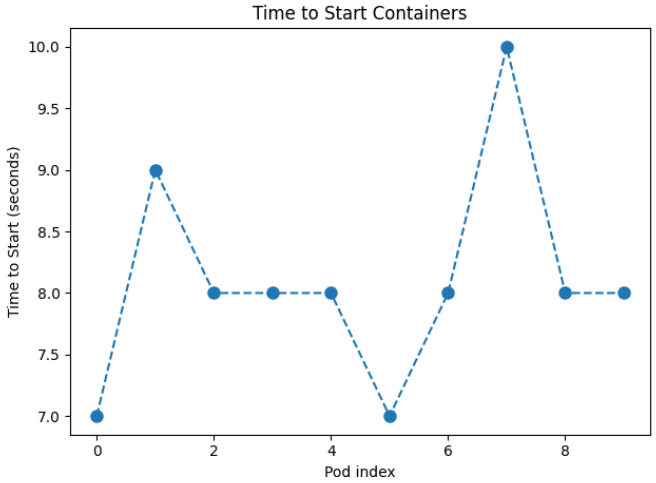
Things to Look Out For

Simulated vs. Real Behavior

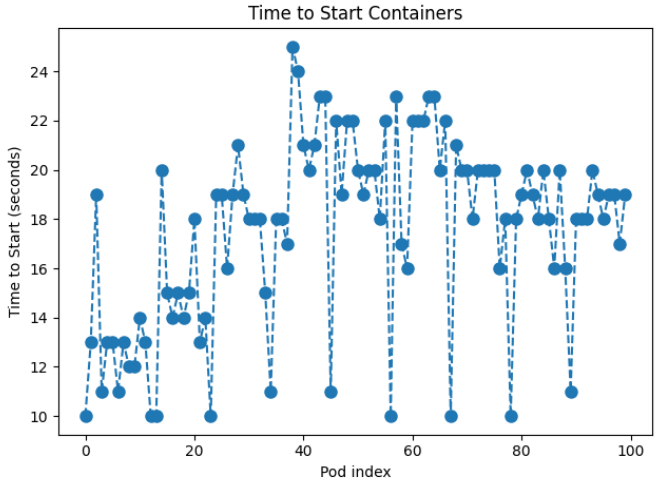
Real Nodes



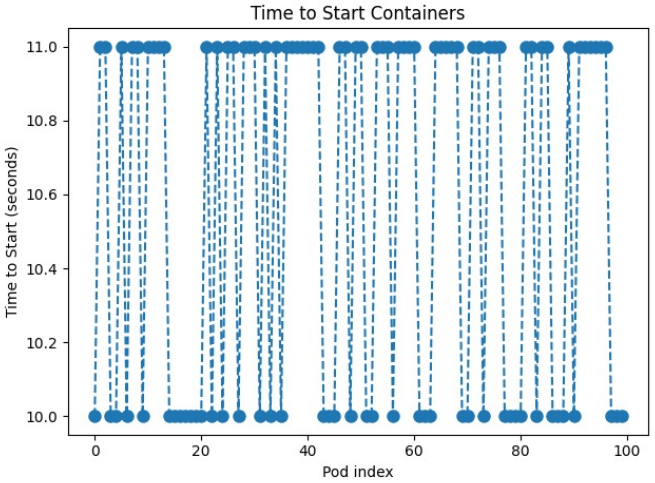
Fake Nodes



10 pods



100 pods



100 simulated pods



KubeCon | CloudNativeCon
