



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Instrumenting Kueue for ML Training

Amy Chen, CoreWeave
Gabriel Saba, Google



Our Talk

- Motivation
- What is Scheduling?
- Kueue Overview
- “Why isn’t my workload scheduling” Checklist
- Instrumenting Kueue



Accelerators are Expensive



“Why isn’t my workload scheduling?”



KubeCon

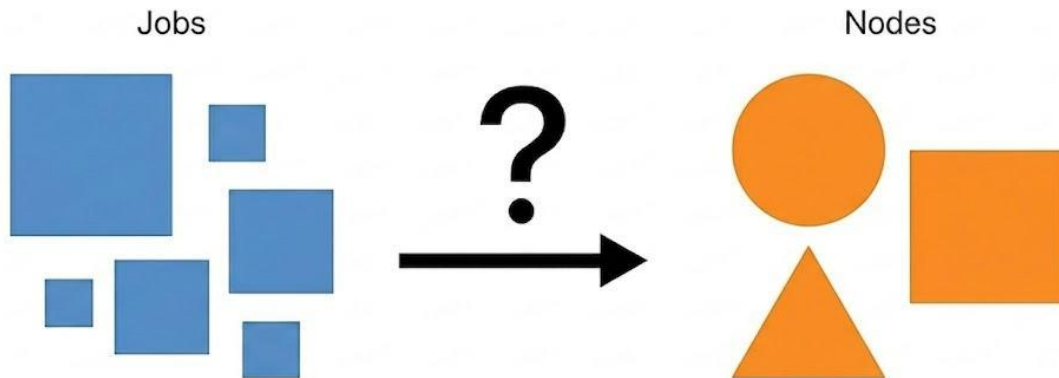


CloudNativeCon

Europe 2026

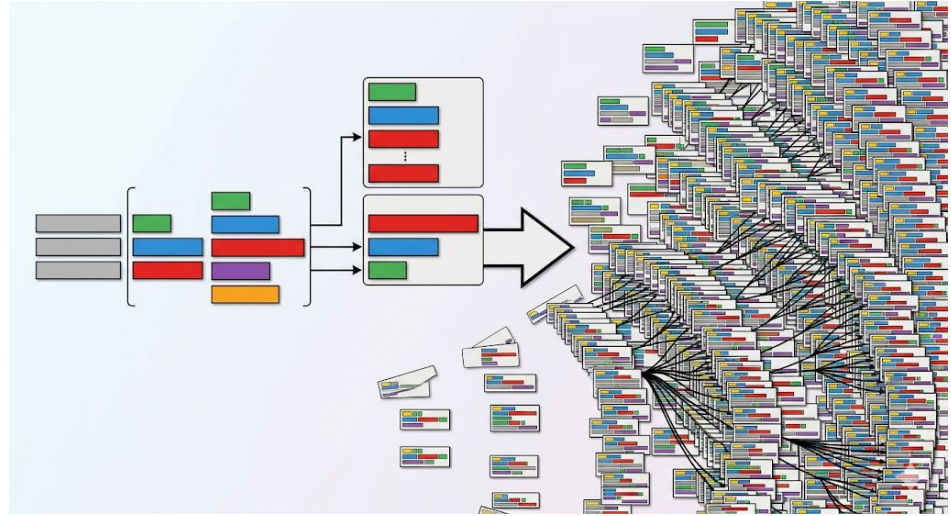
What is Scheduling?

- Temporarily assign resources to jobs



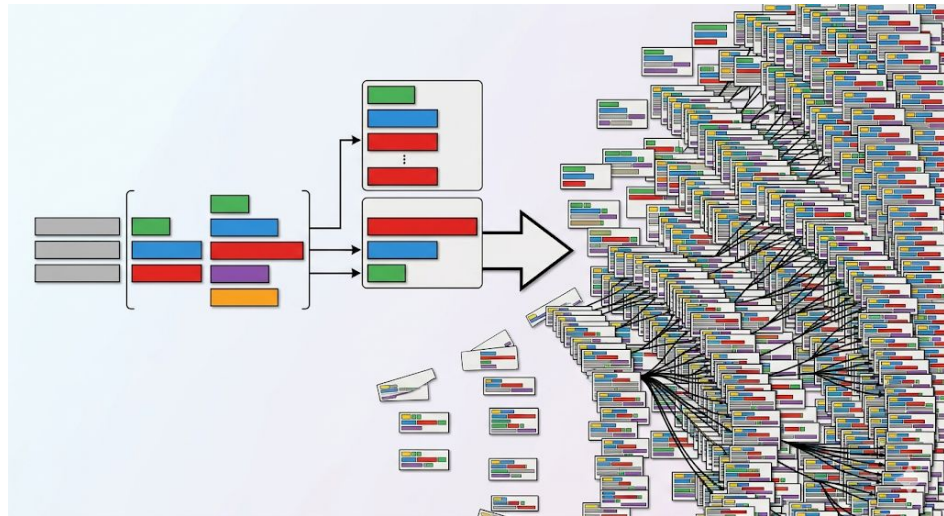
Scheduling is Hard

- Even simple (decision) problems are NP-Complete
 - 3 identical processors
 - n compute tasks, which have different run times
 - minimize completion time
- Tractable approximations
 - greedy online - 1.67x
 - greedy offline - 1.22x
- Queue's problem is much more complex



