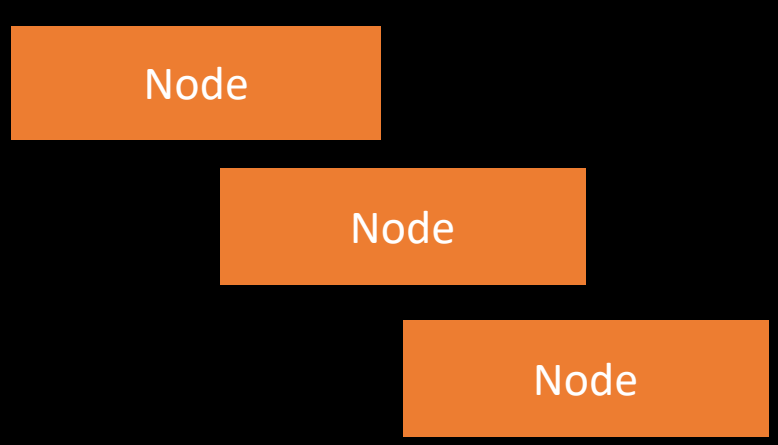
Misadventures In Large Scale Cluster Performance

Shane Corbett, AWS
Dima Ilchenko, Fortinet

Scaling

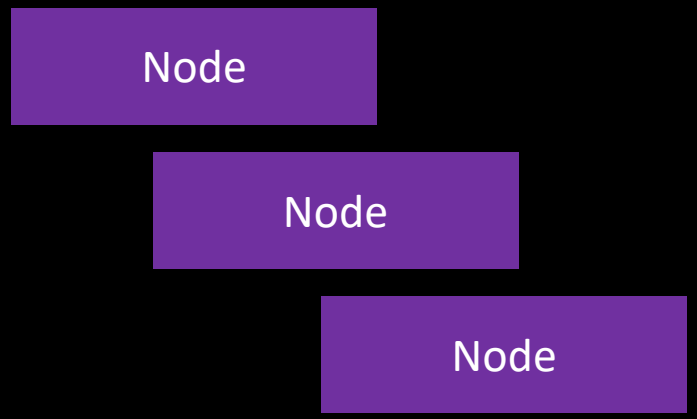
5,000 nodes
150,000 total pods
300,000 total containers
110 pods per node

Why Not Nodes?



500 Nodes

Crash



6,000 Nodes

FINE

Why Not Nodes?

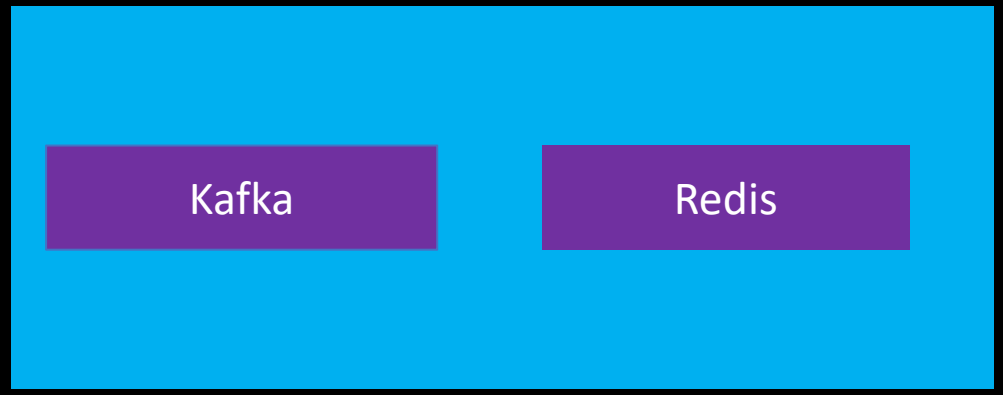
16 vCPU Node

VS.

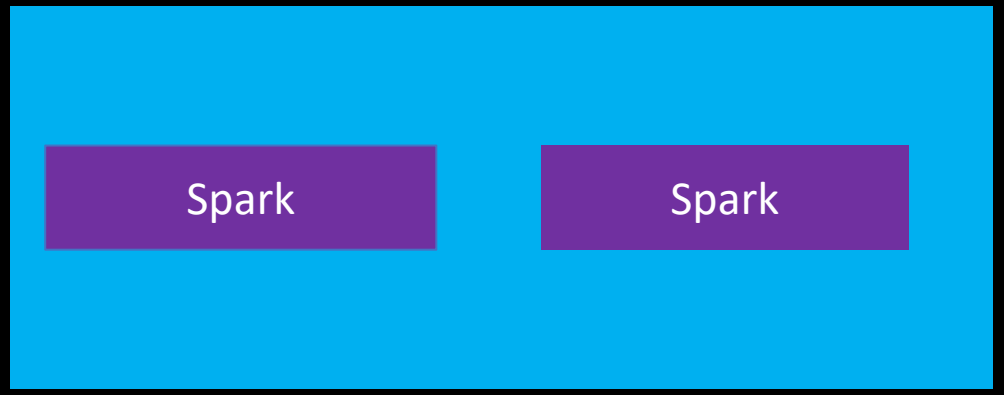
96 vCPU Node

Why Not Pods?

10K Pods
5 Day Pod lifetime

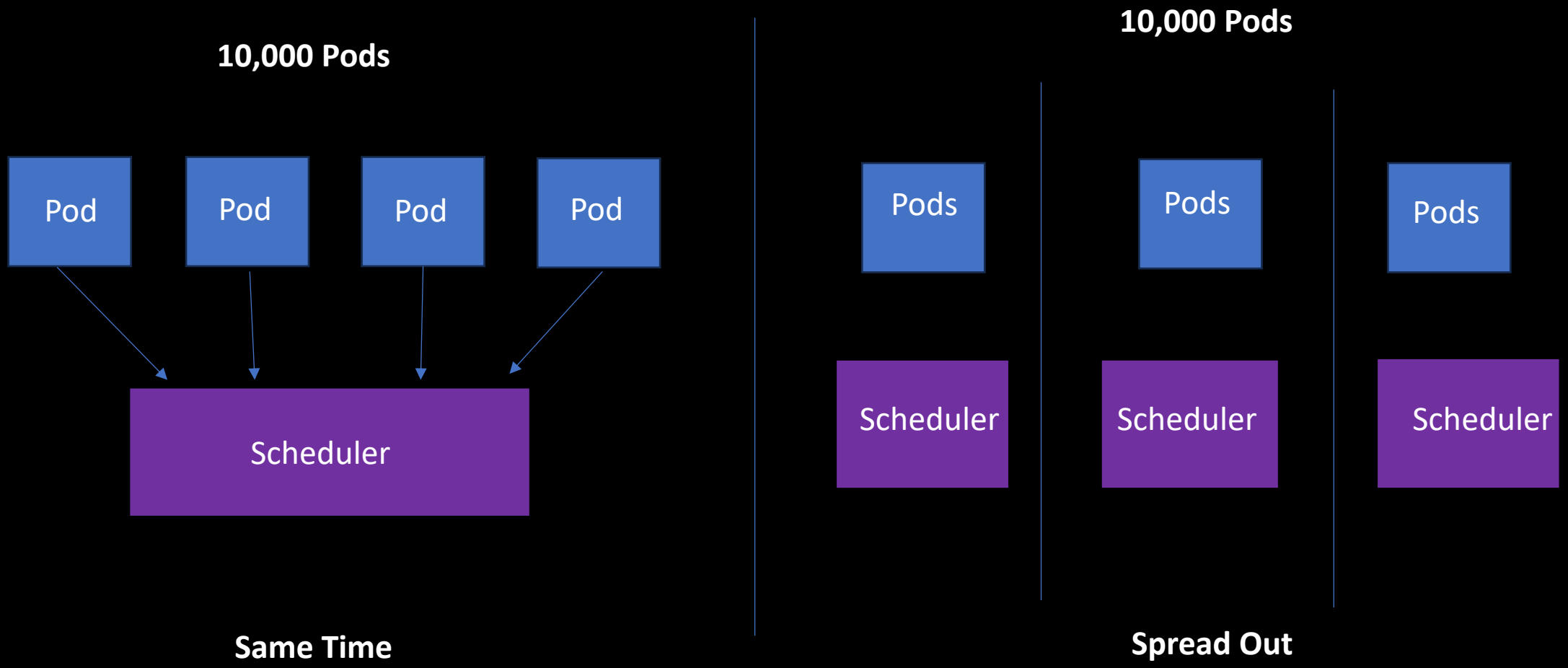


10K Pods
15 Min Pod lifetime



Node and Pods

Slow Down


10,000 Pods
Pod
Pod
Pod
Pod
Scheduler
Same Time
10,000 Pods
Pods
Pods
Pods
Scheduler
Scheduler
Scheduler
Spread Out

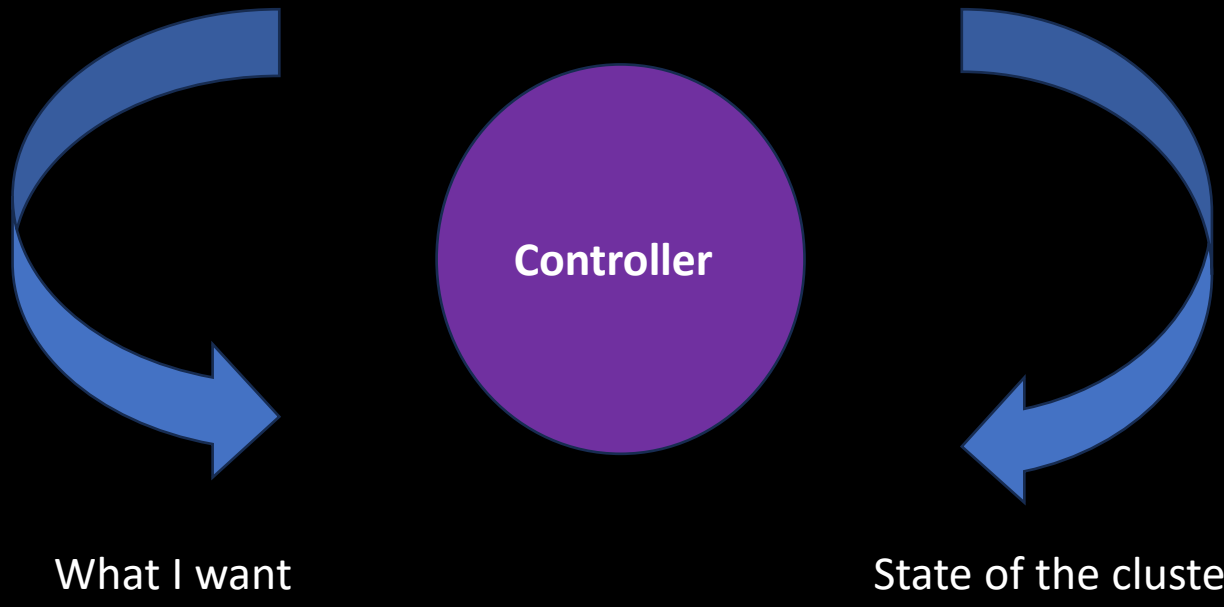
Volume of Change

API Server

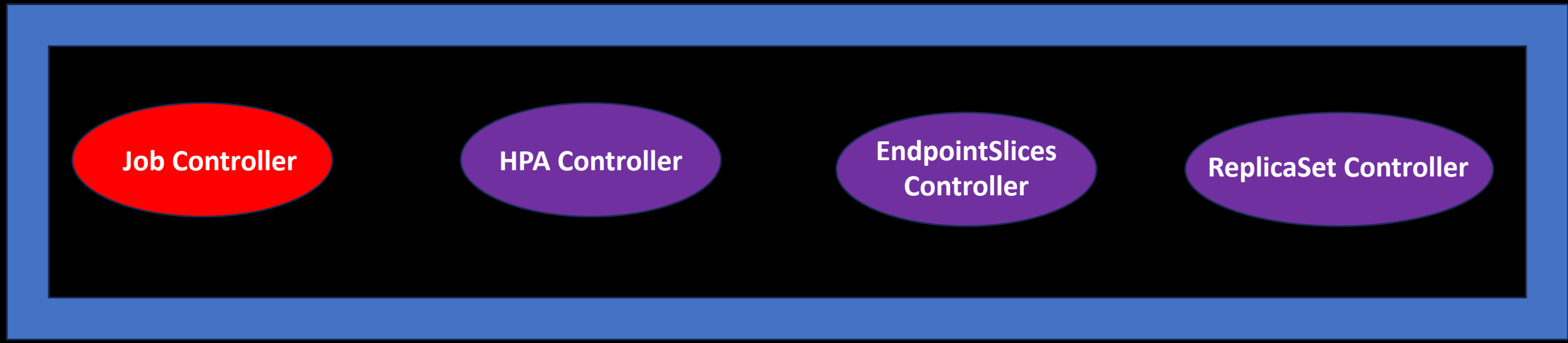
https://aws.amazon.com/blogs/containers/troubleshooting-amazon-eks-api-servers-with-prometheus/