North America 2023

Example Use Cases

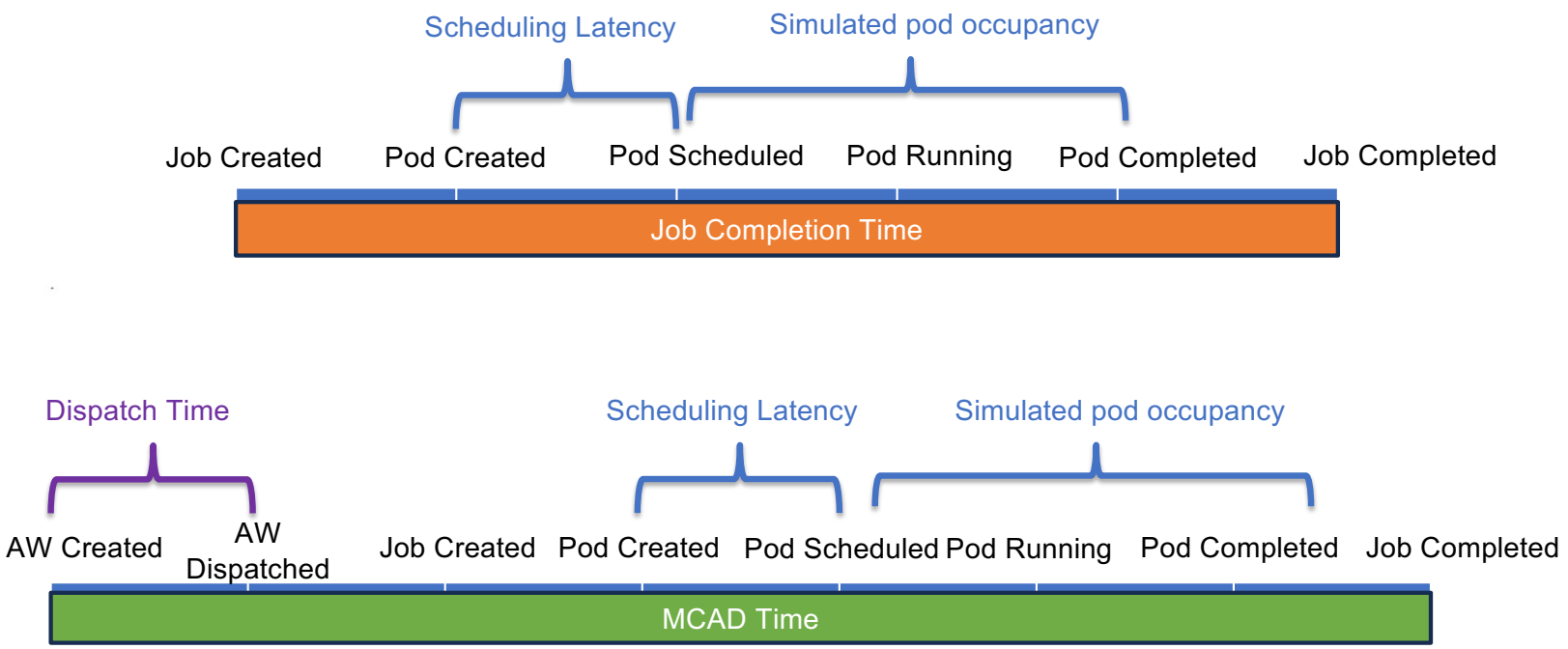
Why KWOK?

KubeCon CloudNativeCon North America 2023

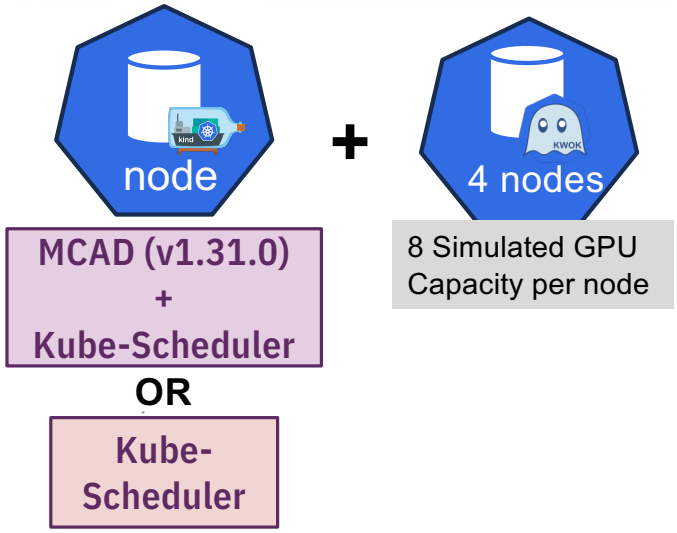
*"Best Practices: Improving **Batch Scheduling** Performance at **Scale** Using MCAD and KWOK "*