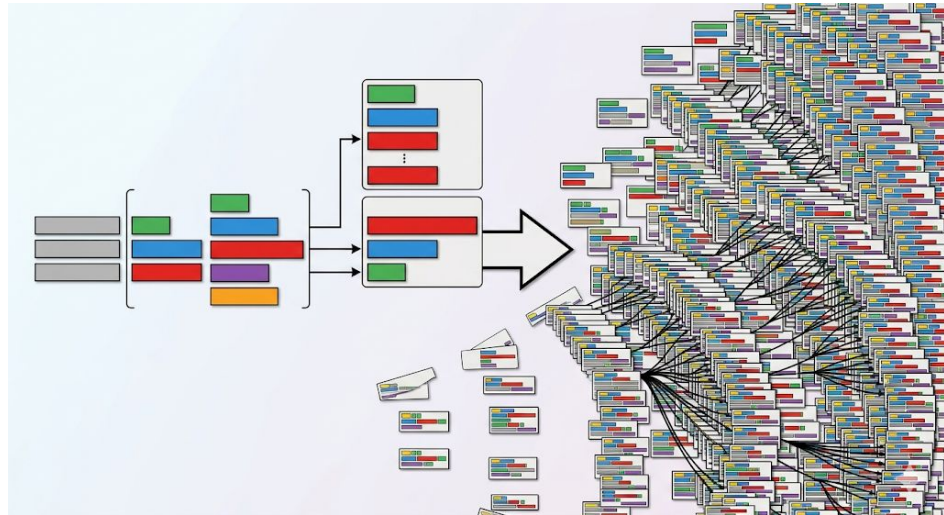
Scheduling is Hard

- Even simple (decision) problems are NP-Complete
 - 3 identical processors
 - n compute tasks, which have different run times
 - minimize completion time
- Tractable approximations
 - greedy online - 1.67x
 - greedy offline - 1.22x
- Kueue's problem is much more complex



Scheduling is Hard

- Even simple (decision) problems are NP-Complete
 - 3 identical processors
 - n compute tasks, which have different run times
 - minimize completion time
- Tractable approximations
 - greedy online - 1.67x
 - greedy offline - 1.22x
- Kueue's problem is much more complex



Kueue's Problem

- Heterogeneous machines
- Online problem
- Many-faceted and configurable objective
 - quota enforcement
 - prefer high-priority workloads
 - most of capacity being used
 - ★ avoid unnecessary preemptions

Kueue's Problem

- Heterogeneous machines
- Online problem
- Many-faceted and configurable objective
 - quota enforcement
 - prefer high-priority workloads
 - most of capacity being used
 - ★ avoid unnecessary preemptions

Kueue's Problem

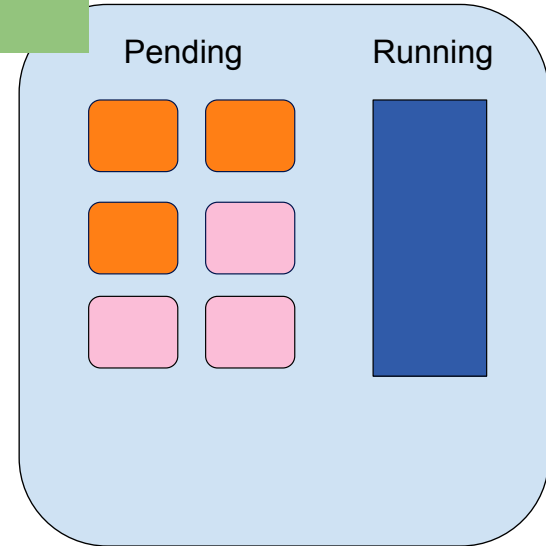
- Heterogeneous machines
- Online problem
- Many-faceted and configurable objective
 - quota enforcement
 - prefer high-priority workloads
 - most of capacity being used
 - ★ avoid unnecessary preemptions

What is Kueue

- Quota Management
 - admitted $\leq$ capacity
- Queueing
 - which job is admitted next?

orange job with 3 pods
pink job with 3 pods

Cluster with
capacity = 3



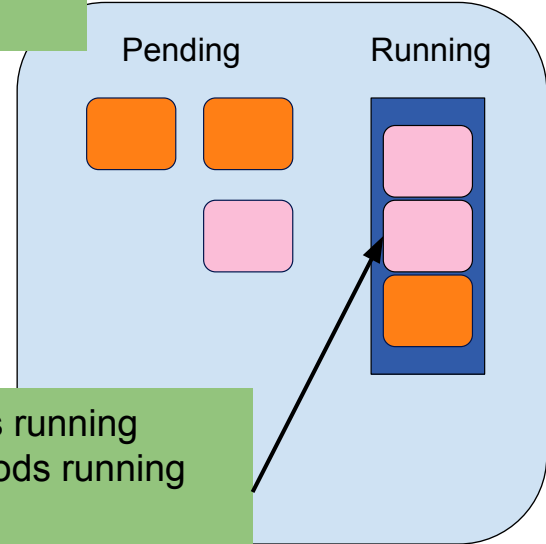
What is Kueue

- Quota Management
 - admitted $\leq$ capacity
- Queueing
 - which job is admitted next?

orange job with 3 pods
pink job with 3 pods

Cluster with
capacity = 3

without Kueue



$\frac{2}{3}$ pink pods running
 $\frac{1}{3}$ orange pods running
neither makes progress