Controller

Volume of Change

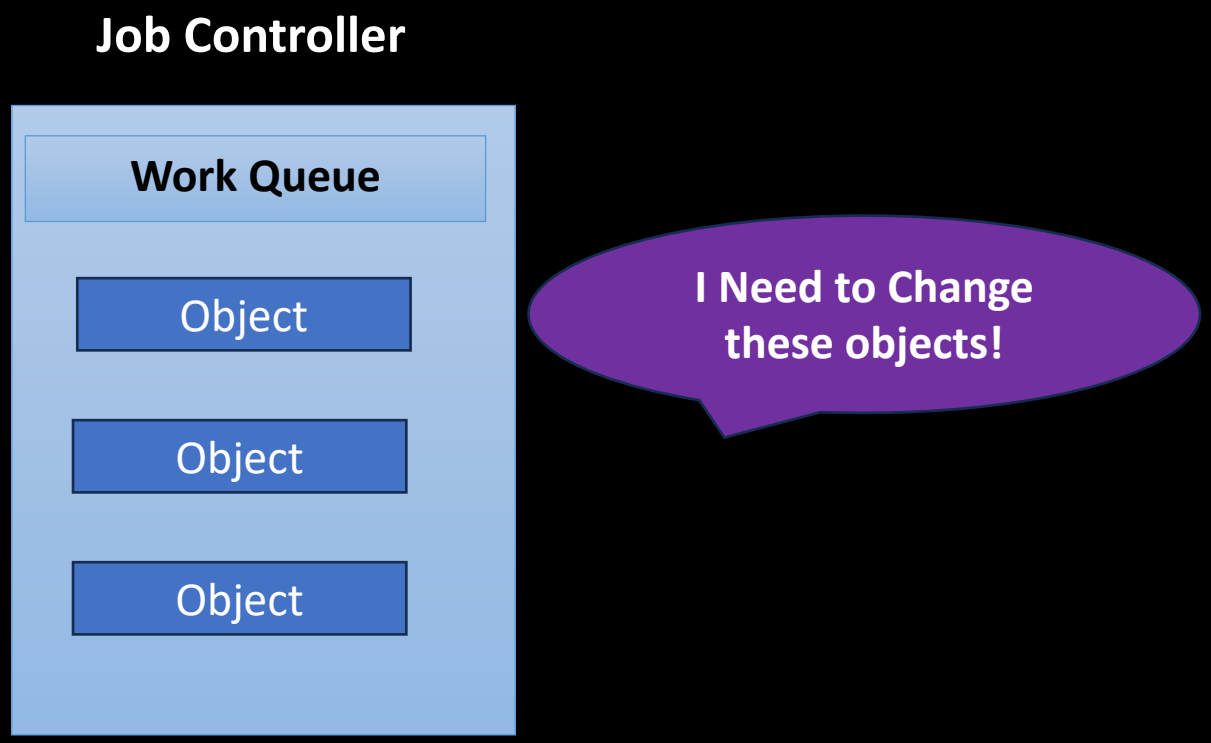


Kube Controller Manager

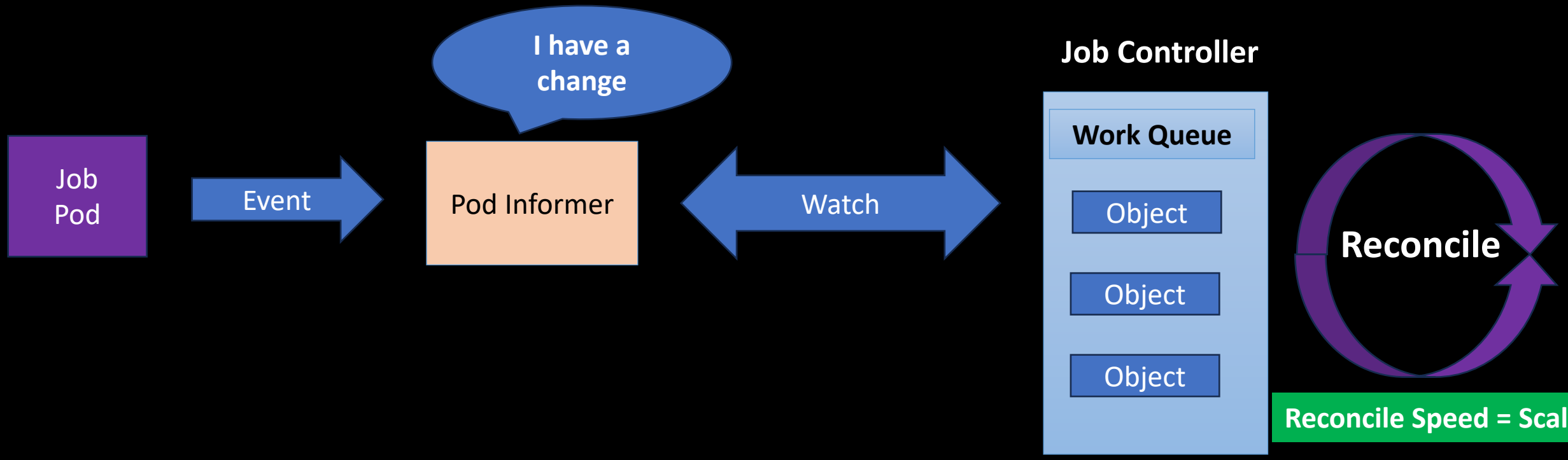


Think in Controller Volume

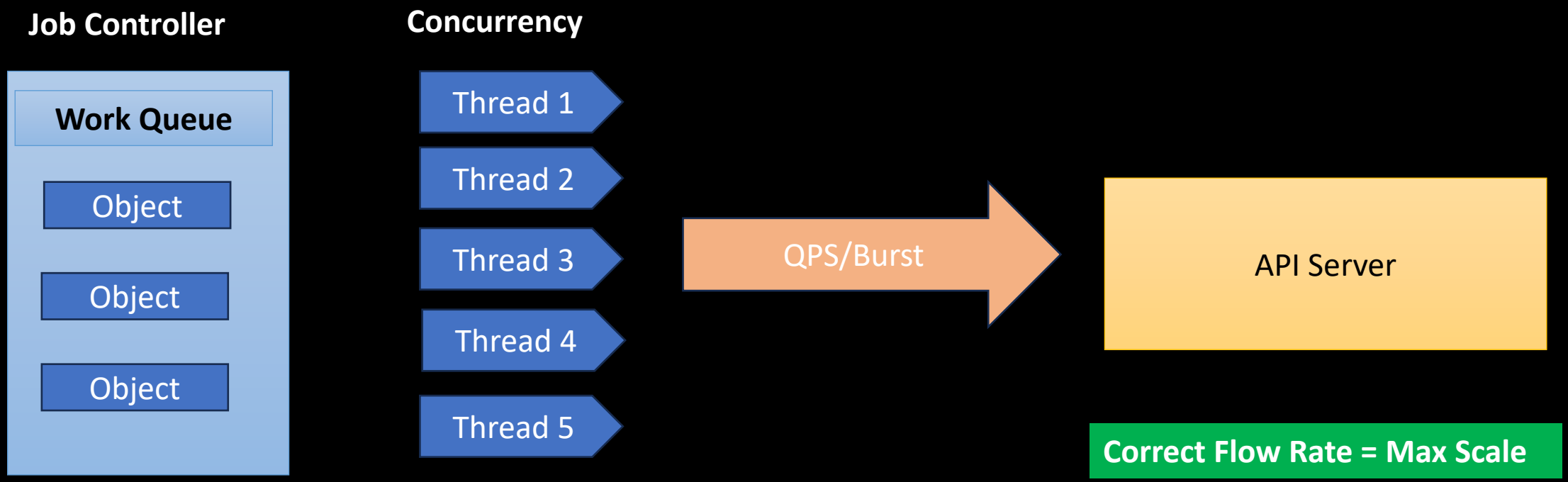
Looks like a Queue



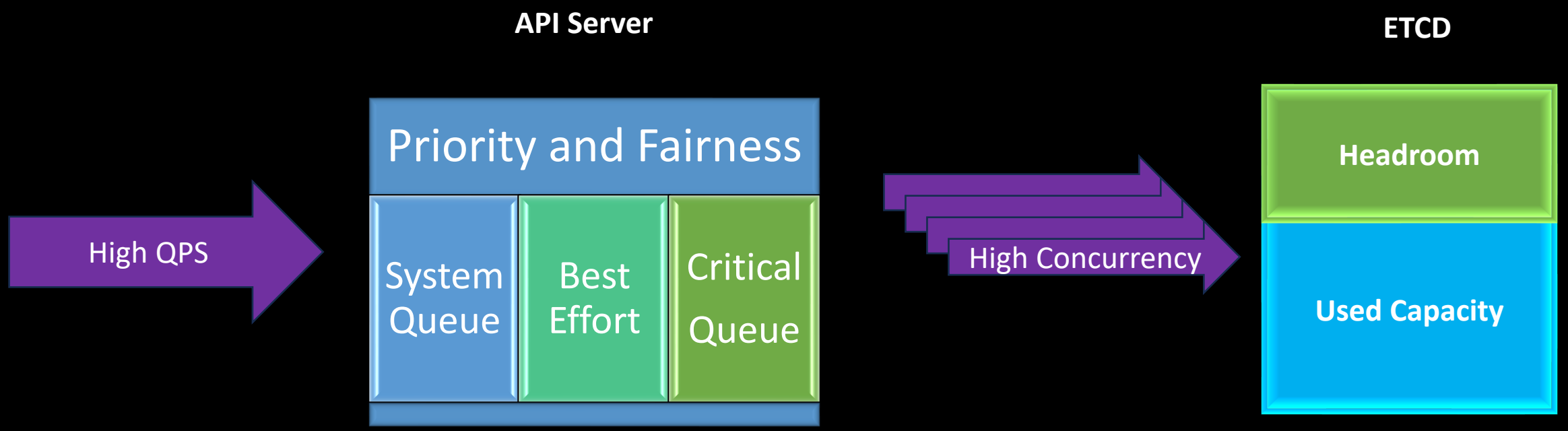
Kube Controller Manager



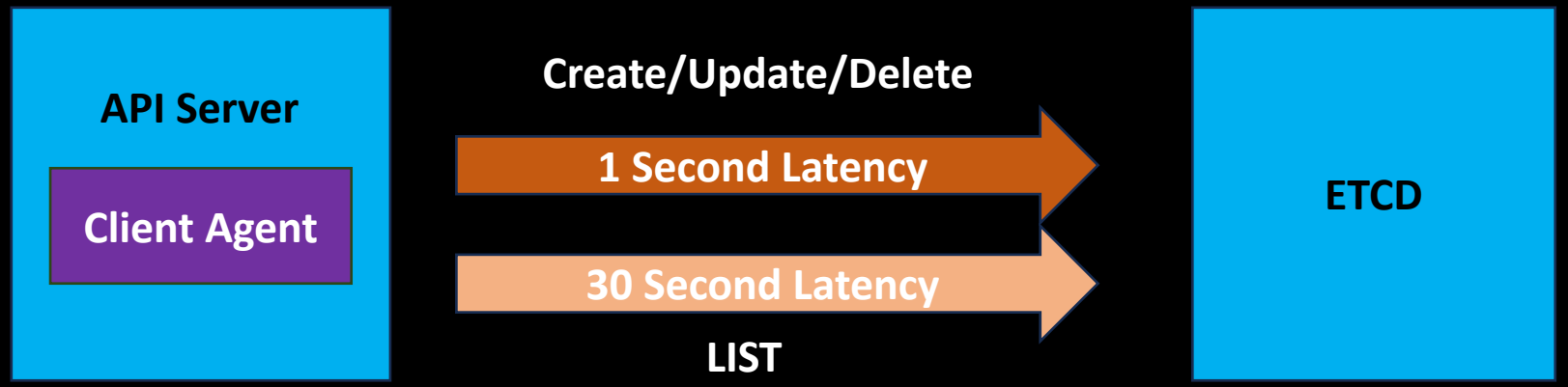
Controllers



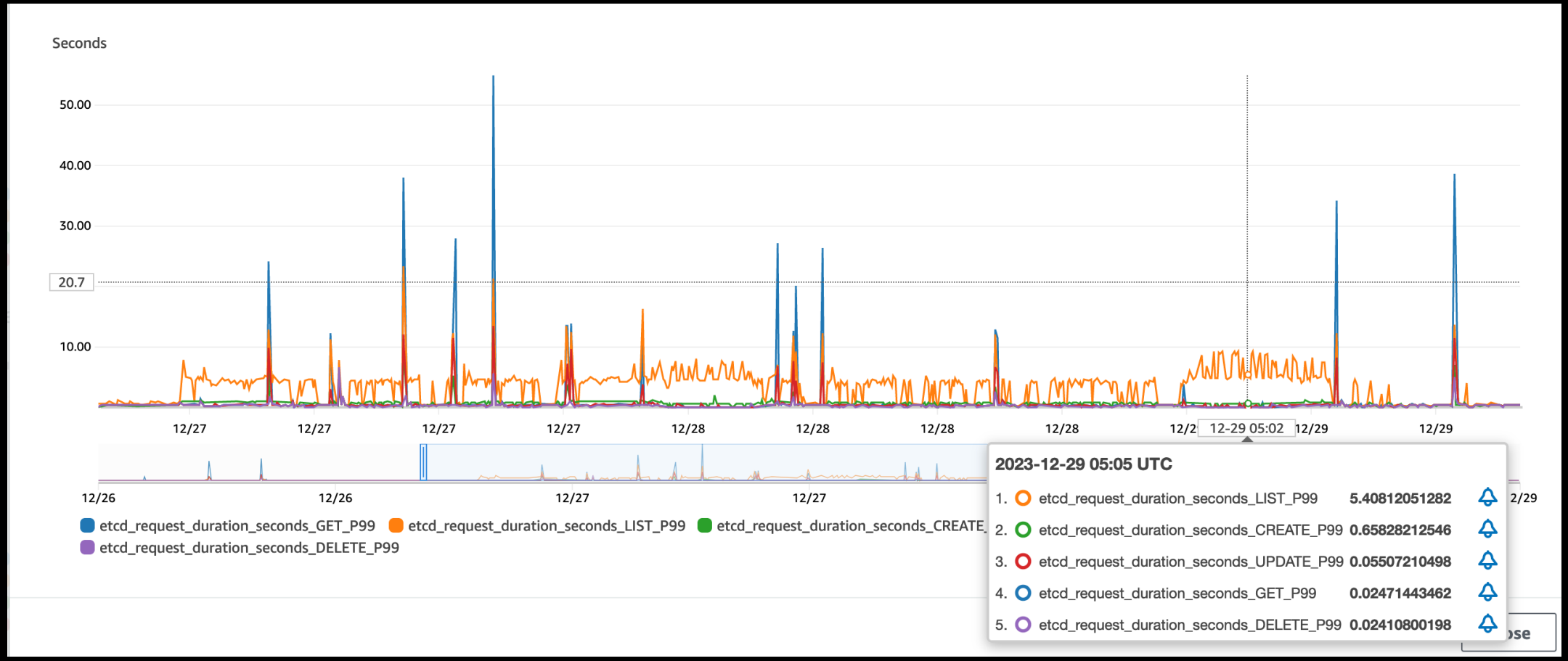
The Right Value?