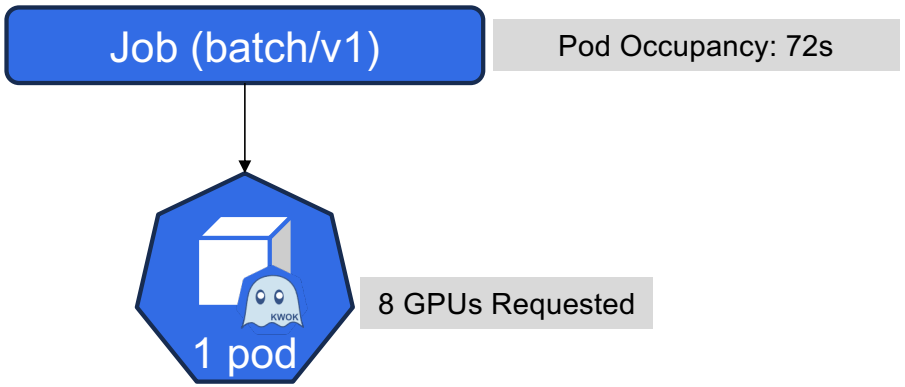
- **Challenges with AI Batch Scheduling**
  - Large AI model training demand advanced batch scheduling and preemption.
  - Long-running concurrent workloads are common.
- **Native Kubernetes Limitations**
  - Native Kubernetes scheduler lacks batch scheduling.
- **Viable Option: MCAD with Co-Scheduler**
  - MCAD offers advanced resource management options and follows an open-source model of continuous development and community contributions
  - Identifying and resolving bottlenecks for optimal performance is crucial.
- **The Role of KWOK**
  - Tests with various MCAD versions or commits help identify performance bottlenecks

Timestamped Events of Interest



Baseline Use Case Validation





A system with inter-job request interval **greater than 18s should not induce queueing**

Mode	Mean arrival (s)	Pod occupancy (s)	Job Completion Time (s)	Scheduling Latency (s)	Dispatch Time (s)
No MCAD	20	72	72	0	N/A
MCAD	20	72	71.9	0	0.38