KubeCon

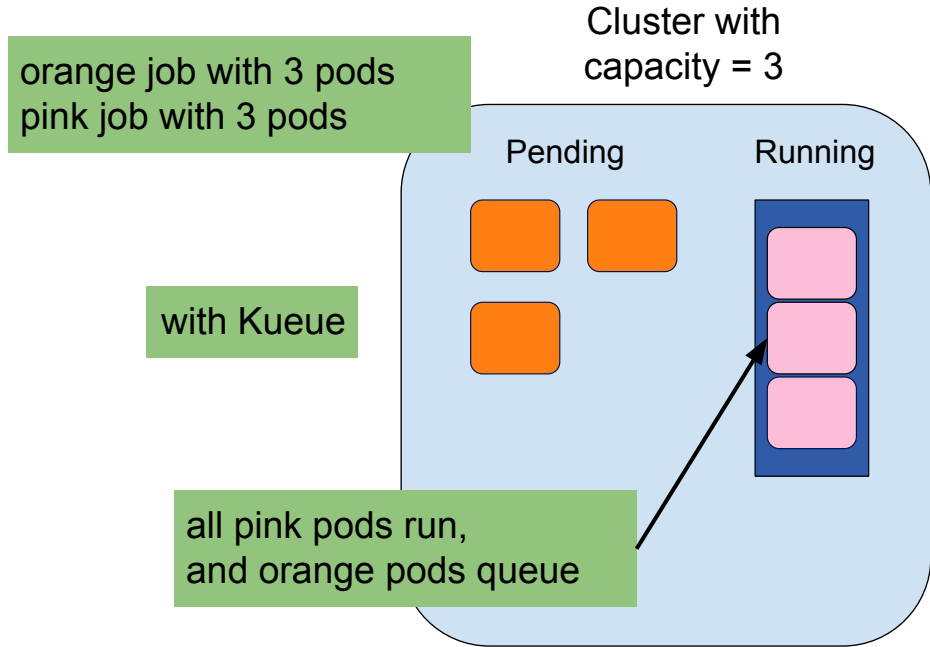


CloudNativeCon

Europe 2026

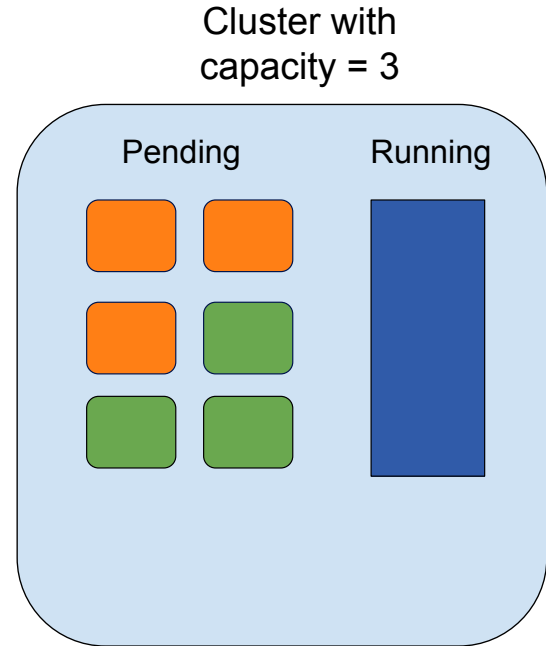
What is Kueue

- Quota Management
 - admitted $\leq$ capacity
- Queueing
 - which job is admitted next?



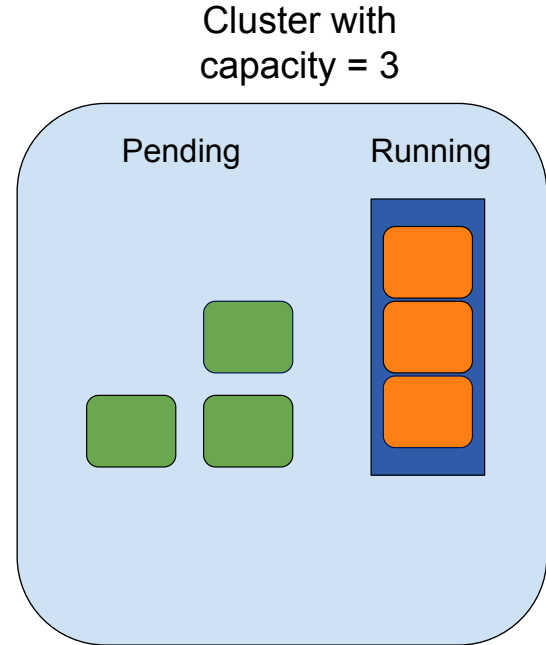
What is Kueue

- Quota Management
 - $\text{admitted} \leq \text{capacity}$
- Queueing
 - which job is admitted next?



What is Kueue

- Quota Management
 - $\text{admitted} \leq \text{capacity}$
- Queueing
 - which job is admitted next?



Kueue Basics

- **ResourceFlavors** abstract hardware differences
 - e.g. **gpu types**
 - via **node labels**/taints
- **ClusterQueues** define quotas and queue workloads
- **Cohorts** define quotas and represent organizations

```
apiVersion: kueue.x-k8s.io/v1beta2
kind: ResourceFlavor
metadata:
  name: "h100-flavor"
spec:
  nodeLabels:
    nvidia.com/gpu.product: H100-SXM5-80GB
---
apiVersion: kueue.x-k8s.io/v1beta2
kind: ResourceFlavor
metadata:
  name: "a100-flavor"
spec:
  nodeLabels:
    nvidia.com/gpu.product: A100-SXM4-80GB
```

Kueue Basics

- **ResourceFlavors** abstract hardware differences
 - e.g. gpu types
 - via node labels/taints
- **ClusterQueues** define quotas and queue workloads
- **Cohorts** define quotas and represent organizations

```
apiVersion: kueue.x-k8s.io/v1beta2
kind: ClusterQueue
metadata:
  name: project-queue
spec:
  cohortName: research-cohort
  queueingStrategy: BestEffortFIFO
  resourceGroups:
    - coveredResources: ["nvidia.com/gpu"]
      flavors:
        - name: h100-flavor
          resources:
            - name: "nvidia.com/gpu"
              nominalQuota: 8
```

Kueue Basics

- **ResourceFlavors** abstract hardware differences
 - e.g. gpu types
 - via node labels/taints
- **ClusterQueues** define quotas and queue workloads
- **Cohorts** define quotas and represent organizations

```
apiVersion: kueue.x-k8s.io/v1beta2
kind: Cohort
metadata:
  name: research-cohort
spec:
  parentName: root-cohort
  resourceGroups:
    - coveredResources: ["nvidia.com/gpu"]
      flavors:
        - name: h100-flavor
          resources:
            - name: "nvidia.com/gpu"
              nominalQuota: 32
              borrowingLimit: 0
              lendingLimit: 16
        - name: a100-flavor
          resources:
            - name: "nvidia.com/gpu"
              nominalQuota: 128
```