What's The Target

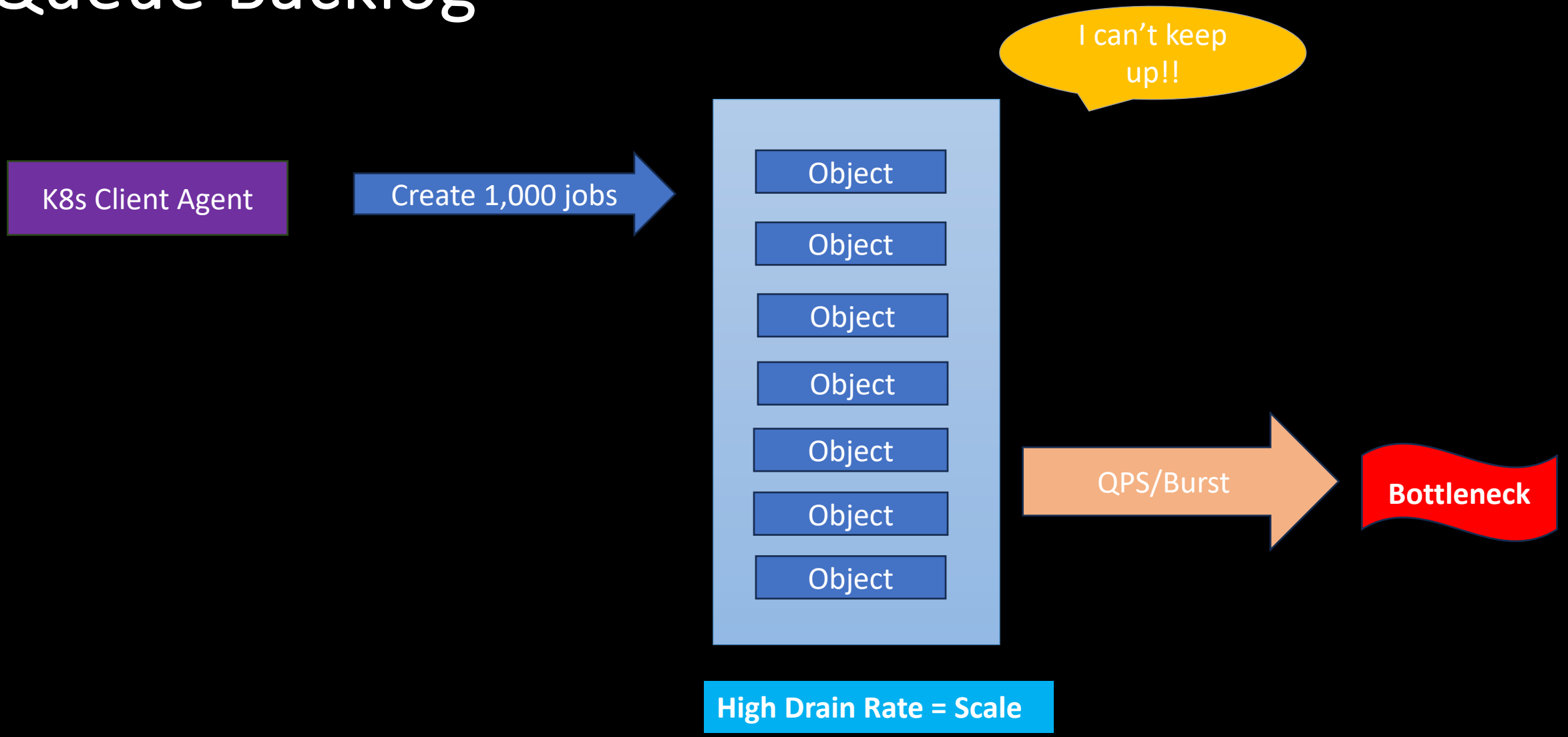


API to ETCD Latency

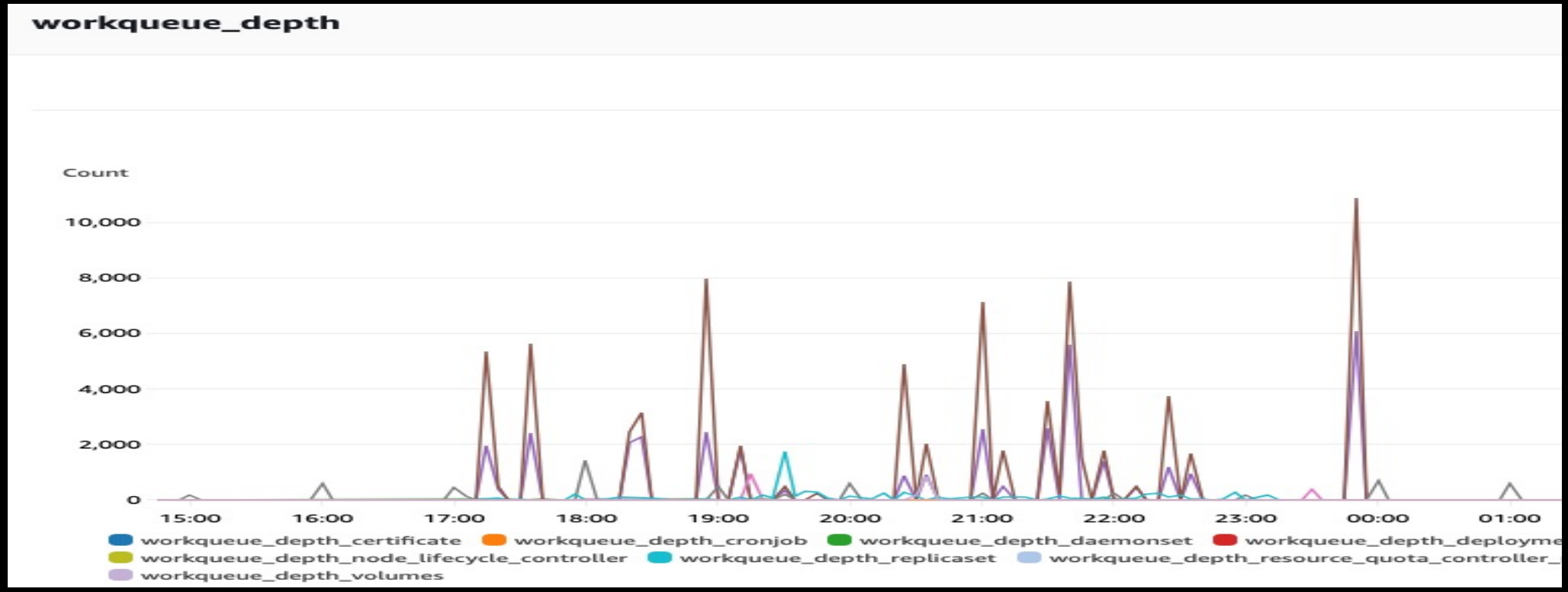


Which Controller is Sad?
(and why)