Baseline Use Case (No-MCAD) - High load

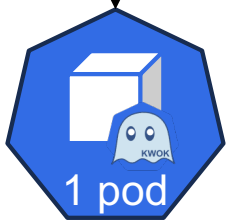
+

Kube-Scheduler

8 Simulated GPU Capacity per node

---

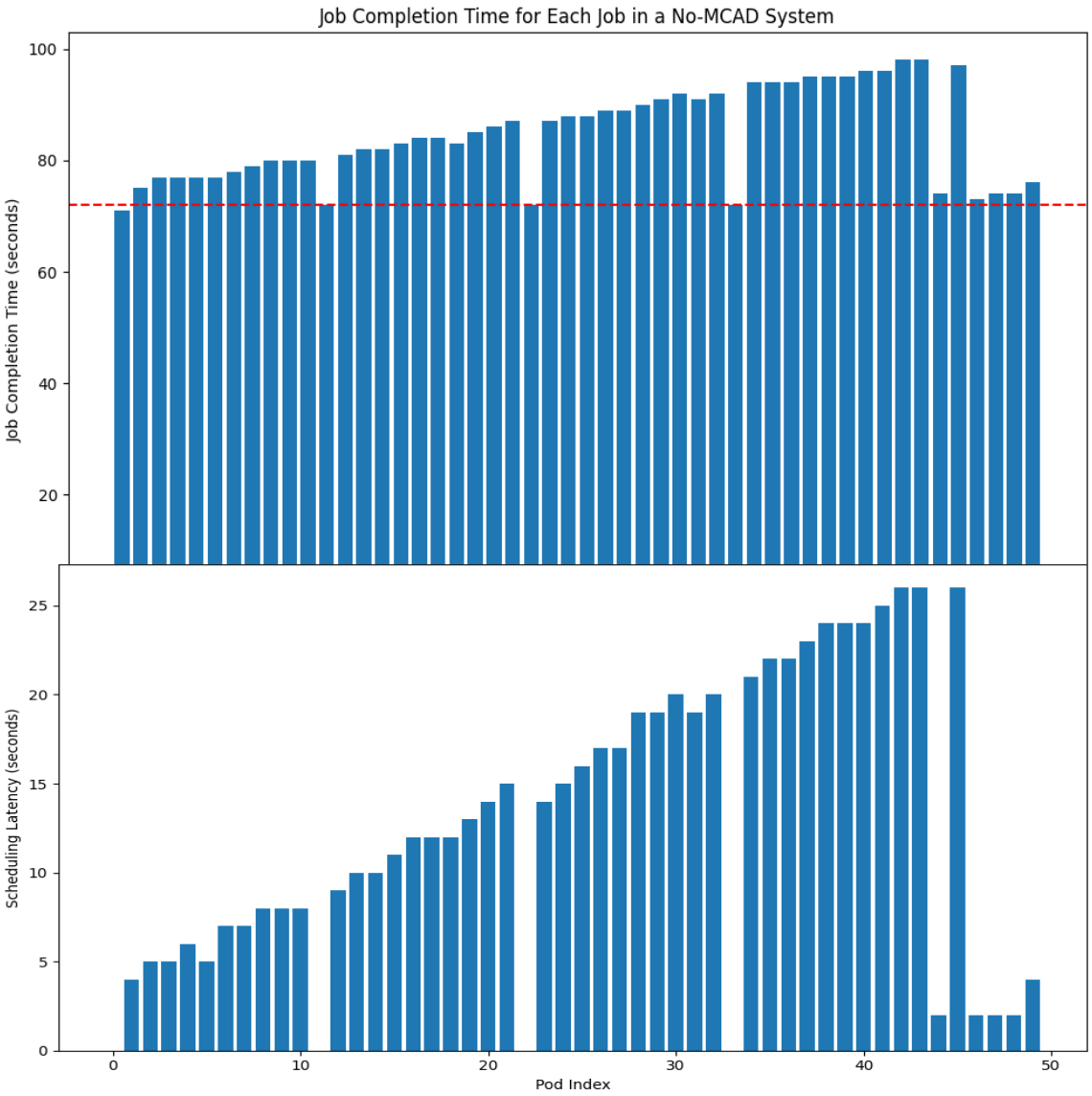
50 Job (batch/v1)