Scheduling Cycle - Overview

- Consider head workload from each ClusterQueue
 - priority queue
- Simulate resource assignment (and preemptions)
 - score flavors based on borrowing/preemption
 - configurable via FlavorFungibility
- Schedule workloads
 - fair sharing, borrowing, priority, preemption, etc for ordering
 - skip overlapping preemption

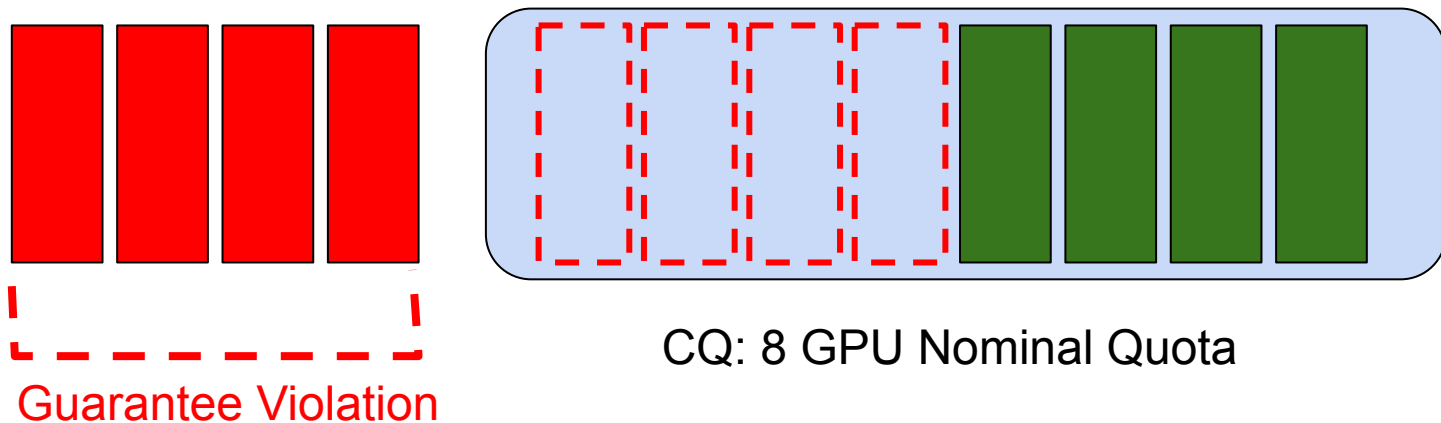
Scheduling Cycle - Overview

- Consider head workload from each ClusterQueue
 - priority queue
- Simulate resource assignment (and preemptions)
 - score flavors based on borrowing/preemption
 - configurable via FlavorFungibility
- Schedule workloads
 - fair sharing, borrowing, priority, preemption, etc for ordering
 - skip overlapping preemption

Scheduling Cycle - Overview

- Consider head workload from each ClusterQueue
 - priority queue
- Simulate resource assignment (and preemptions)
 - score flavors based on borrowing/preemption
 - configurable via FlavorFungibility
- **Schedule workloads**
 - fair sharing, borrowing, priority, preemption, etc for ordering
 - skip overlapping preemption

Guarantee Violation



- CQ has 8 GPU nominal
- User submits 8 one GPU workloads
- 4 are admitted, 4 are queued for a long time

“Why can’t I schedule my workload even though I have guarantees available?”

Scheduling Failure Tiers



KubeCon



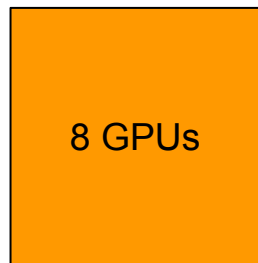
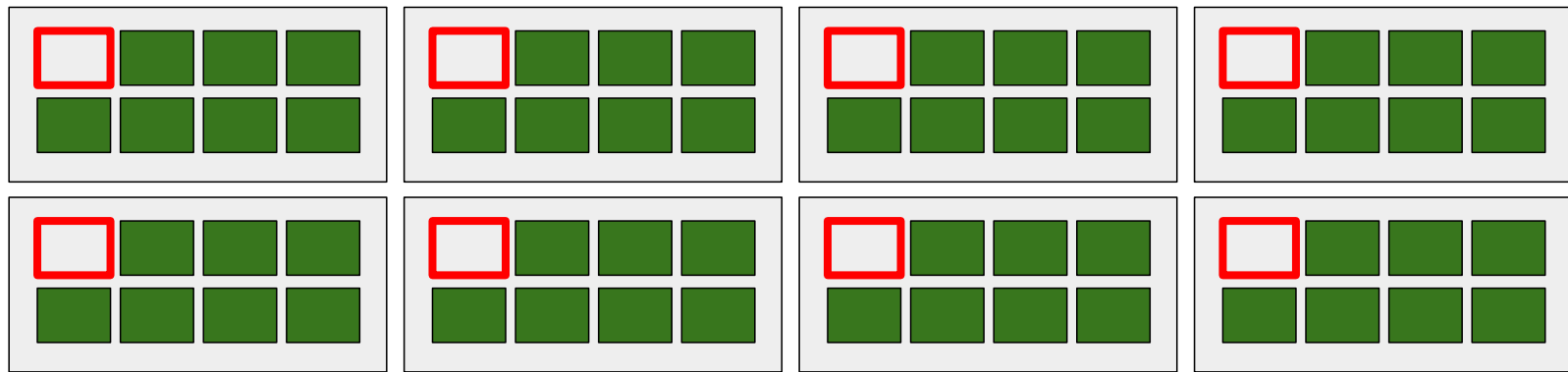
CloudNativeCon

Europe 2026

Why isn't my workload scheduling?