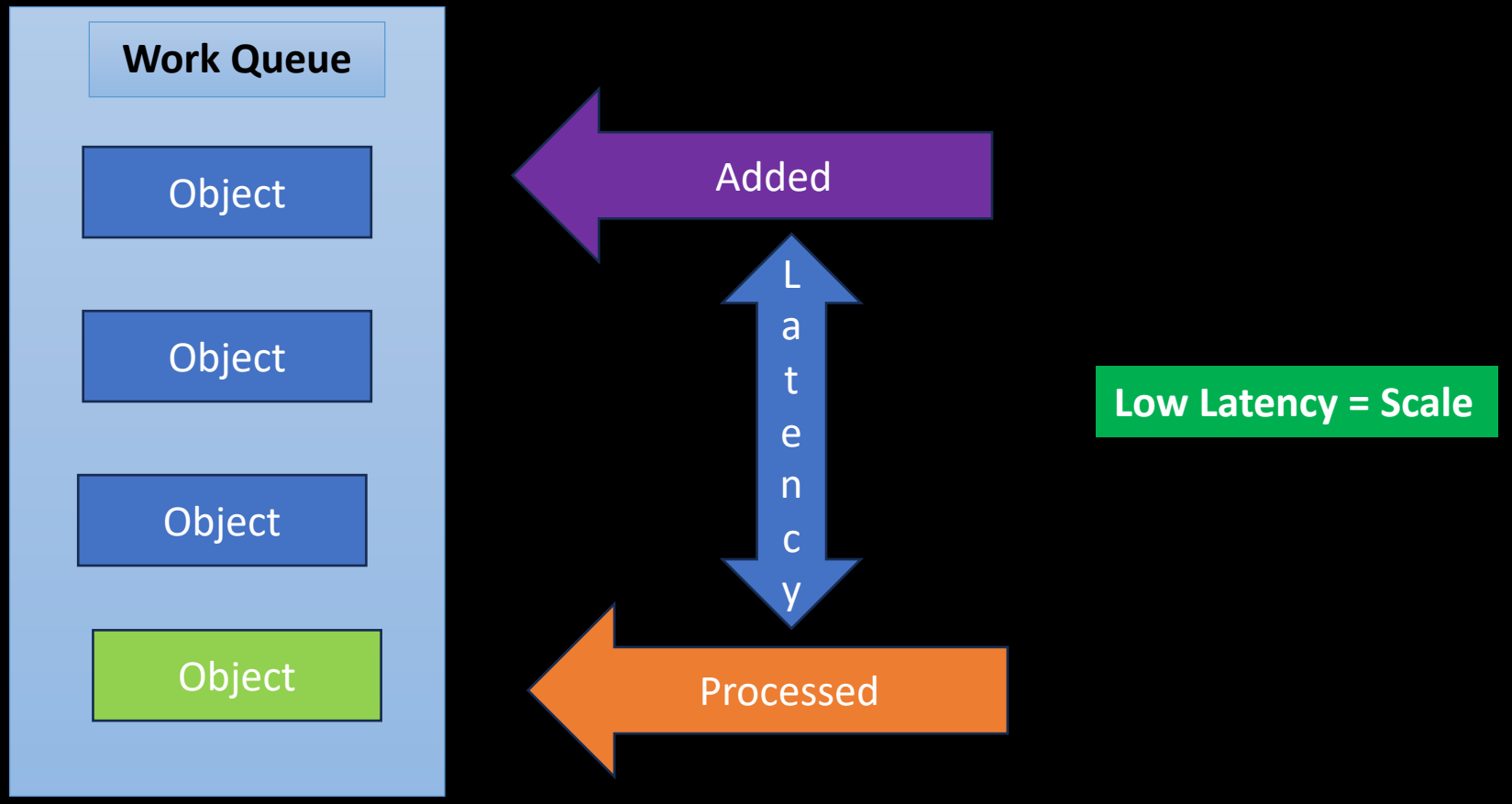
Queue Backlog



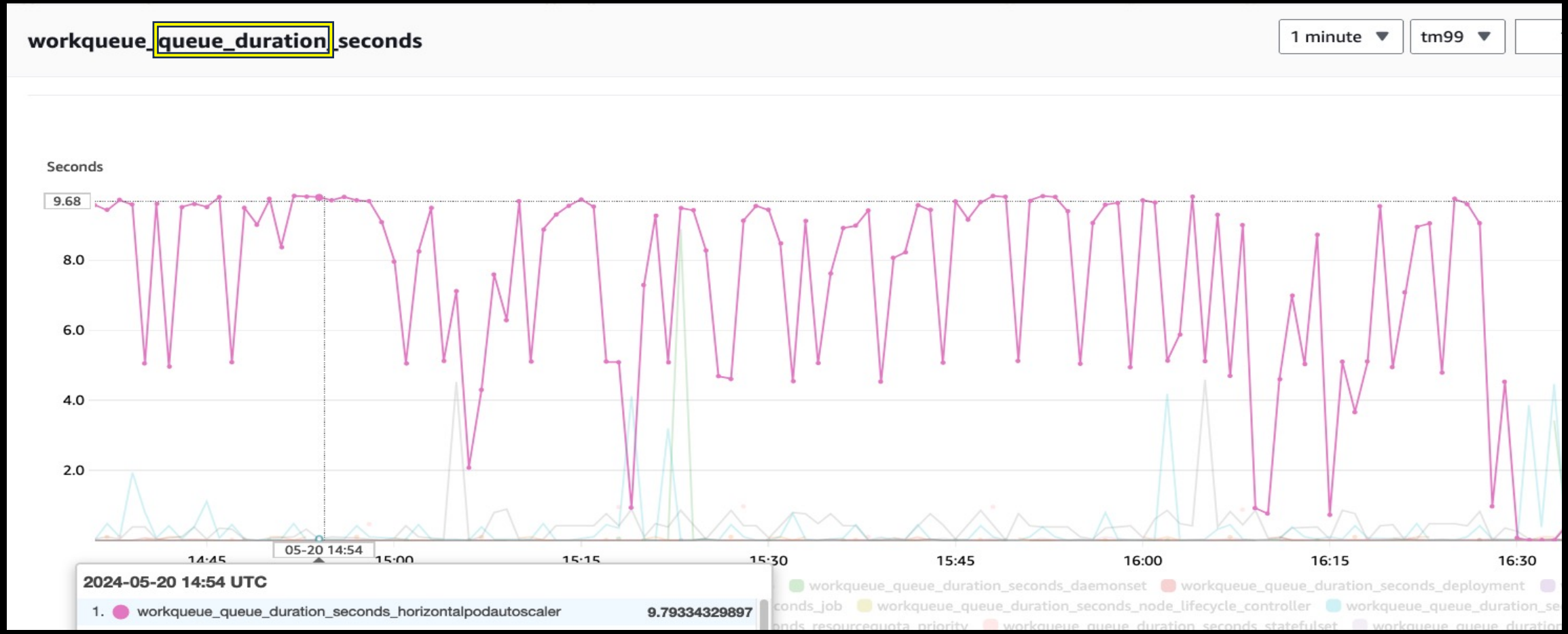
Work Queue Depth



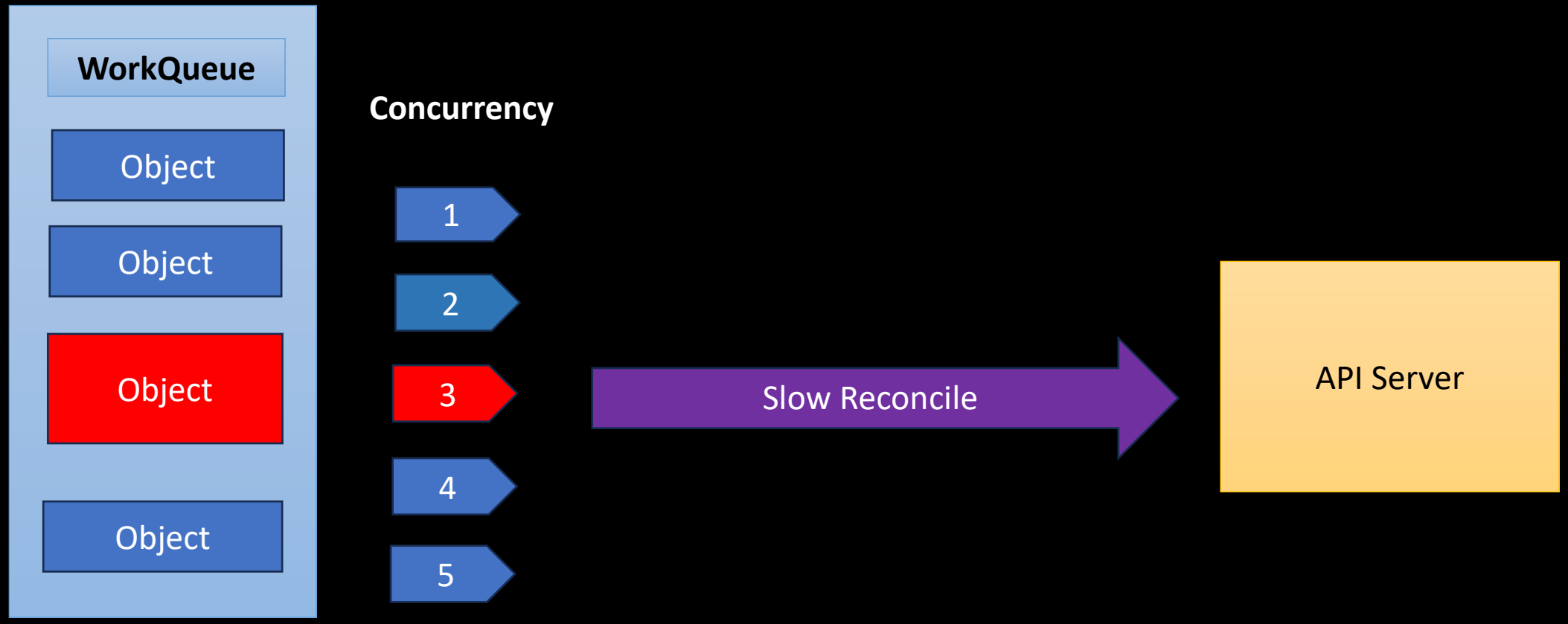
Request Latency



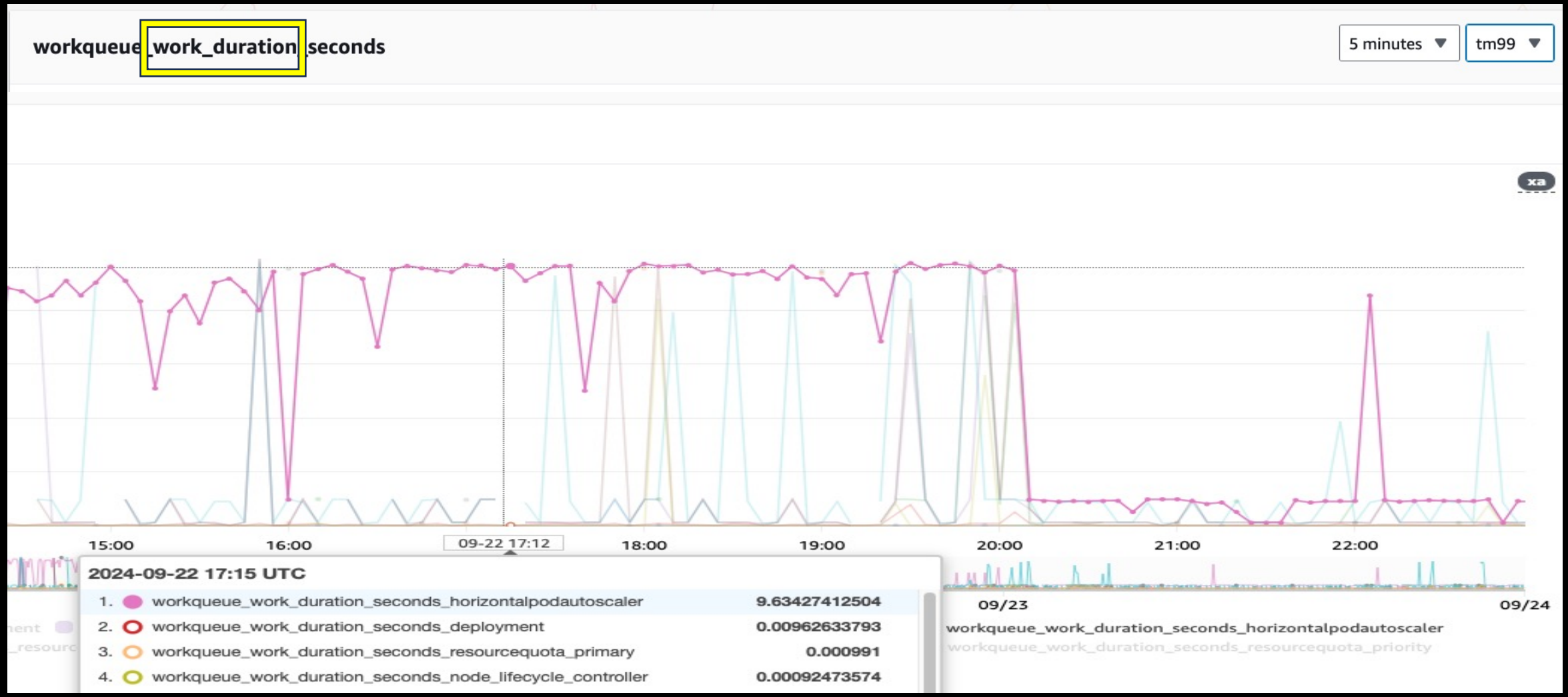
Trusting the Metrics



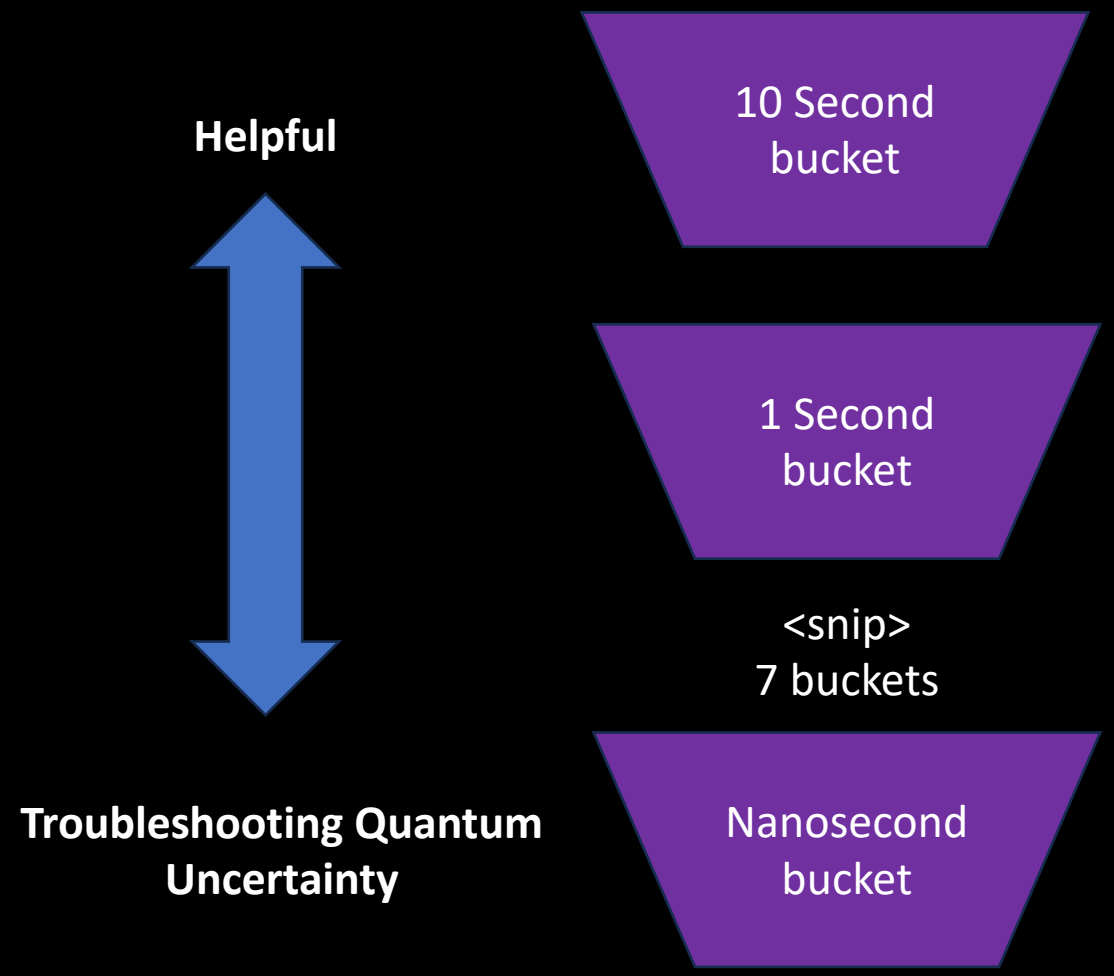
Slow Requests



Long Work Duration



Betrayed by Buckets



10 Seconds to 4 Minutes

sum/ count = AVG

What is Slow?

#	requestURI	userAgent	objectRef.re…	objectRe…	duration_in_sec
▶ 1	/api/v1/pods	datasources-agent/v0.0.0 (linux/amd64) kubernetes/$Format	pods		44.2675
▶ 2	/api/v1/pods	datasources-agent/v0.0.0 (linux/amd64) kubernetes/$Format	pods		42.8898
▶ 3	/api/v1/pods	datasources-agent/v0.0.0 (linux/amd64) kubernetes/$Format	pods		35.7841
▶ 4	/api/v1/pods	datasources-agent/v0.0.0 (linux/amd64) kubernetes/$Format	pods		35.6
▶ 5	/api/v1/pods	datasources-agent/v0.0.0 (linux/amd64) kubernetes/$Format	pods		35.062

Monitoring guide
https://aws.github.io/aws-eks-best-practices/scalability/docs/kcp_monitoring/
Queries
https://github.com/aws-samples/specialist-tam-container-dashboards/blob/main/troubleshooting-queries/CloudWatch-Logs-Troubleshooting-Queries.md#calculate-the-average-response-time-for-each-request-uri

The Unexpected

@timestamp	1717628797993
annotations.apiserver.latency.k8s.io/etcd	8.51405529s
annotations.apiserver.latency.k8s.io/response-write	4.82036605s
annotations.apiserver.latency.k8s.io/serialize-response-object	34.796534043s
annotations.apiserver.latency.k8s.io/total	1m0.002500633s
annotations.apiserver.latency.k8s.io/transform-response-object	2.518µs
annotations.authorization.k8s.io/decision	allow

Response-Write – Slow Client (Lack of Network, CPU)
Serialize-response-object – JSON Serialization (Protobuf)

Mentors

Shyam Jeedigunta

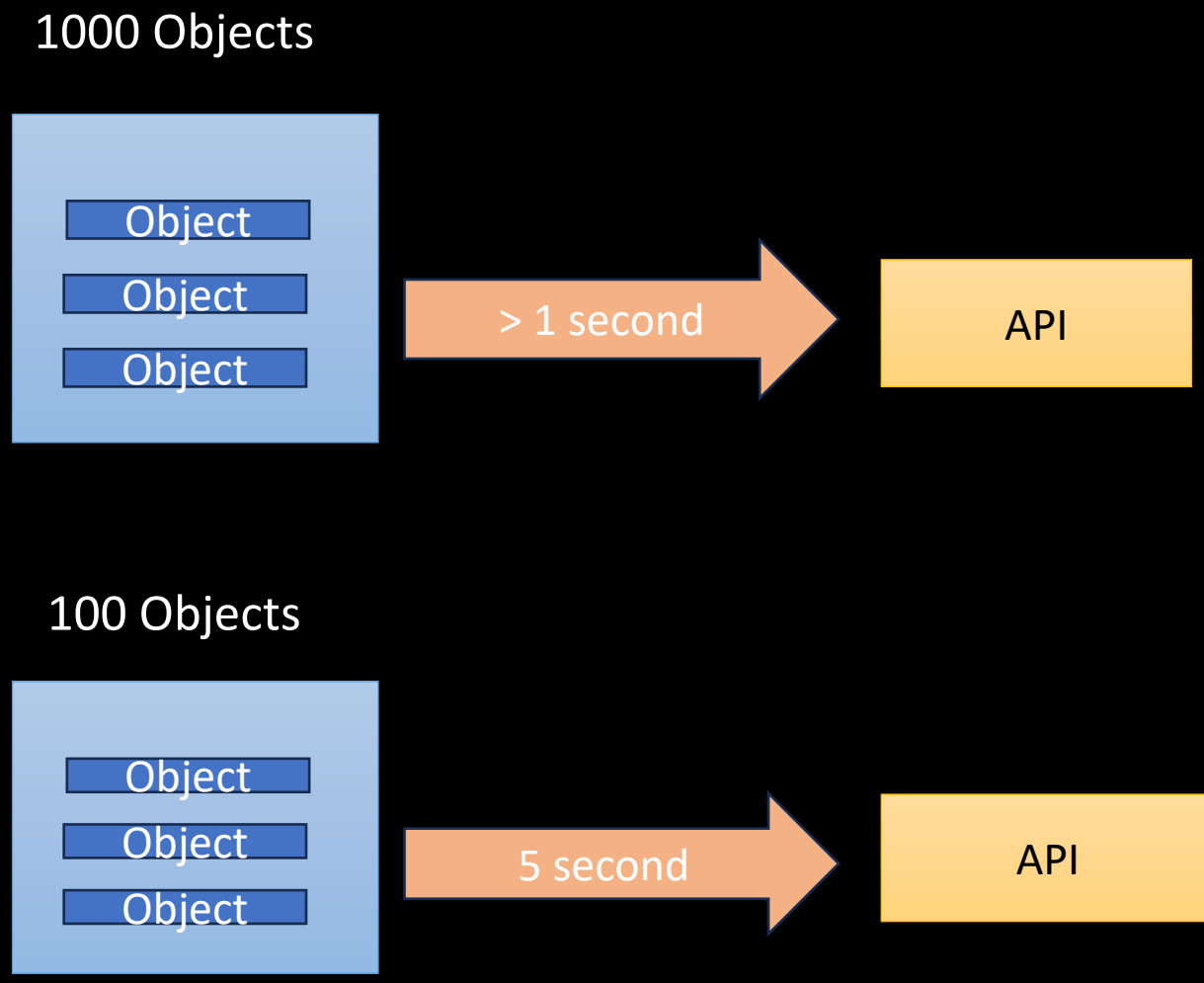
Chao Chen

Nathan Herz

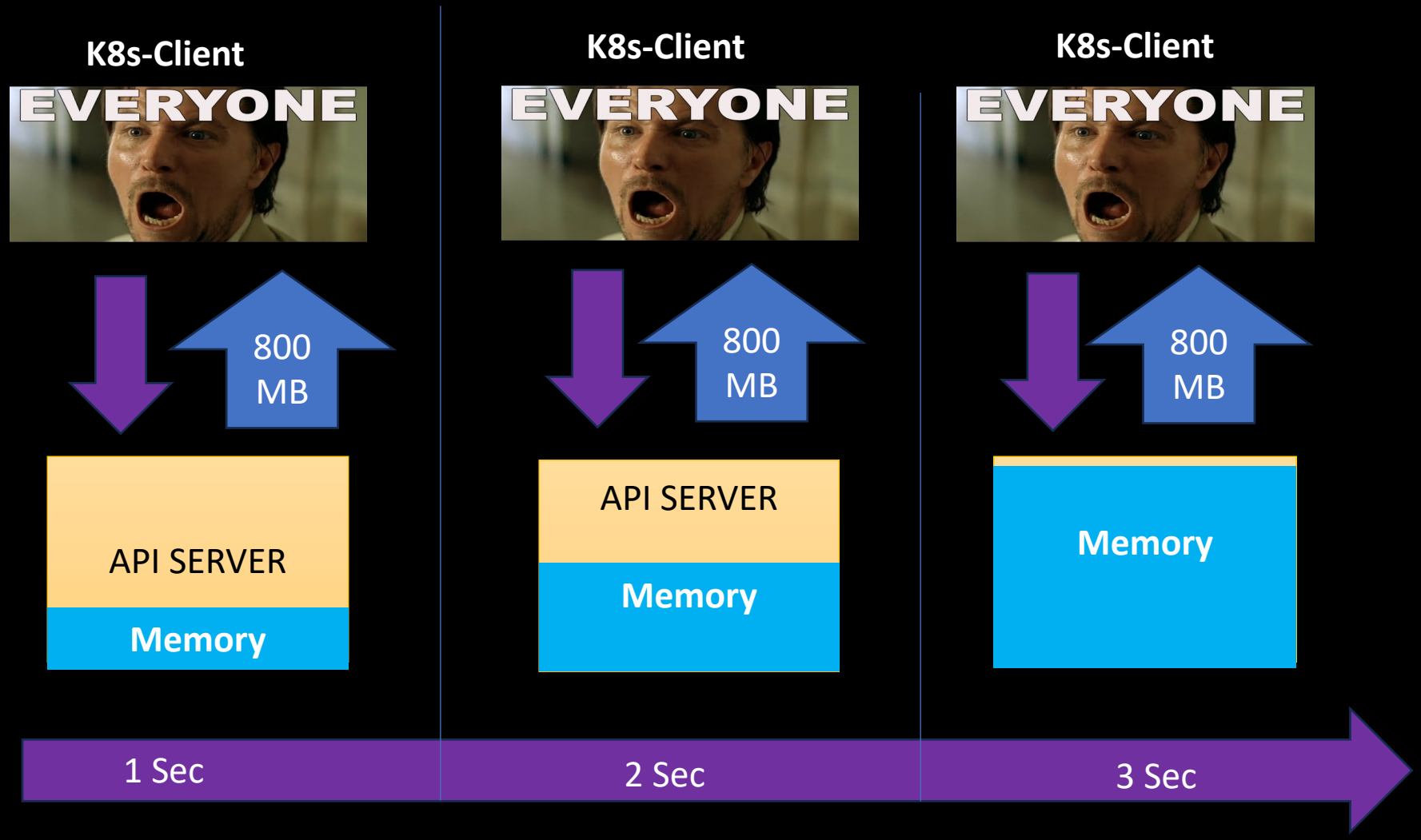
Harish Kuna

Latency Effects

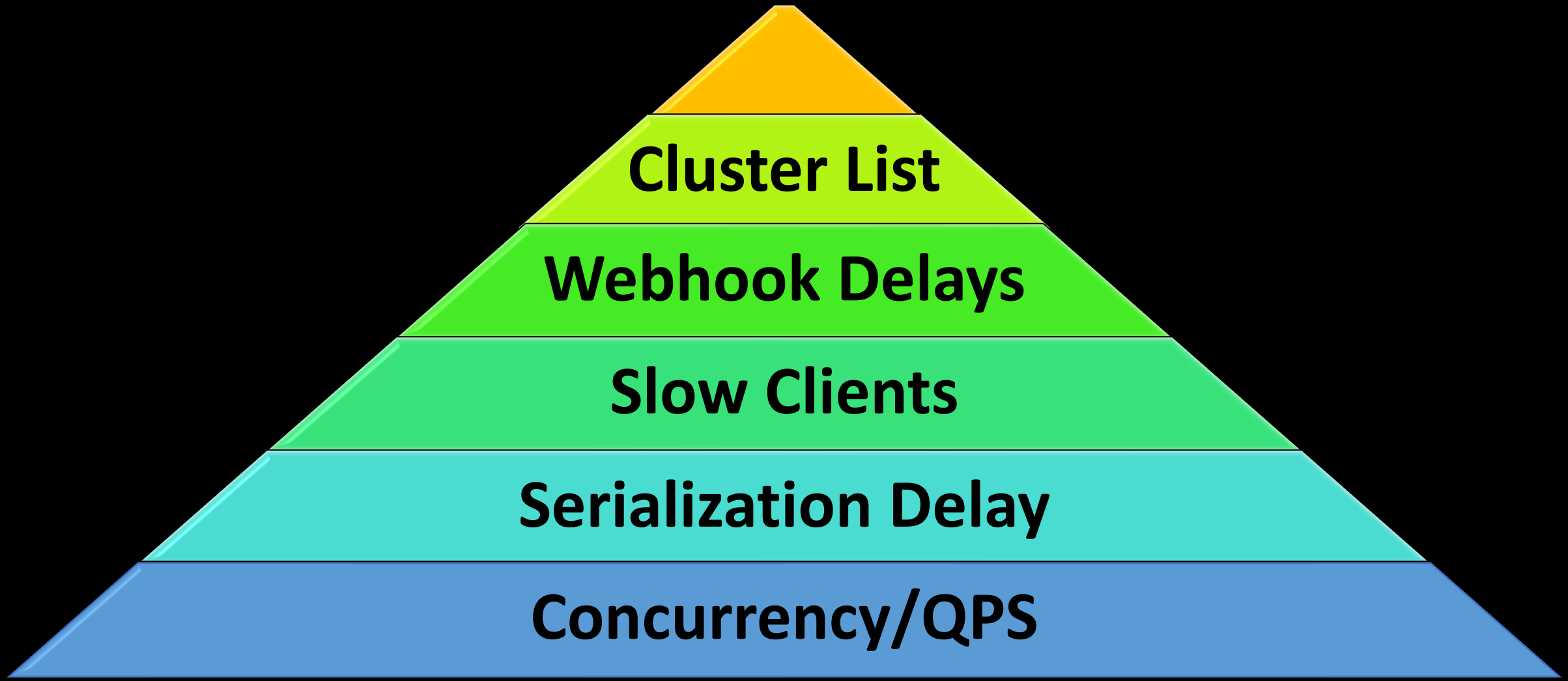
The max number of objects!



