KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

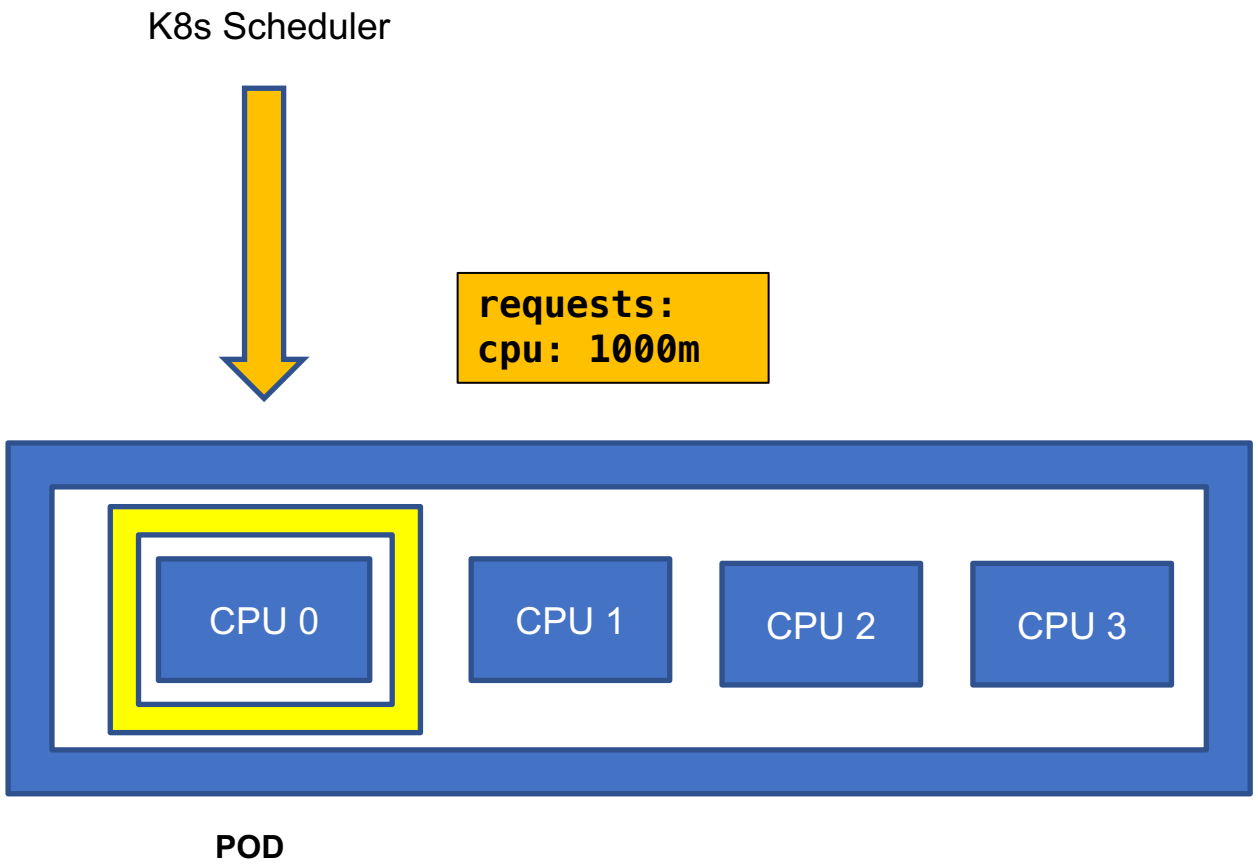
DETROIT 2022

73,000 Pods a Day Misadventures in Multi-Tenant

Wil Reed, Acquia
Shane Corbett, Amazon Web Services

Kubernetes vs. Linux

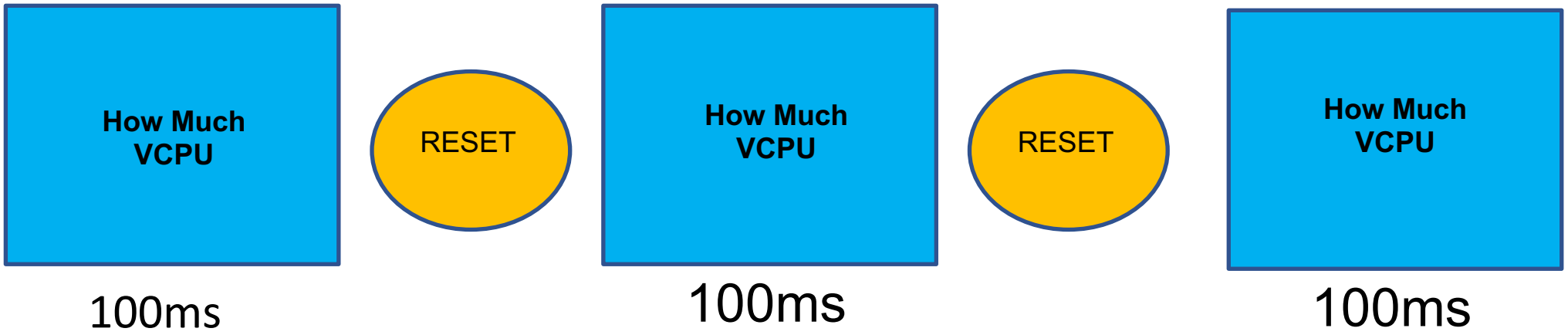
Cores




```
Capacity:
  cpu:              4