1. Guarantee Violation exists: $\text{Queued} \leq \text{Unallocated Nominal Quota}$
2. Rule out fragmentation
3. Slow CQ Admission rate
4. Slow overall Cohort Preemption rate
5. Kueue has released workload to Kubernetes
 - a. `ScheduleGate` -> `Pending`
 - b. `Suspend:true` -> `Suspend:false`
6. Zap log level 5

Fragmentation



8 nodes with 1 GPU "holes"
Submit an 8 GPU workload

Serialized Preemption Issue

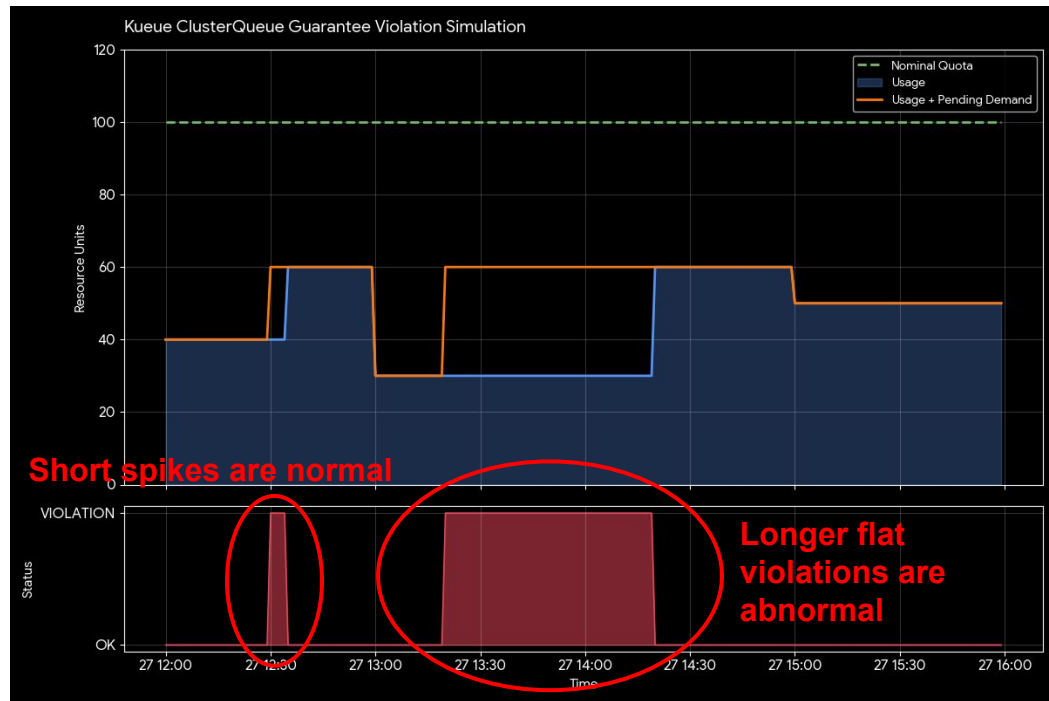
Occurs during...

- Cluster migration
- Sudden spike of high priority inference workloads for a ClusterQueue

NOTE: THE FOLLOWING DATA IS SIMULATED.

Serialized Preemption Issue

Step 1: Guarantee Violation Alert



```
kueue_cluster_queue_resource_usage{cluster_queue="demo-cq" resource="gpu"}
```

```
kueue_cluster_queue_nominal_quota{cluster_queue="demo-cq" resource="gpu"}
```

```
kueue_cluster_queue_pending_workloads_resource_request{cluster_queue="demo-cq", resource="gpu"}
```

- Does not exist in Kueue (yet #9741)
- Temp Workaround:
Create `CustomResourceStateMetrics` gauge for Kube State Metrics

Serialized Preemption Issue

Step 2: Rule Out Fragmentation

Example:

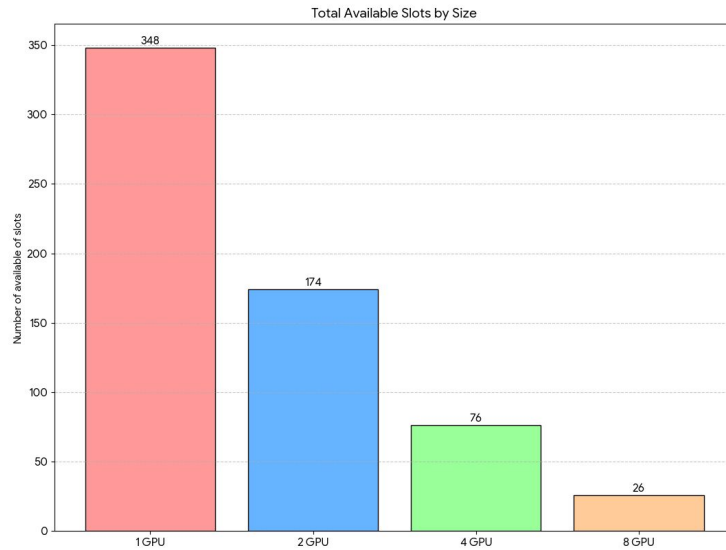
User is submitting a 12 8GPU workloads. You need at least 12 nodes to be completely free.

If you have 8GPU slots free, you can rule out fragmentation.

If there are no 8GPU slots free, the cluster is fragmented. Fragmentation is loosely an “acceptable” reason that prevents workloads from being scheduled.