Workloads Cause Latency



Pyramid of Problems



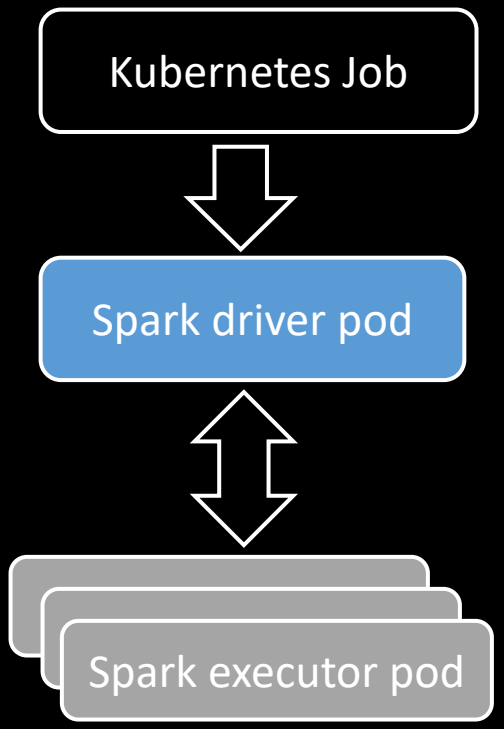
№1 Cluster Reduction Trick

hosting providers hate him

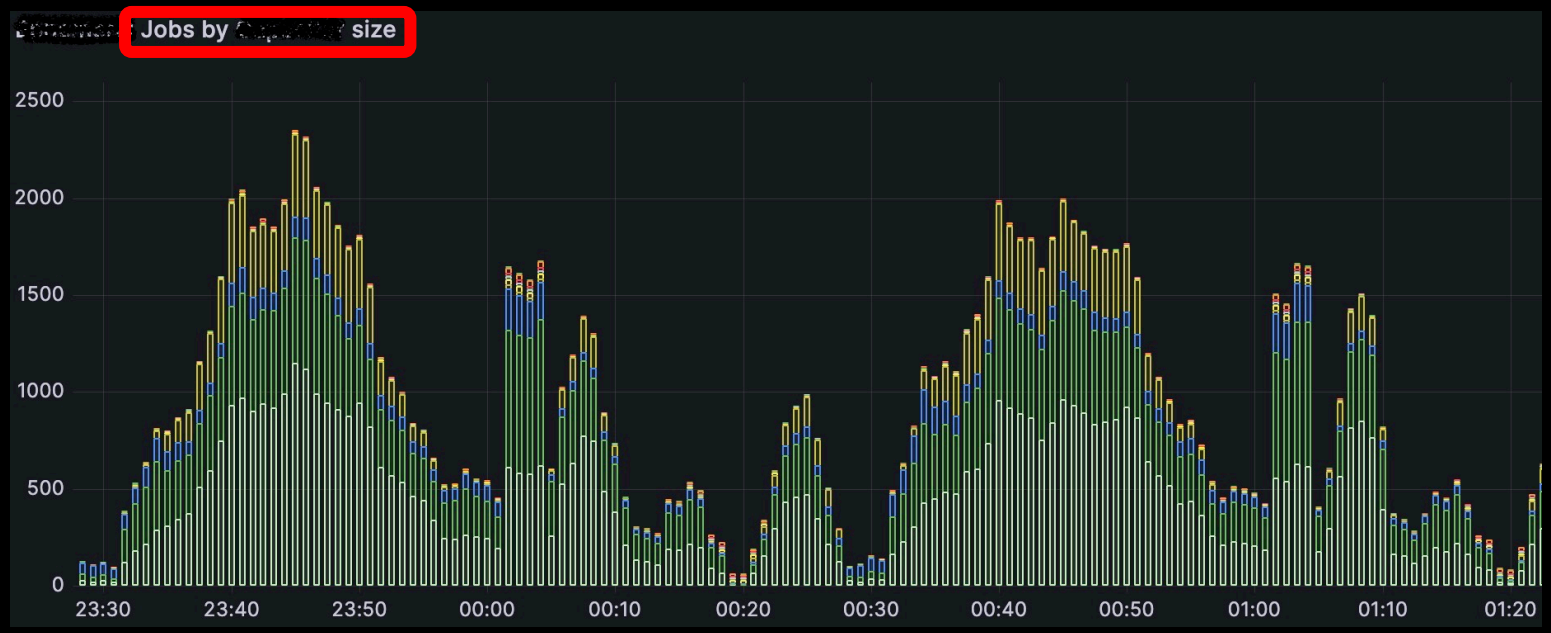
Custom `kube-scheduler` ⇨ `NodeResourcesFit: MostAllocated` ⇨ **50%** Cluster Size

The Mischievous Workload

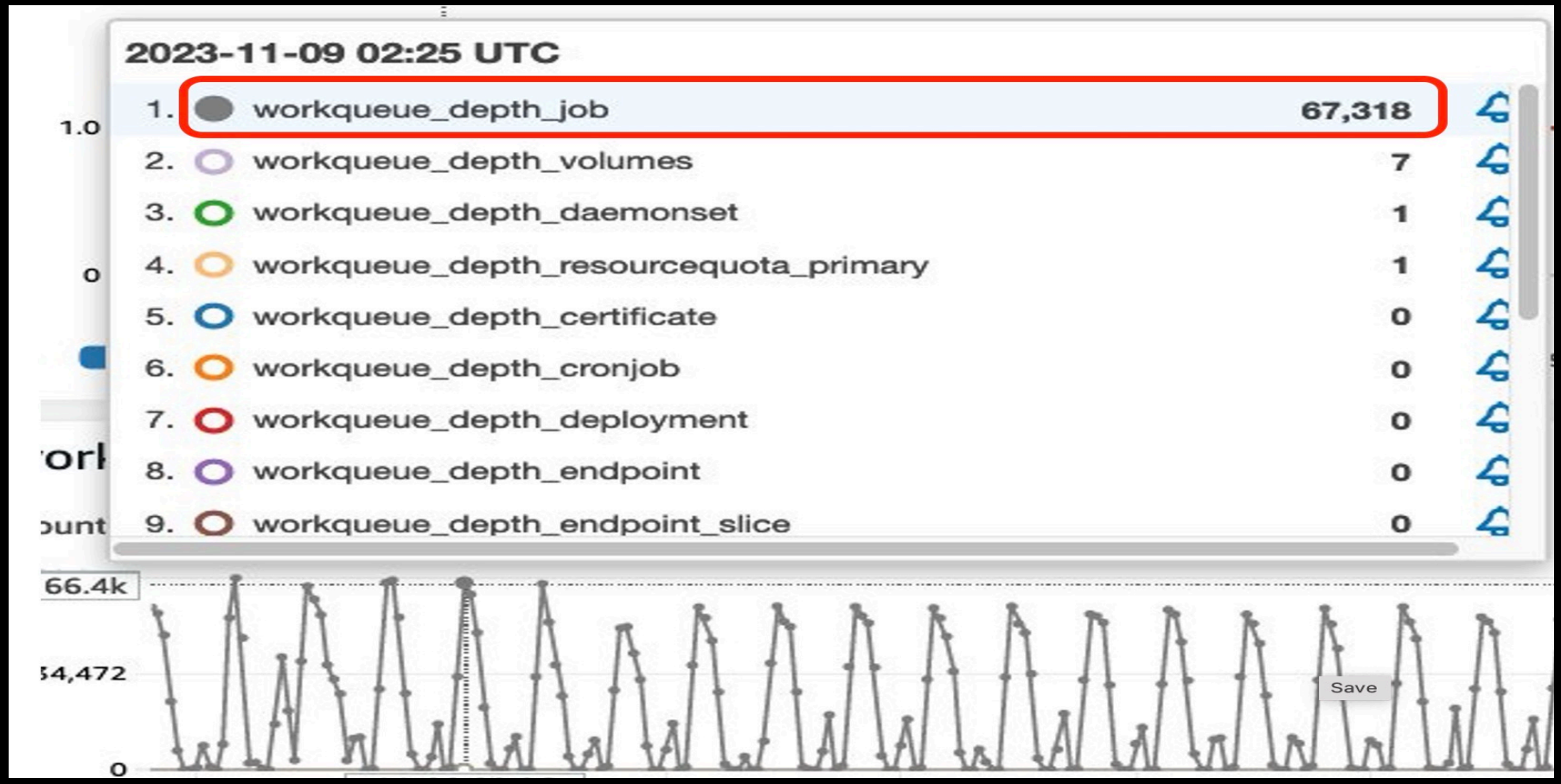
Spark Jobs anatomy



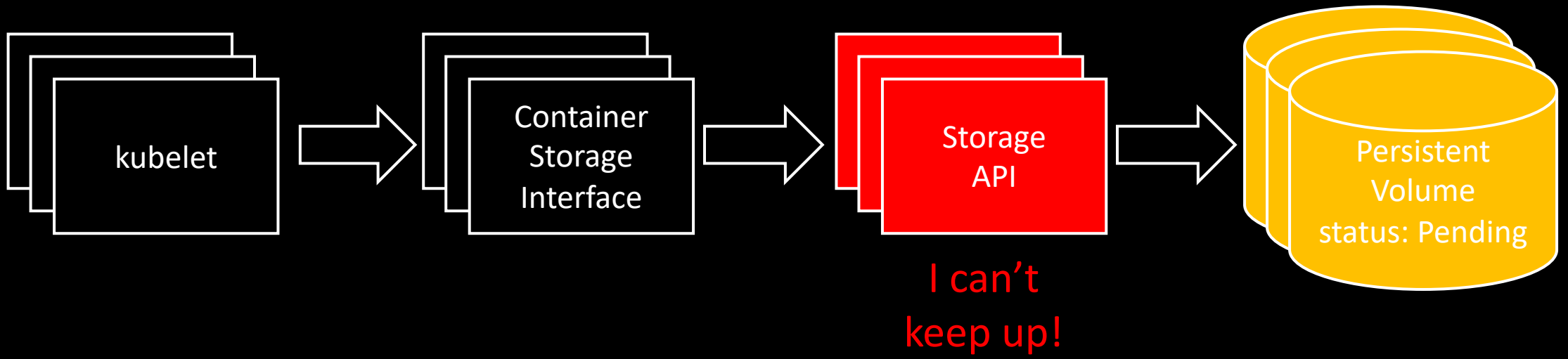
Spark Jobs scale



Control Plane Catastrophe



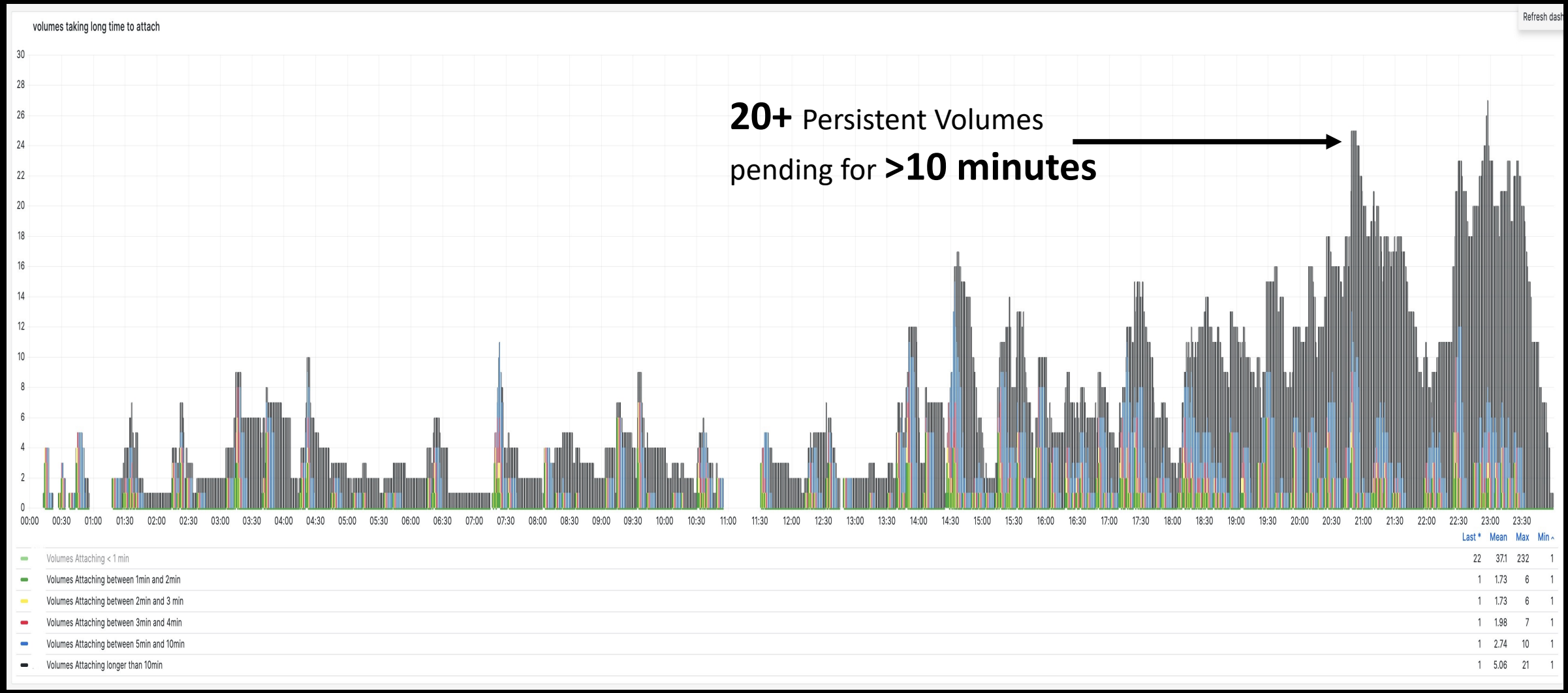
1,000 Persistent Volumes
(PER SECOND)