Pod Occupancy: 72s

8 GPUs Requested

Inter-Job Request: **17s**



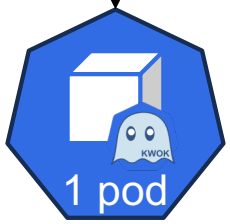
Baseline Use Case (MCAD) - High load

+

MCAD (v1.31.0)
+
Kube-Scheduler

8 Simulated GPU Capacity per node

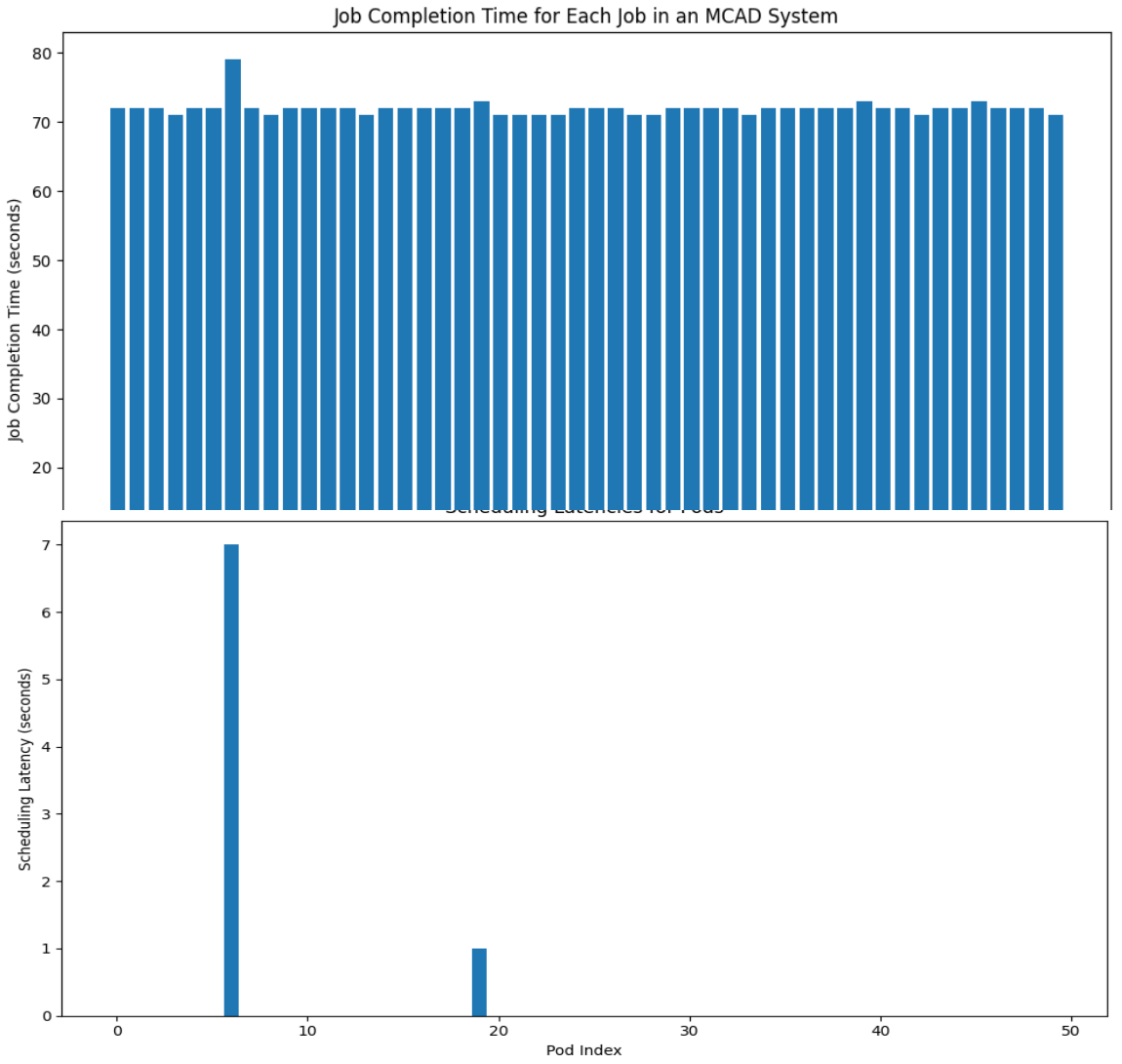
50 Job (batch/v1)



Pod Occupancy: 72s

8 GPUs Requested

Inter-Job Request: **17s**



Simulated Training Job - Coscheduler plugin

Coscheduler Plugin (v0.26.7) : Kubernetes sig-scheduling team Scheduling Framework plugin for batch job scheduling

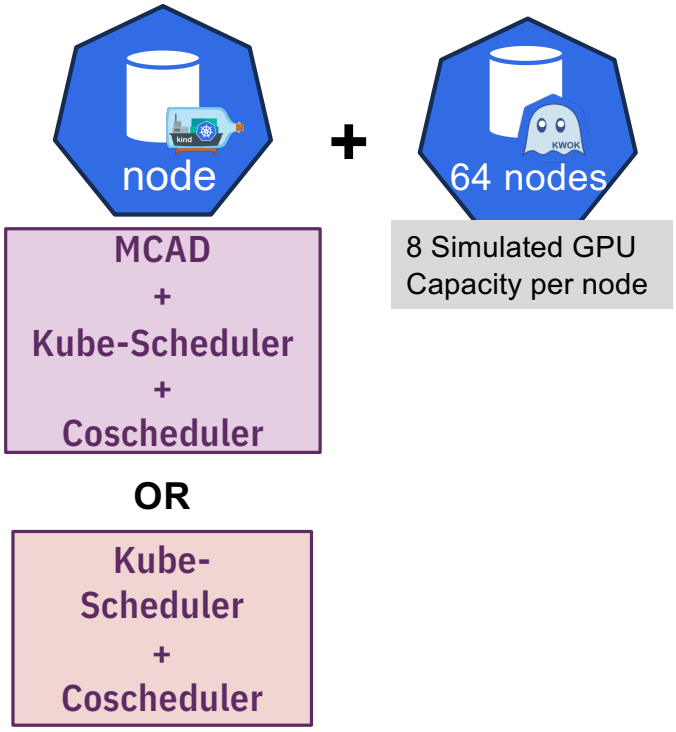
PodGroup: A group of pods that should be scheduled and run together