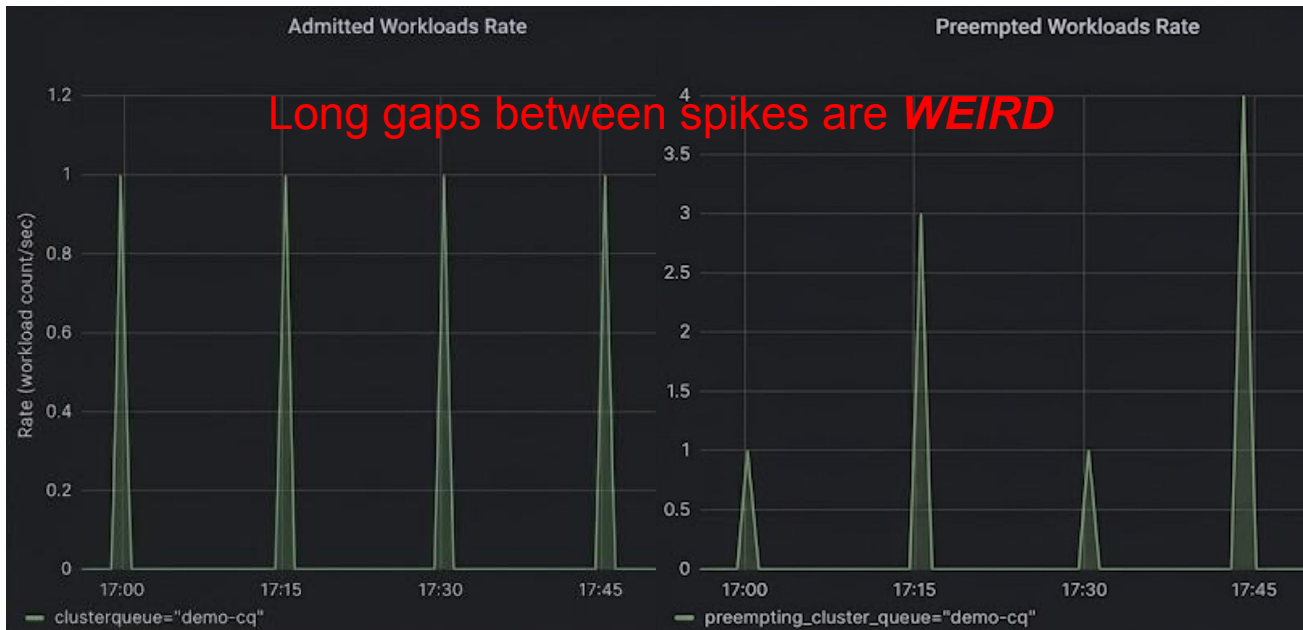
- (Topology Decay is out of scope for this talk. The community still needs a solution.)

Fragmentation is not an issue for this example.



Serialized Preemption Issue

Step 3: Check Admission/Preemption Rates for CQ



```
sum(rate(kueue_admitted_workloads_total[1m])) by (cluster_queue)
```

```
sum(rate(kueue_preempted_workloads_total[1m])) by (preempting_cluster_queue)
```

Checking admission/preemption right after submitting a lot of workloads.

Serialized Preemption Issue

Step 4: Check if Kueue released workload to k8s

```
amy@CW-LP2F6TP6N2-L~ % kubectl get pods -l  
kueue.x-k8s.io/queue-name=demo-cq
```

NAME	READY	STATUS	RESTARTS	AGE
kubecon-pod-a1b2c	0/1	SchedulingGated	0	30s
kubecon-pod-d3e4f	0/1	SchedulingGated	0	30s
kubecon-pod-a3d38	0/1	SchedulingGated	0	30s
kubecon-pod-a4a97	0/1	SchedulingGated	0	30s
kubecon-pod-a61b6	0/1	SchedulingGated	0	30s
kubecon-pod-x5y6z	0/1	SchedulingGated	0	30s
kubecon-pod-m0n9p	0/1	SchedulingGated	0	30s
kubecon-pod-q1r2s	0/1	SchedulingGated	0	30s
kubecon-pod-t3u4v	0/1	SchedulingGated	0	30s
kubecon-pod-m9p9p	0/1	SchedulingGated	0	30s
kubecon-pod-b7c8d	0/1	SchedulingGated	0	30s
kubecon-pod-e9f0g	0/1	SchedulingGated	0	30s

No Kueue did not release
workloads to k8s. See
SchedulingGated

Serialized Preemption Issue

Step 5: Search for “weird” logs

Step 6: Check for overall system health/change

For this bug there were no weird logs.

I usually just search for the workload name in logs.

Checklist: Serialized Preemption

- ✓ Saw guarantee violations
- ✗ Fragmentation is not the issue
- ✓ Saw “trickling” admission/preemption
- ✗ Kueue did not release workload to kubernetes
- ✗ No “weird” logs
- ✗ Check overall system health to see if anything stands out

🤔 So whats going on?

Dig into Kueue scheduling.