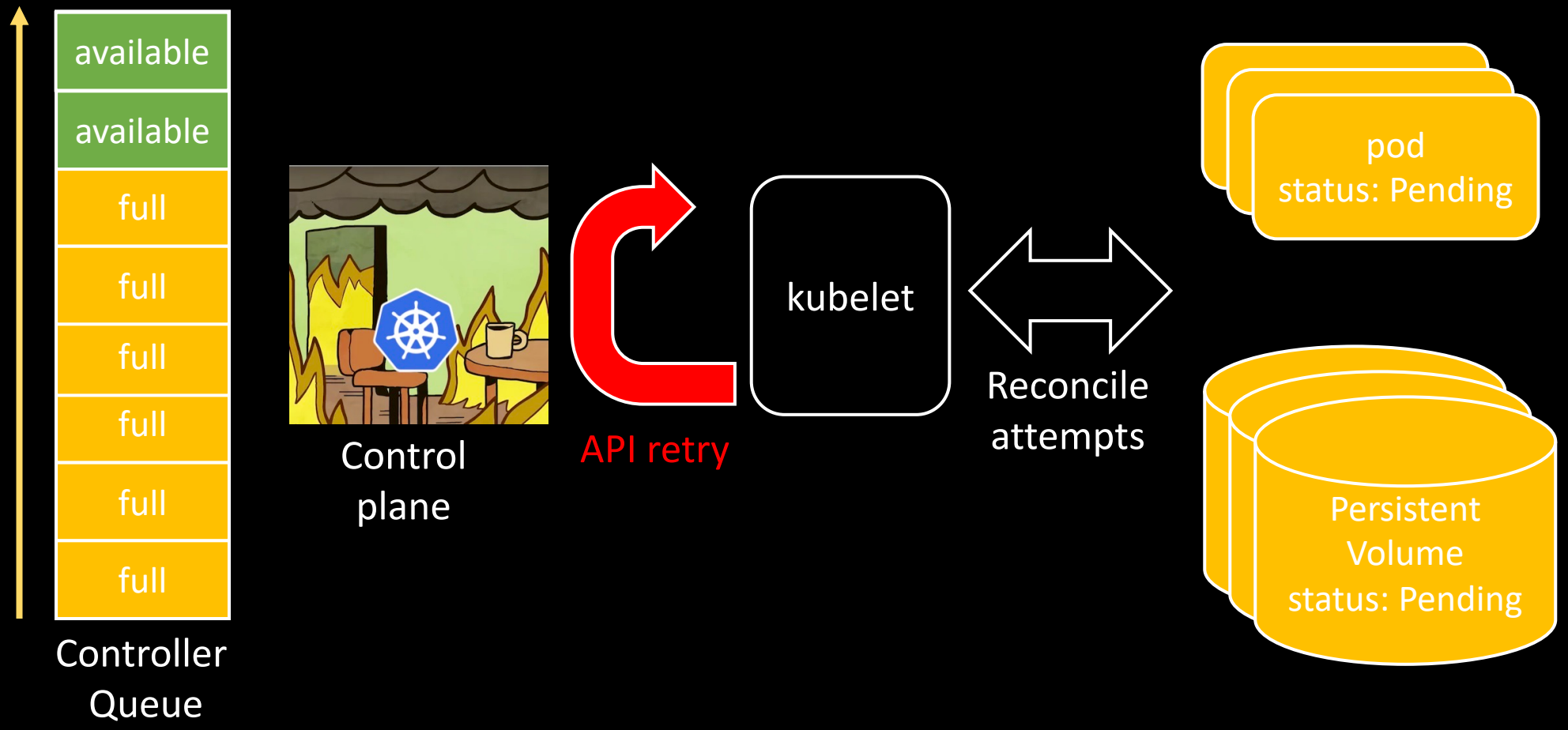
CSI Problem: Logs

```
kcm created an event at [redacted] -
AttachVolume.Attach failed for volume "pvc-567770b1-b2f9-
4807-9707-21aab54c37b8" : timed out waiting for external-
attacher of [redacted] CSI driver to attach volume
[redacted]
```

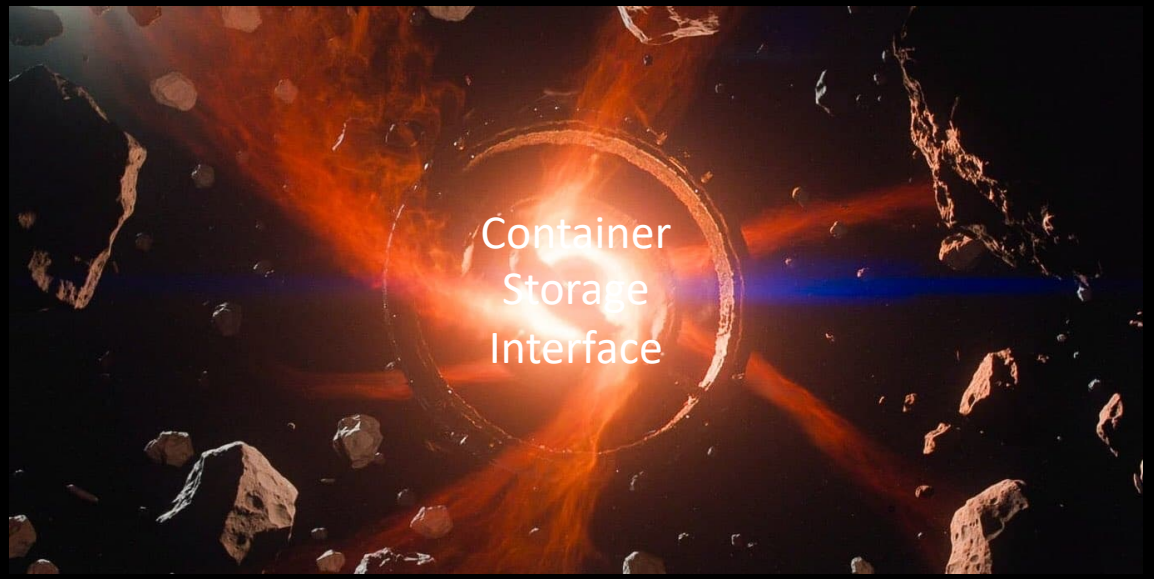
CSI Problem: Not Very Scary Graph

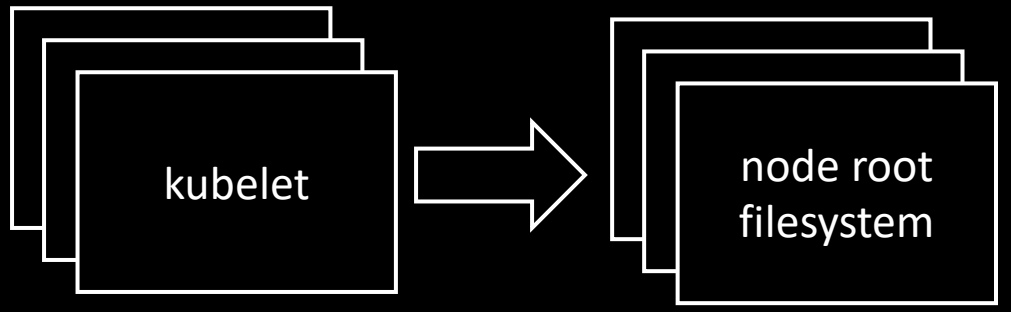


(Not Just) CSI Problem

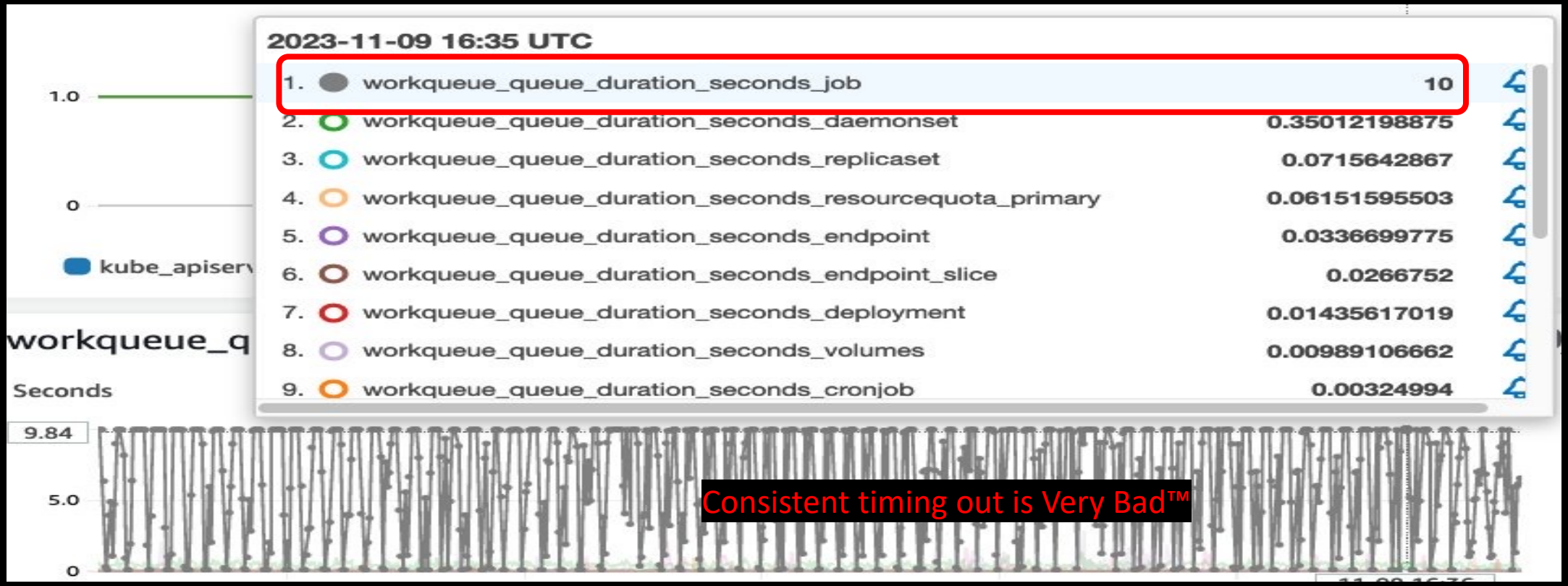


No CSI – No Problem





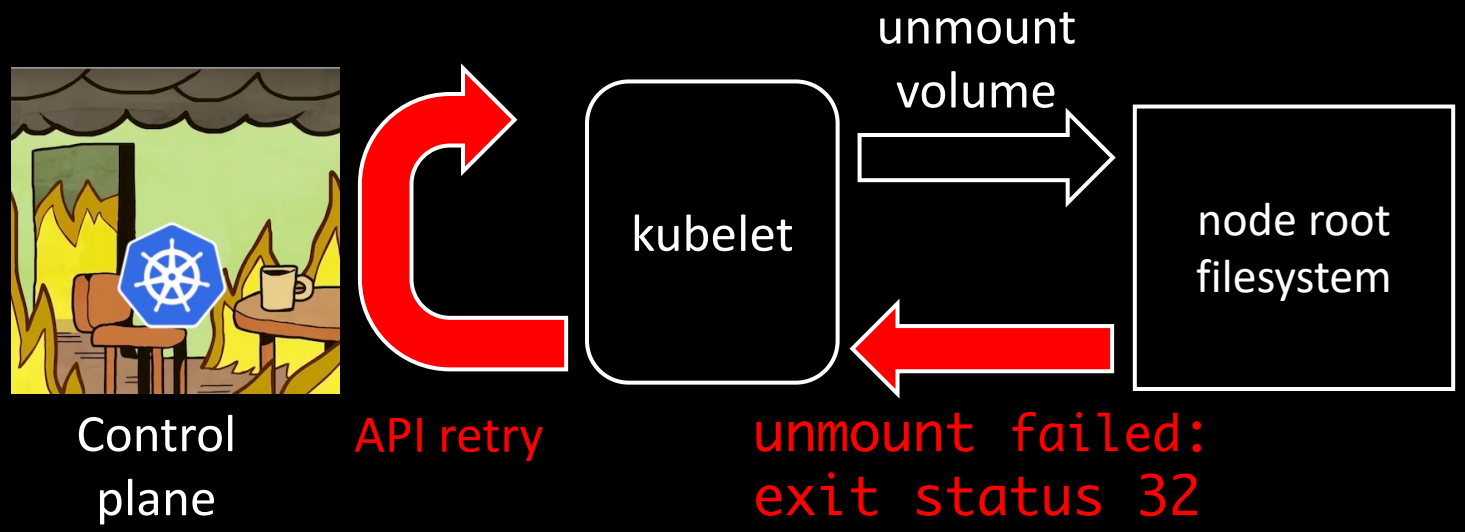
No CSI – ~~No Problem~~ Different Problem