  pods:              110
Allocatable:
  cpu:              3000m
  pods:              110
```

Thinking in Time
Not Cores

Limits



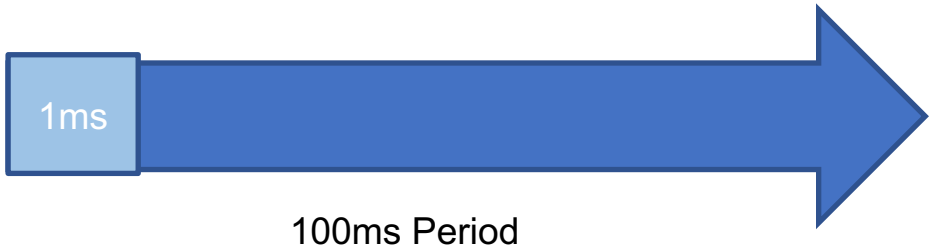
Period – 100ms

Conversion

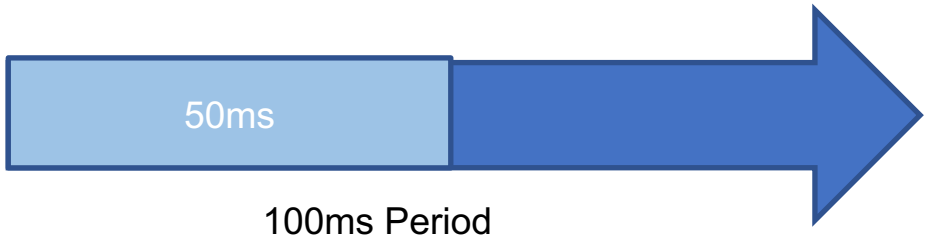
1 Core = 1000m

```
Resources:
  limits:
    cpu: 10m
```