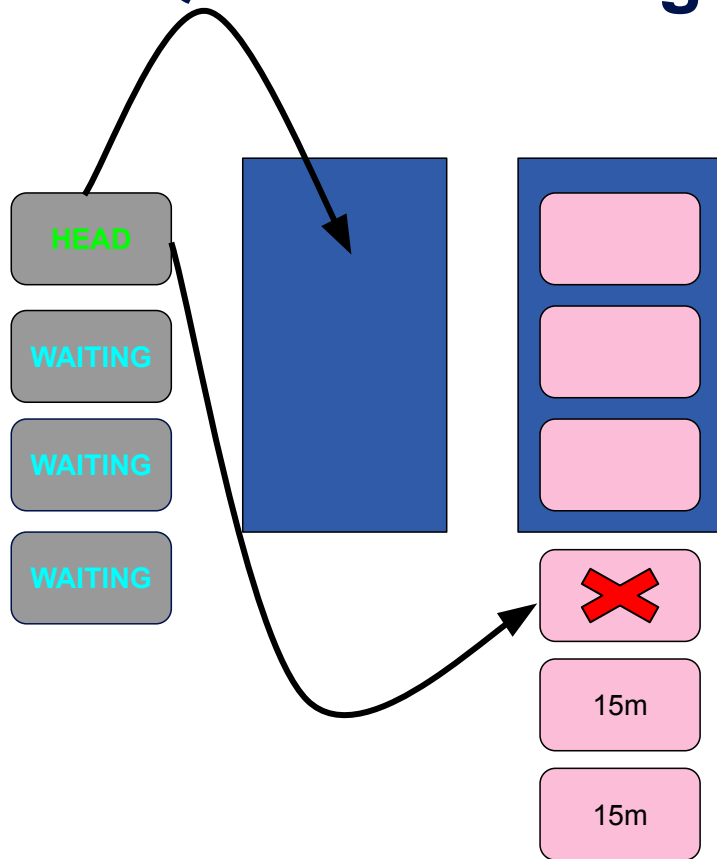
KubeCon



CloudNativeCon

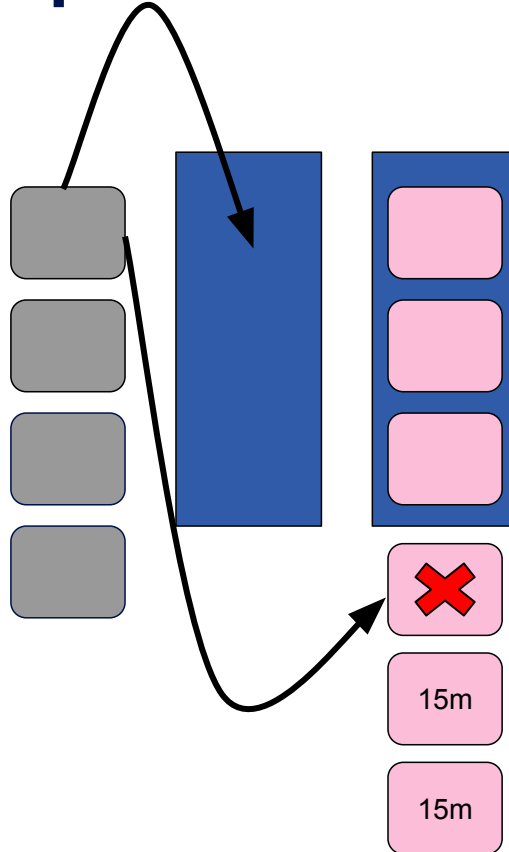
Europe 2026

Head of Cluster Queue Blocking



Serialized Preemption: Root Cause

1. Workload is reclaiming guarantees and requires preemption target to complete before admitting workload.



2. Each preemption candidate at worst needs 15min to complete preemption due to `terminationGracePeriodSeconds`

Checklist: Zombie Pod

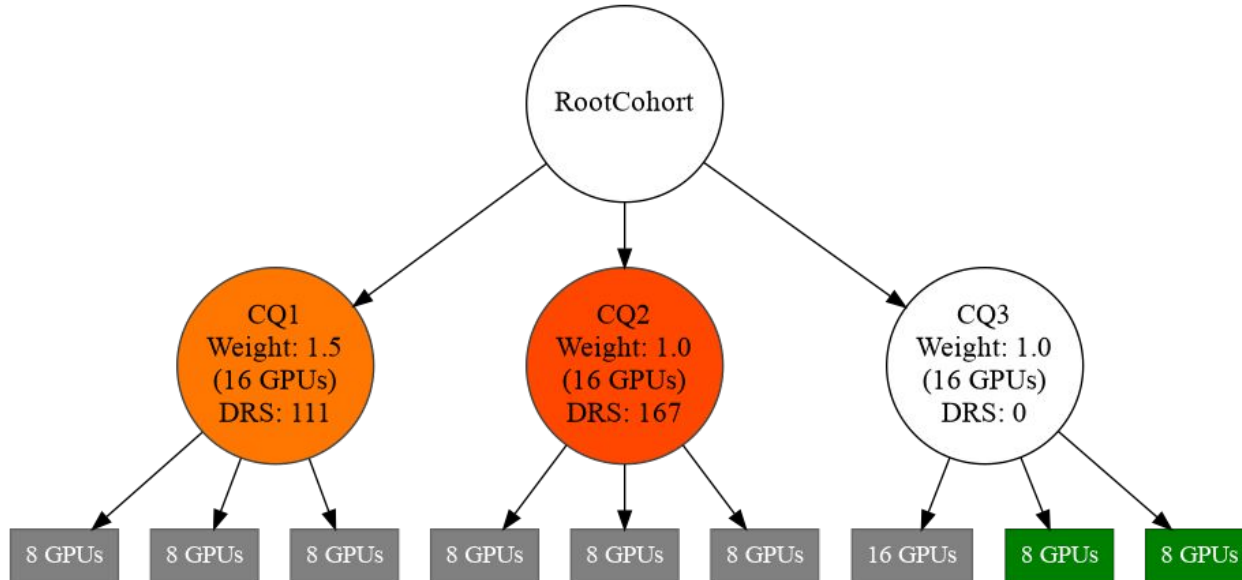
- ✓ Saw guarantee violations
- ✗ Fragmentation is not the issue
- ✓ Saw overall slowdown in admission rates across ClusterQueues
- ✗ No buildup of `Pending / Suspend: false` workloads
- 🟡 Potentially “weird” logs
 - "Workload has overlapping preemption targets with another workload"
- ✓ Saw something weird in system health: Pod living past its `terminationGracePeriodSeconds`

🤔 So whats going on?

Dig into Kueue scheduling.