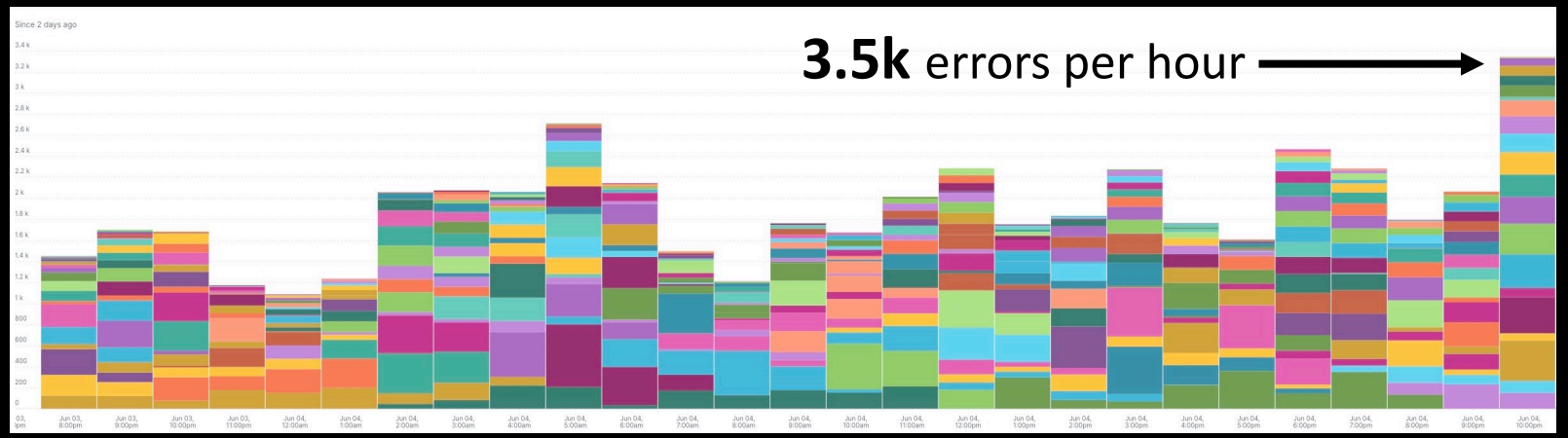


Error 32

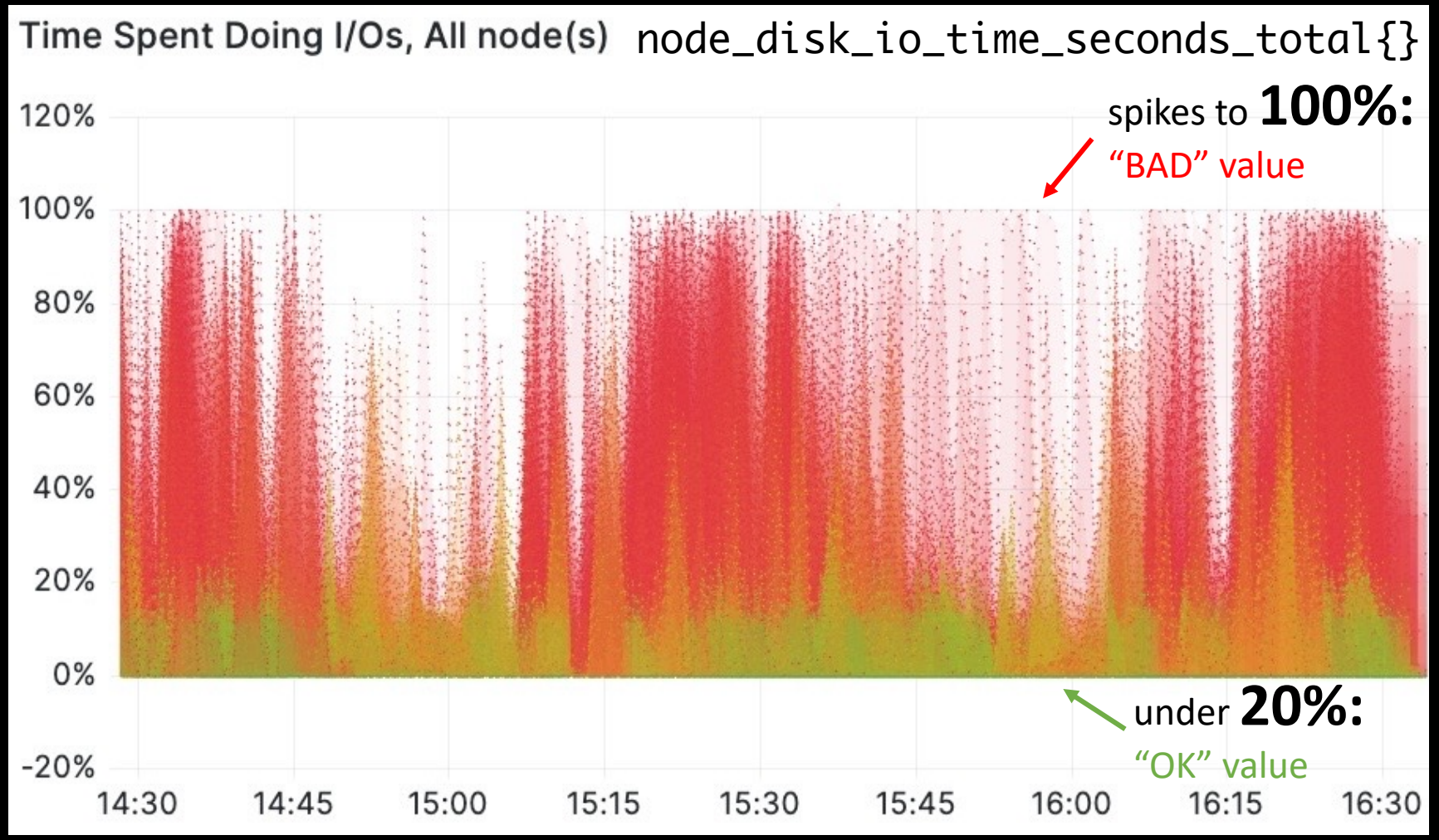
```
Error: UnmountVolume.TearDown failed for volume "kube-api-access-bc5sr" (UniqueName: "kubernetes.io/projected/693f4a5e-f879-47af-baae-6235cccd667b-kube-api-access-bc5sr") pod "693f4a5e-f879-47af-baae-6235cccd667b" (UID: "693f4a5e-f879-47af-baae-6235cccd667b") : unmount failed: exit status 32
```

~~The Good,~~ the Bad, and the Control Plane



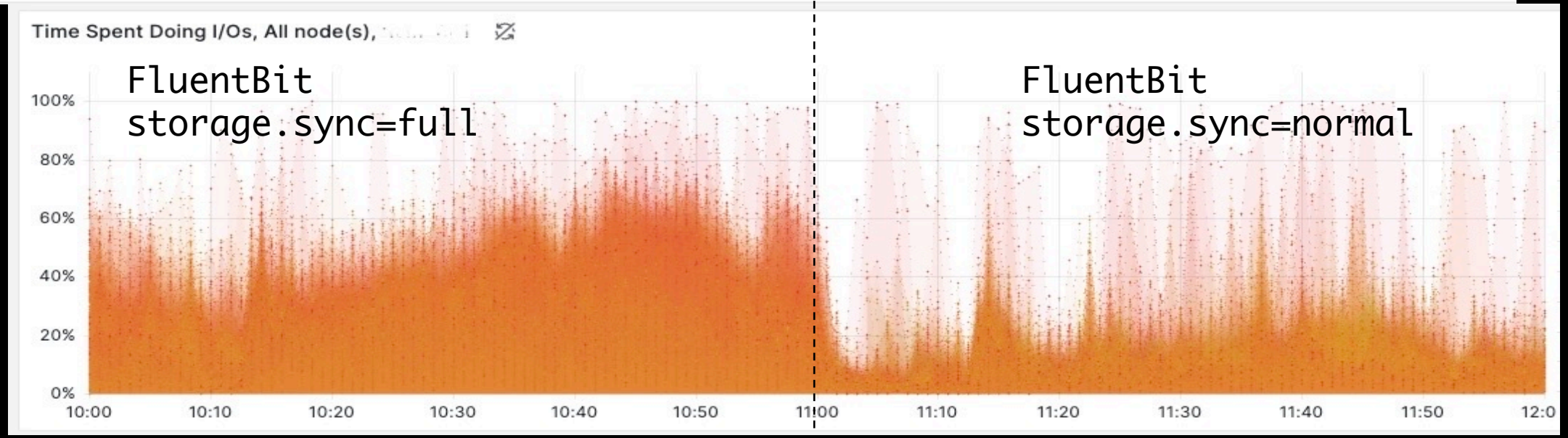


Storage: Red Alert



The Truth Is ~~Out There~~ At The Node Level

2x
reduced

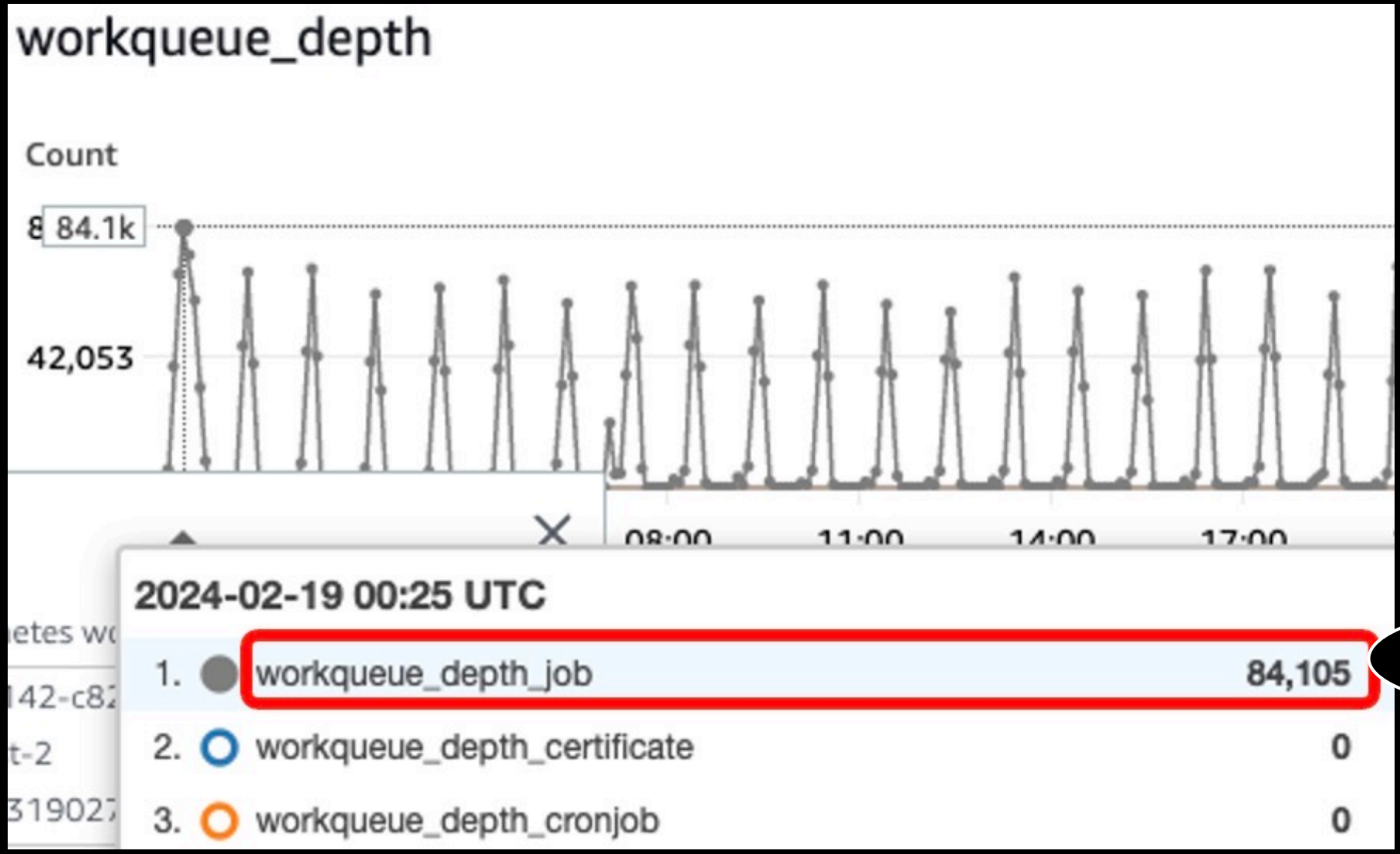


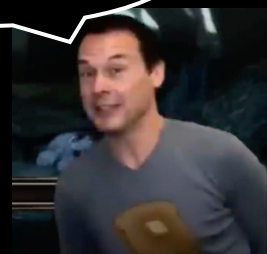
Storage Goes Brrrrr

```
unmount failed:
exit status 32
```