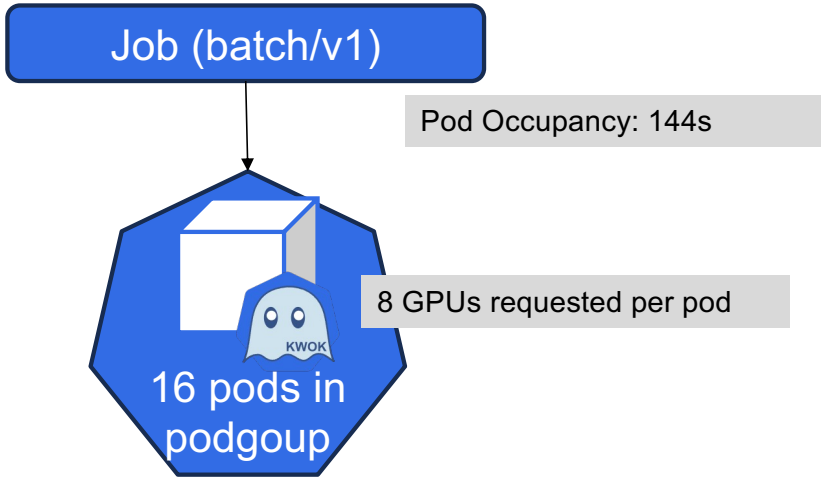
```
kind: PodGroup
        metadata:
          name: pg-example
          namespace: default
        spec:
          scheduleTimeoutSeconds: 10
          minMember: {num_pod}
```

```
labels:
        scheduling.x-k8s.io/pod-group: pg-example
```

https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/kep/42-podgroup-coscheduling

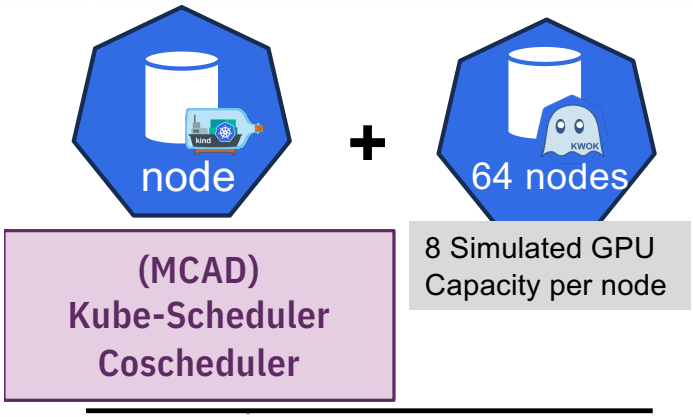
Coscheduler Use Case

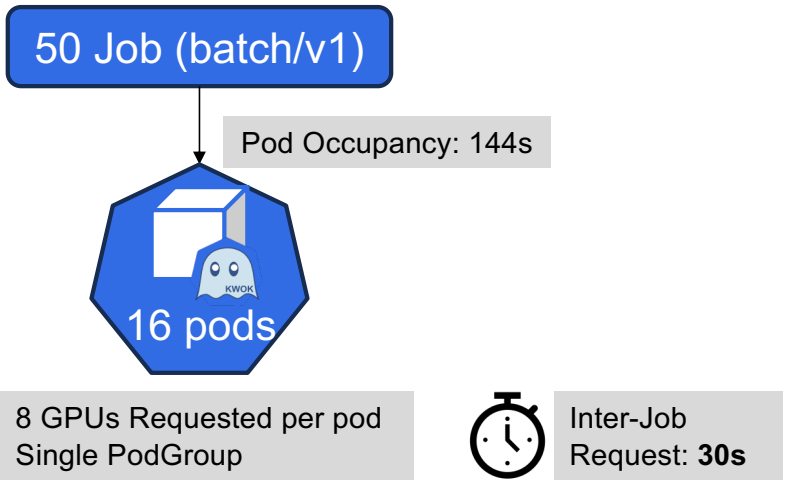




A system with inter-job request interval **greater than 36s should not induce queueing**