10m/1000m = 1%



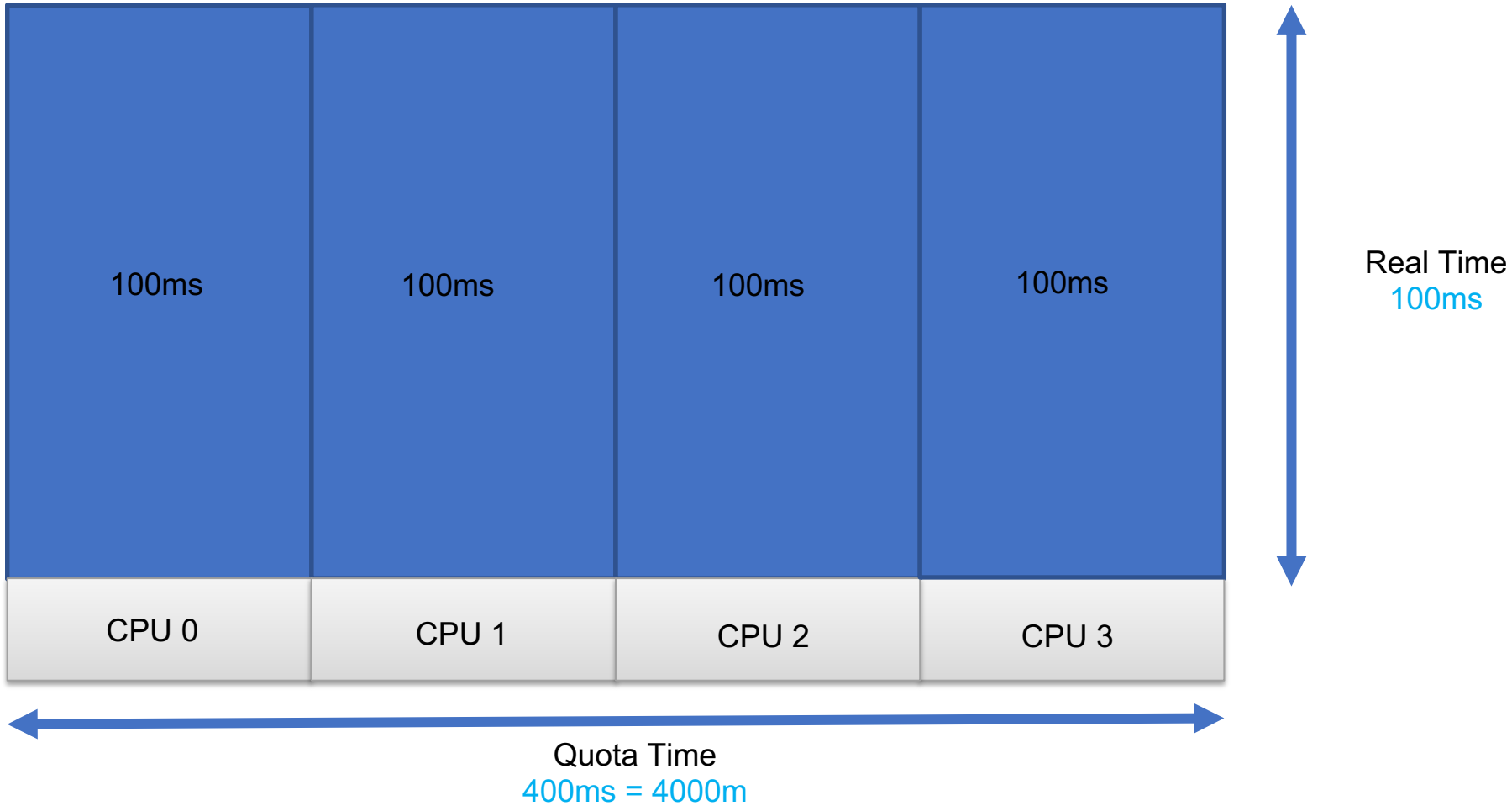
500m/1000m = 50%



Time

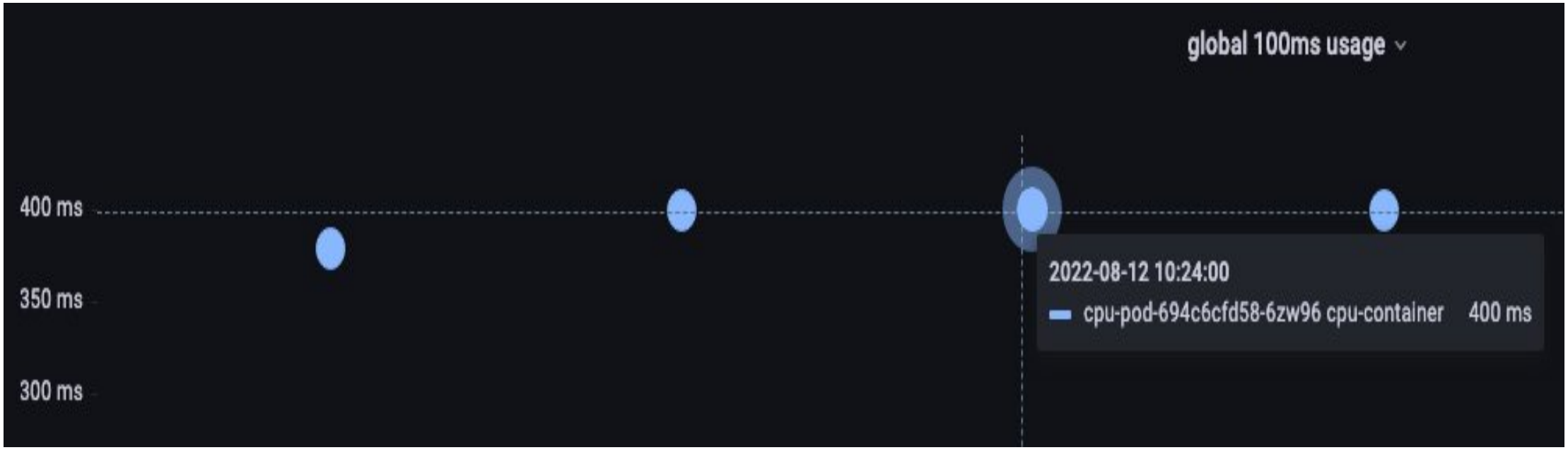
Four Thread app

4 Core Node



Thinking in Time