Control plane

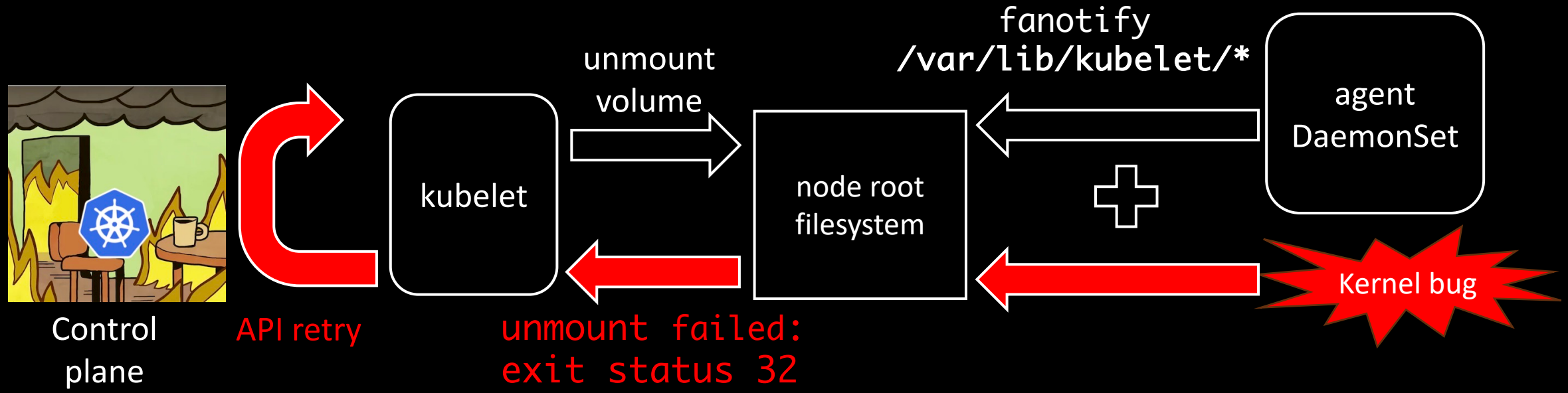
Control Plane: New Low



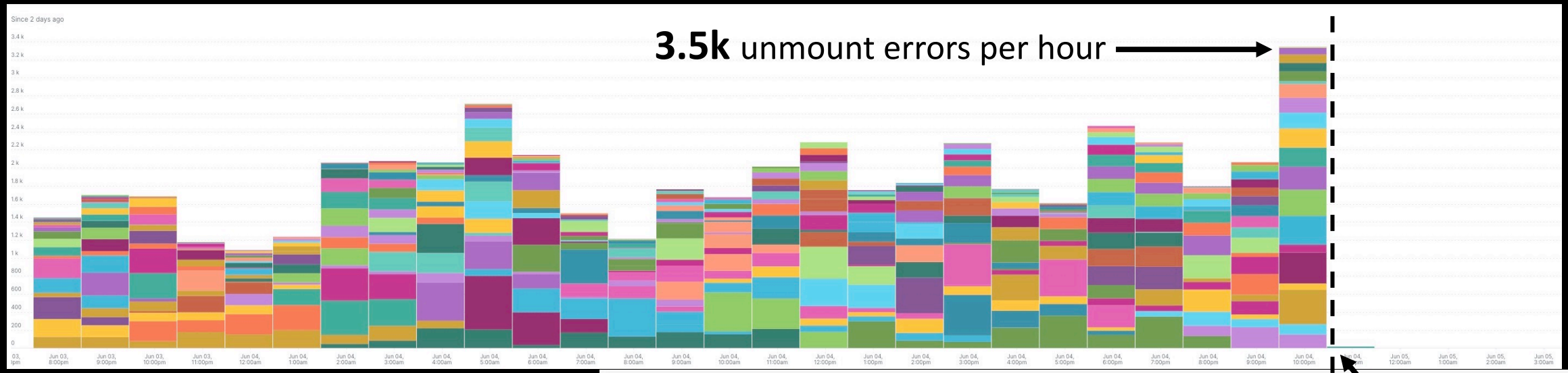


Job to Pod Delay

It Cannot be a DaemonSet



It was a DaemonSet

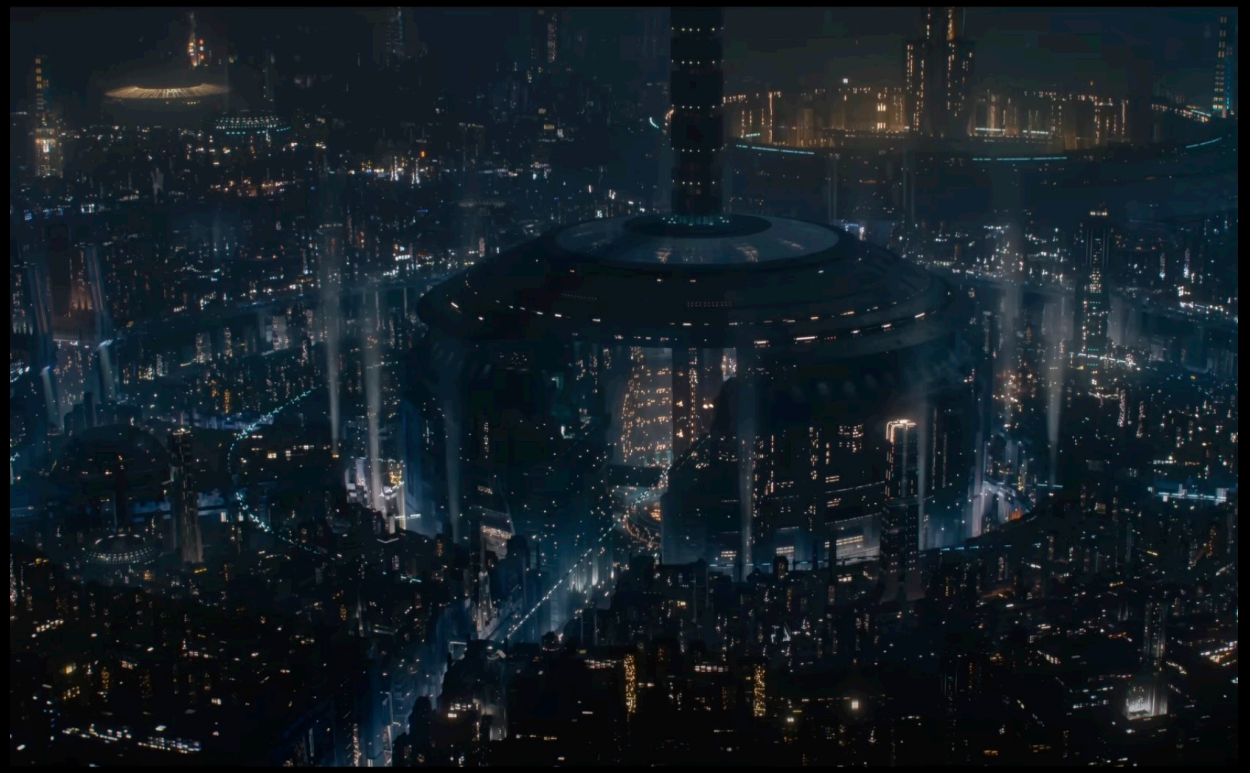


`/var/lib/kubelet/*` **excluded** from the agent config

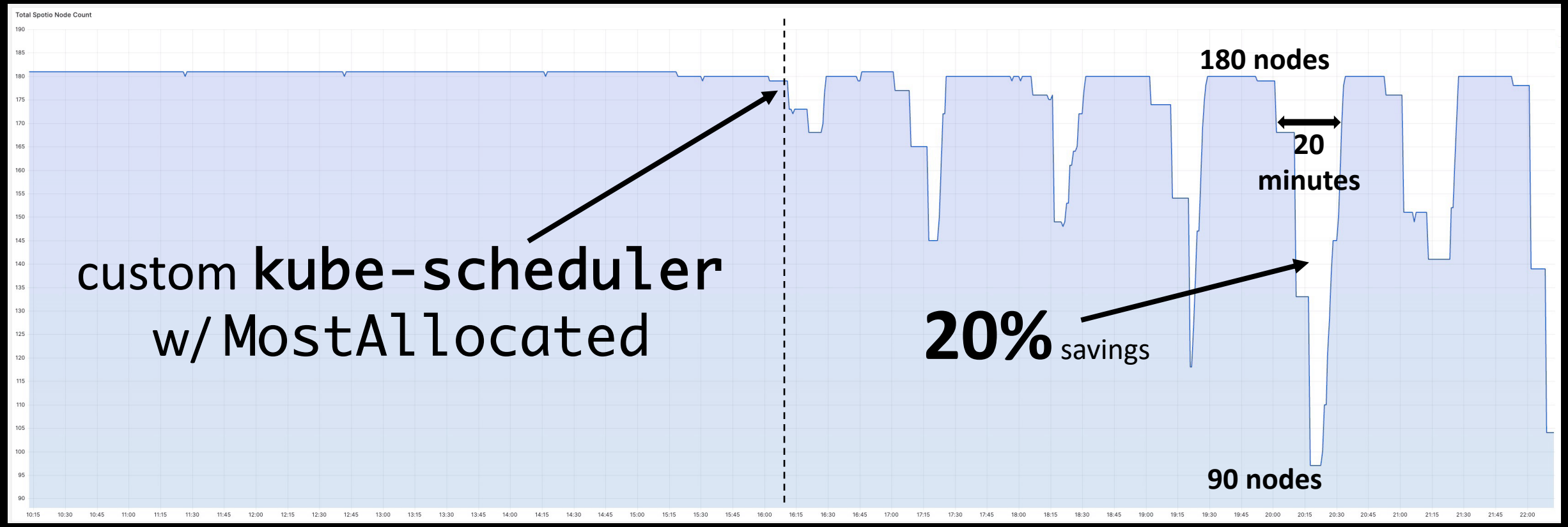
Custom `kube-scheduler`: Efficiency

`LeastAllocated` (default)

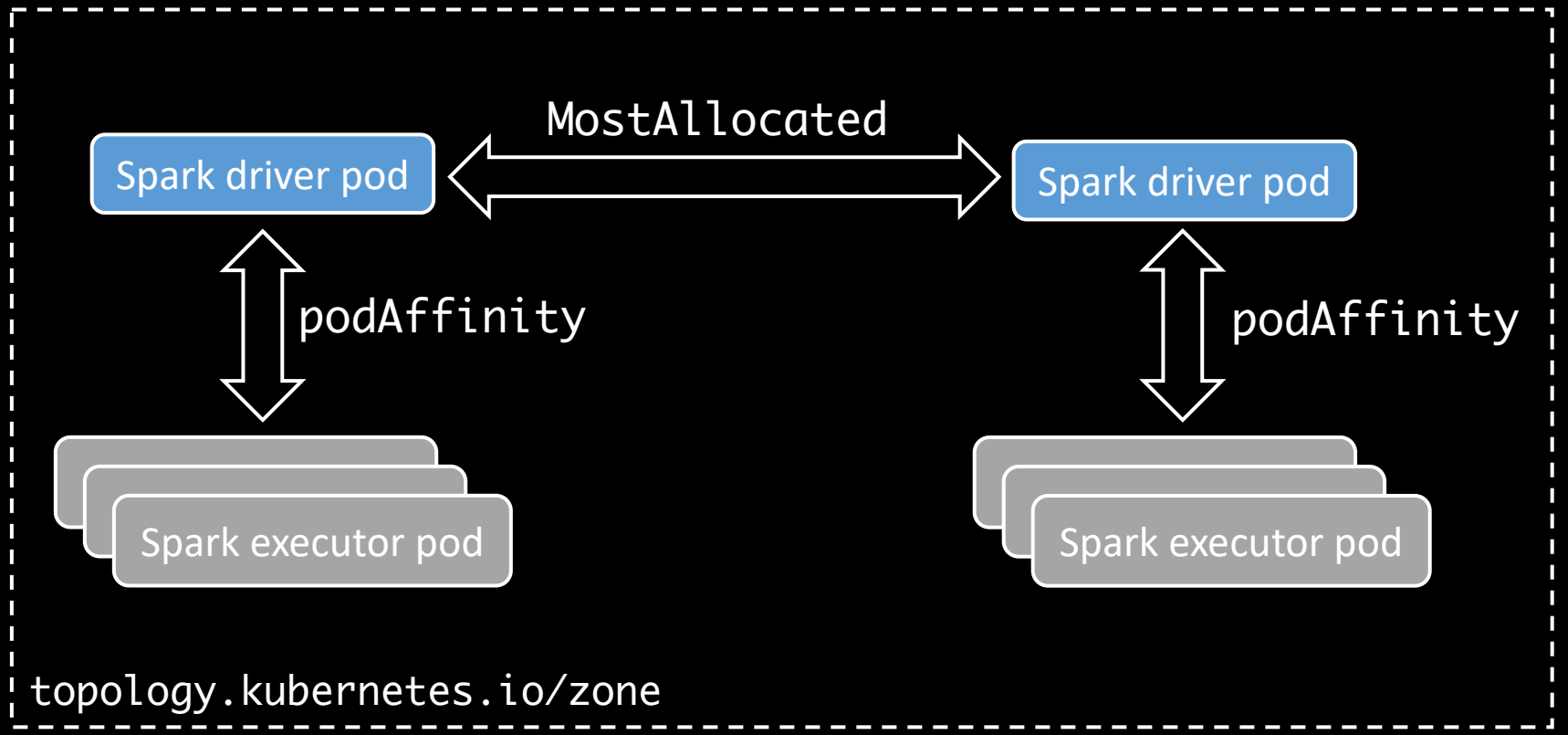
`MostAllocated`



Custom `kube-scheduler`: Taste of Victory



podAffinity + kube-scheduler: a Problem



Very Default, Very Demure

cluster-wide default

Suggestion, **allows scheduling**

kube-scheduler

```
defaultConstraints:
- maxSkew: 5
    topologyKey: "topology.kubernetes.io/zone"
  whenUnsatisfiable: ScheduleAnyway
```

Enforced Pod Spread

Match number of pods across zones

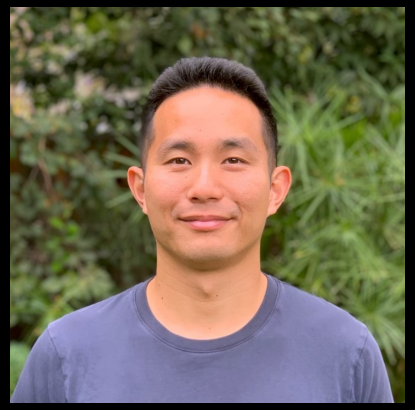
Enforced, **blocks scheduling**

Spread drivers, executors follow

Spark driver pod

```
topologySpreadConstraints:
- maxSkew: 1
  minDomains: 3
  topologyKey: topology.kubernetes.io/zone
  whenUnsatisfiable: DoNotSchedule
  labelSelector:
    matchLabels:
      spark-pod-kind: driver
  matchLabelKeys:
    - tier
```

No Tricks, Just Teamwork

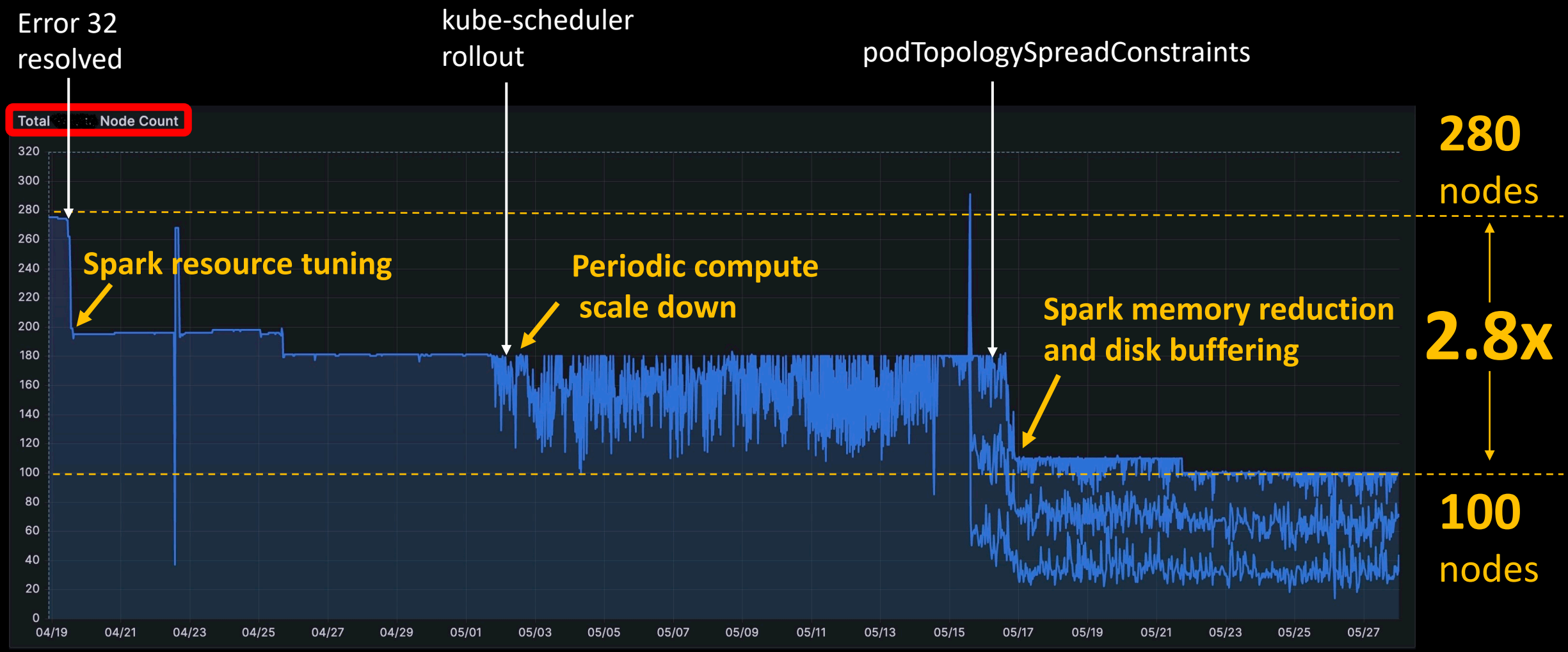


Neil
Chao

Chonghan
Chen

Craig
Skinfill

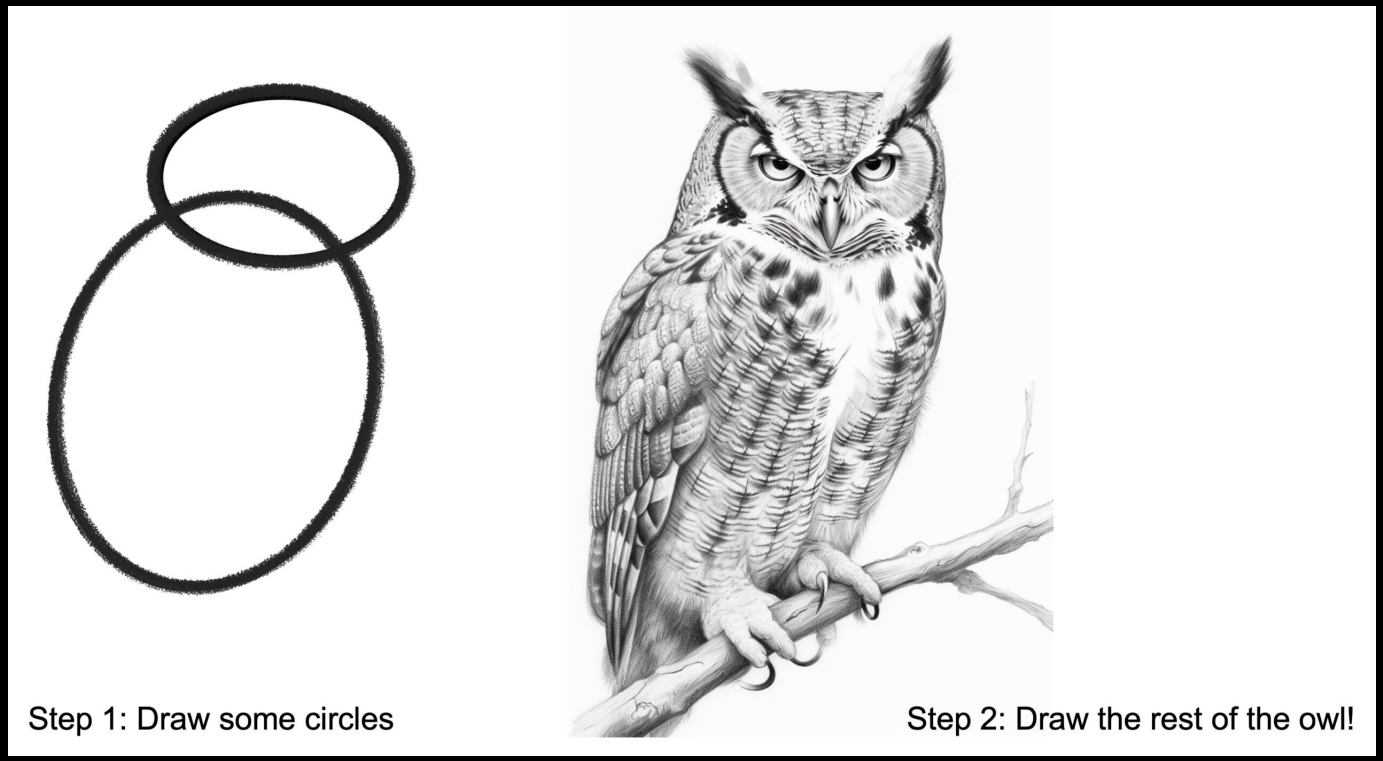
No Tricks, Just Deliberate Optimization



The Secret of Scale

- Find the right flow control settings
- Find chaos causing clients
- Optimize the workload

Optimize the workload



Guides
https://aws.github.io/aws-eks-best-practices/scalability/docs/node_efficiency/
https://aws.github.io/aws-eks-best-practices/scalability/docs/scaling_theory/
Talk
https://www.youtube.com/watch?v=NqtfDy_KAqg

Want More?