Coscheduler Use Case - High load



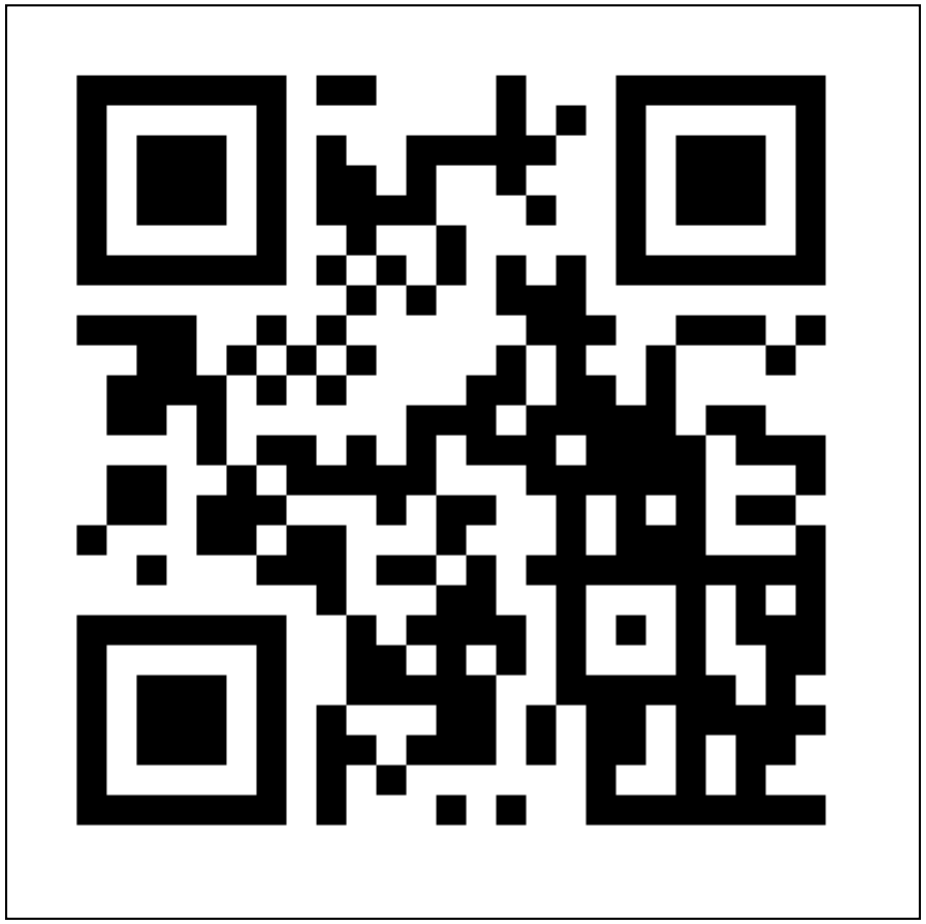


Resource Management	Avg. Job Completion Time (s)	Avg. Scheduling Latency (s)	Avg. Dispatch Time (s)
Kube-Scheduler + Coscheduler	332.26	188.40	N/A
MCAD v1.31.0 + Kube-Scheduler + Coscheduler	186.98	42.47	355.21
MCAD v1.34.1 + Kube-Scheduler + Coscheduler	**144.02**	**0**	**181.66**

Conclusions

- MCAD for managing batch jobs in a single or multi-cluster environment
- Testing resource management at scale is critical
- Tools like KWOK provide a lightweight simulator

https://github.com/project-codeflare/multi-cluster-app-dispatcher



Please scan the QR Code above
to leave feedback on this session