Preemption Target Selection

- Reverse criteria as during scheduling
 - Lowest priority
 - Highest Dominant Resource Share (DRS)
- Many workloads will end up targeting the same workloads



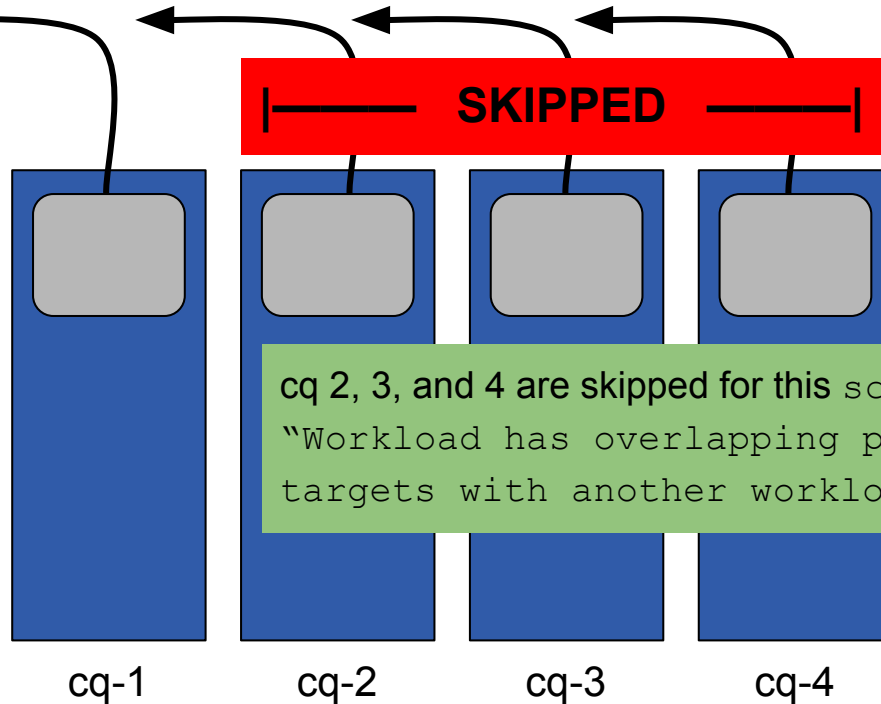
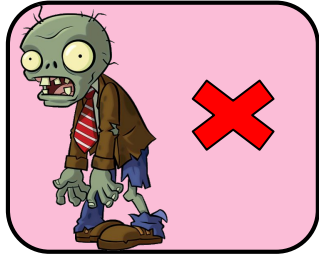
KubeCon



CloudNativeCon

Europe 2026

Conclusion: Zombie Pod



Checklist: FairSharing Preemption Bug

- ✓ Saw guarantee violations
- ✗ Fragmentation is not the issue
- ✗ Did not see an overall slowdown in admission/preemption rates across ClusterQueues
- ✗ No buildup of `Pending / Suspend: false` workloads
- ✗ No “weird” logs
- ✓ Errors coincided with system changes: enabled Preemption FairSharing
 - Turn on zap log level 5
 - “DominantResourceShare values used during tournament”

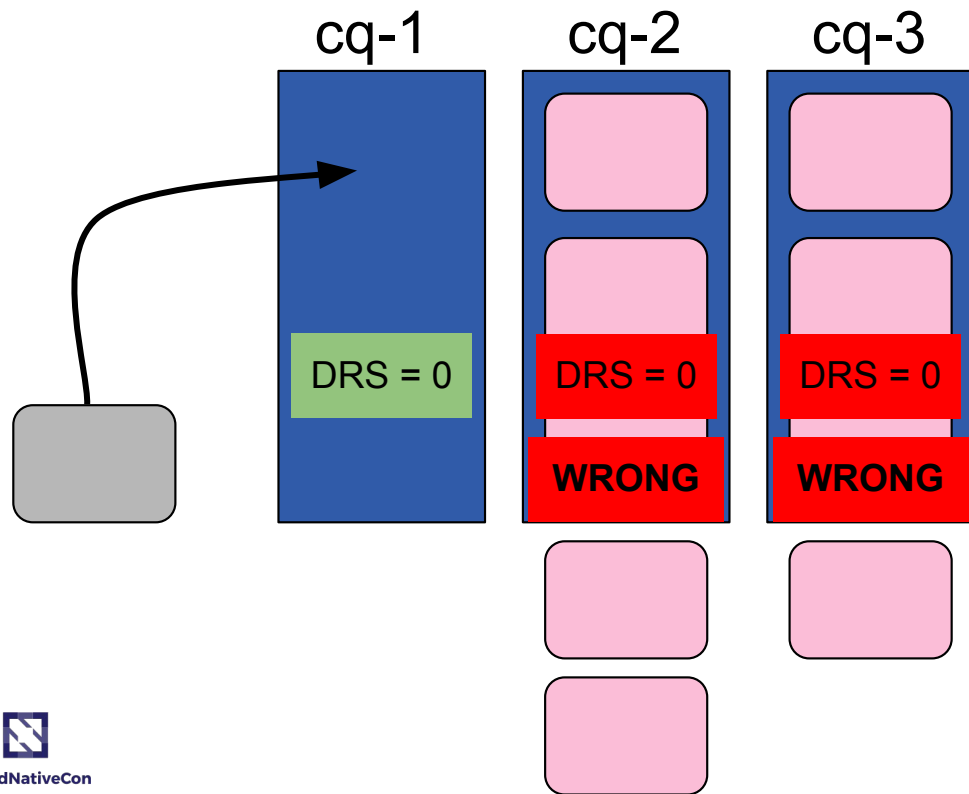
🤔 So whats going on?

Dig into Kueue scheduling.

Workload Ordering

- Simulate DRS after admission of workloads
- Tournament: schedule workloads with low DRS first
- 0 DRS implies within guarantees for all resources

Conclusion: Fairshare Preemption Bug



DRS = 0 means
Reclaiming guarantees

DRS > 0 means
CQ is bursting past guarantees

Dashboarding Recommendations

- Guarantee violations
 - Issue #9741
 - Total pending resources per CQ
 - Issue #7539
 - Cohort quotas and usage
 - Hierarchy directly in prometheus
- Admission/Preemption rates
- Fragmentation

Join us!

Docs: <https://kueue.sigs.k8s.io/>

Slack: #wg-batch

(meet every other Thursday at 7am PST)