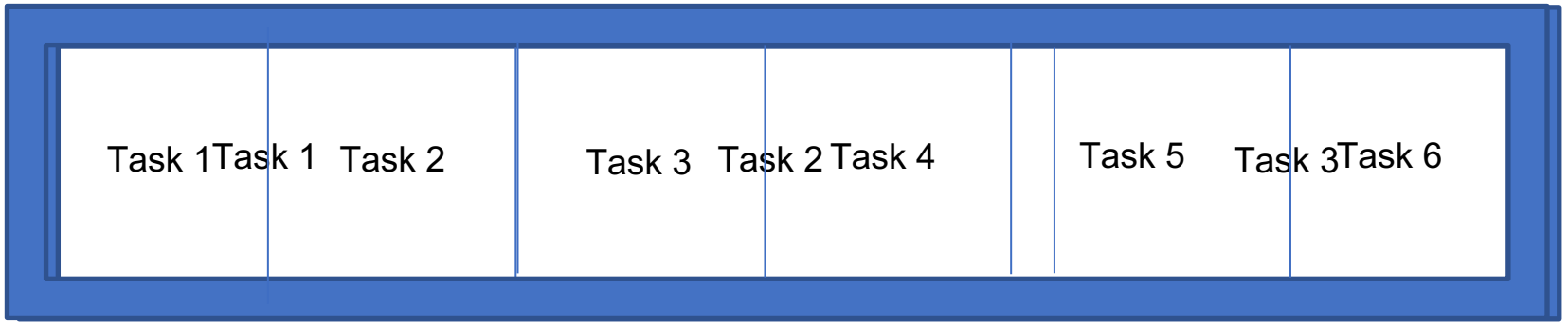
- **container_cpu_usage_seconds_total**



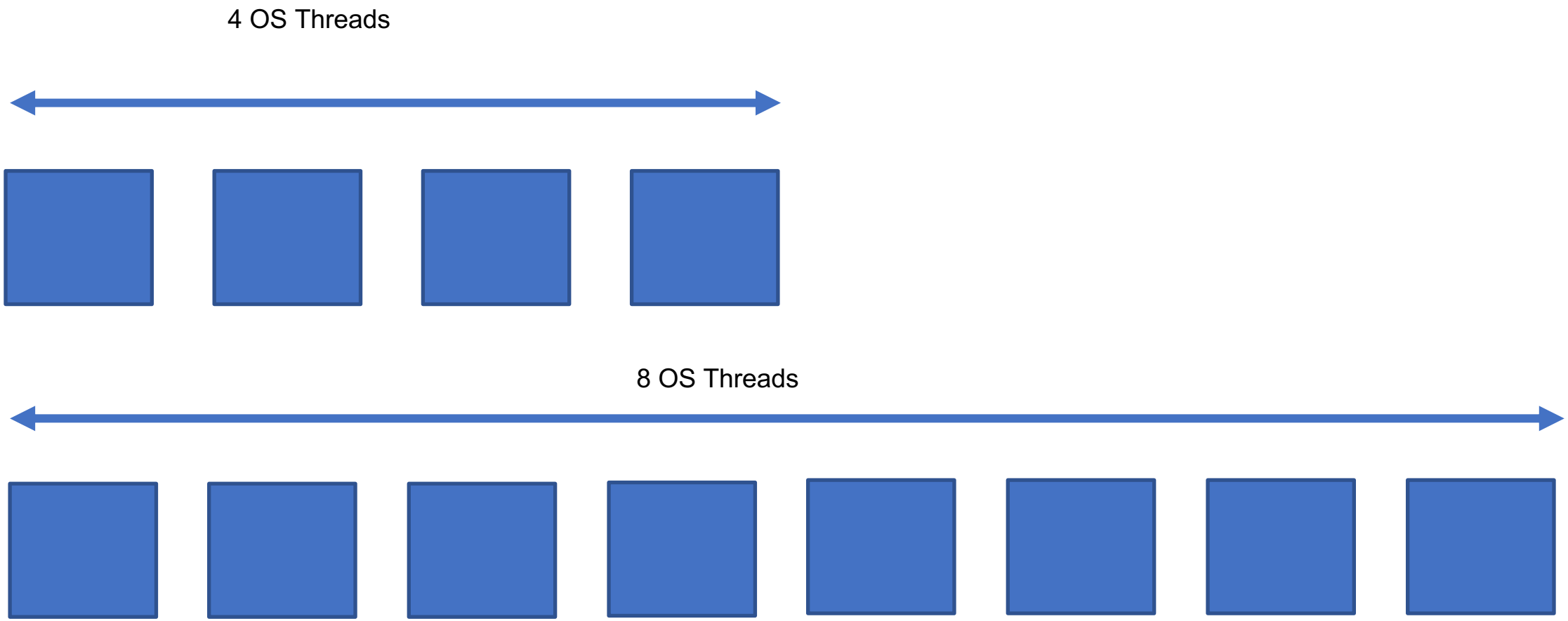
** Per Period*

Threads

Container 1



Swapping Cores



Throttles

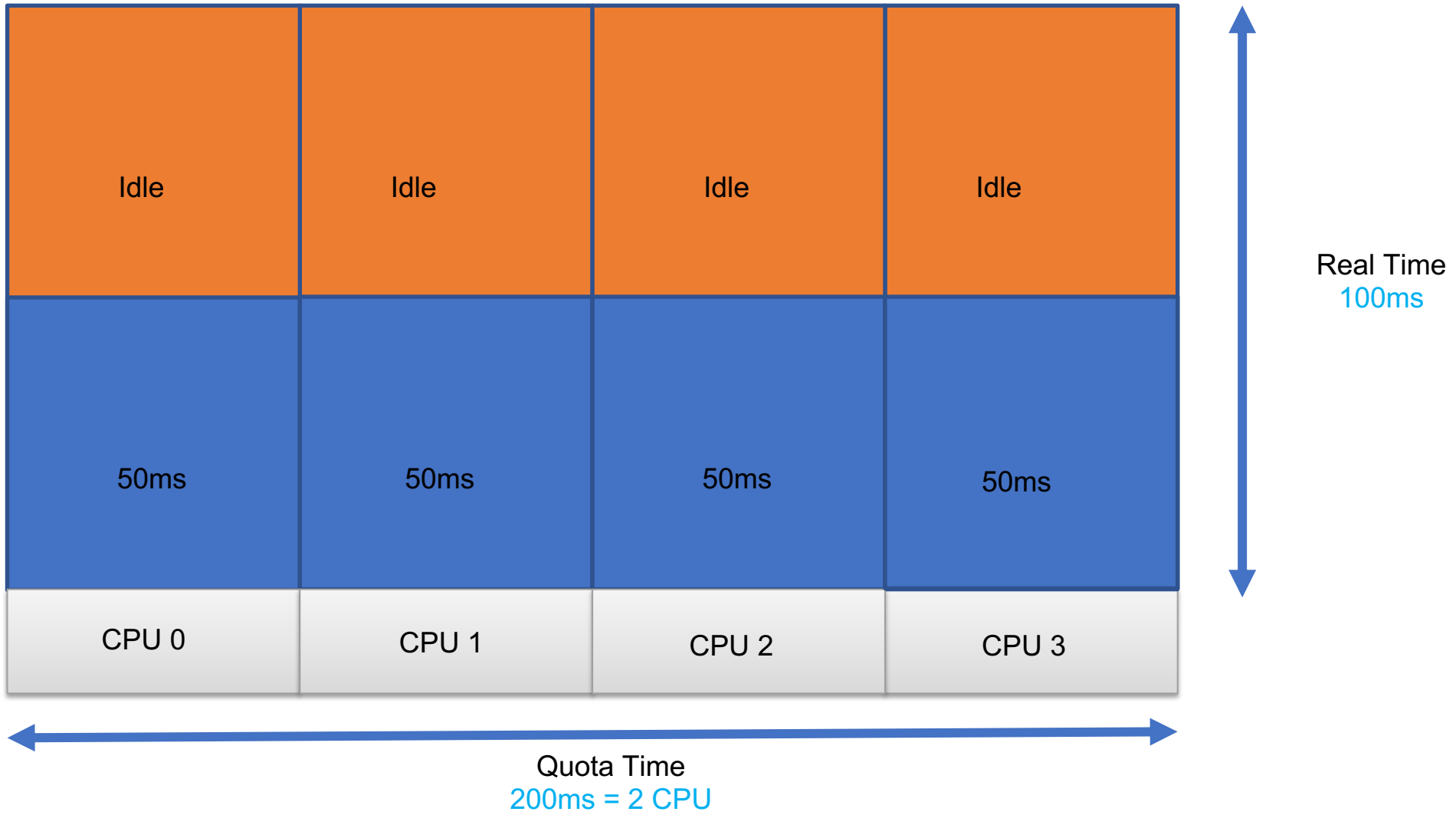
Throttles - Bad Pods Limits

pod full-test-27434725--1-b6qn7	99.8%
pod full-test-27434725--1-w55zh	99.8%
pod full-test-27434725--1-7zsdd	99.7%

Query 1 Transform 0

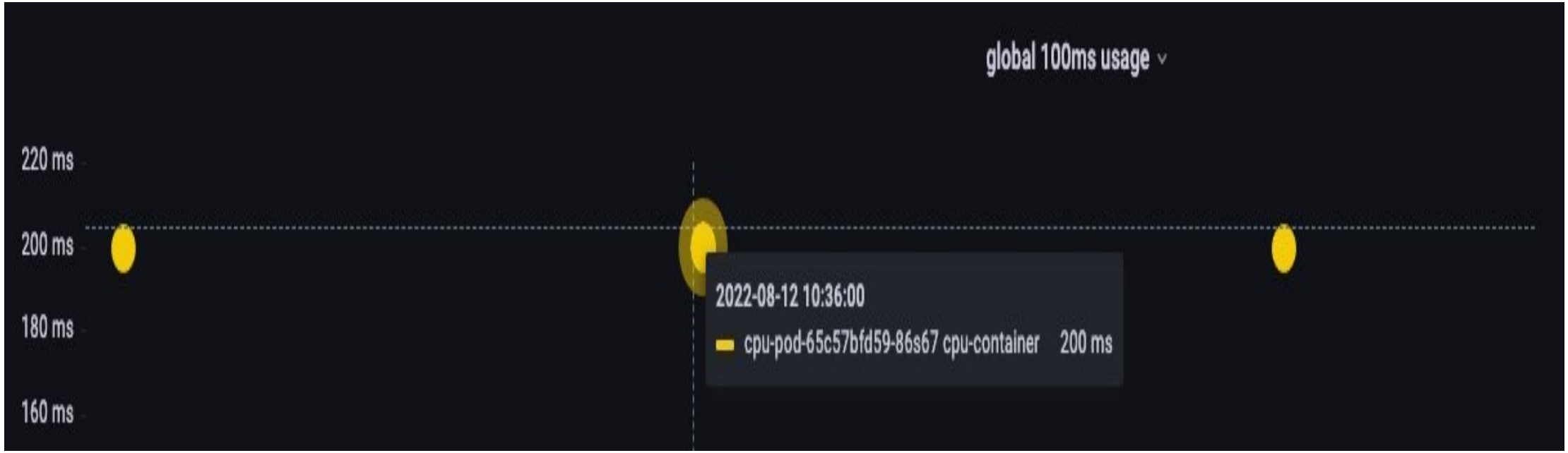
Double It!

```
Resources:
  limits:
    cpu: 2000m

    or

    200ms
```

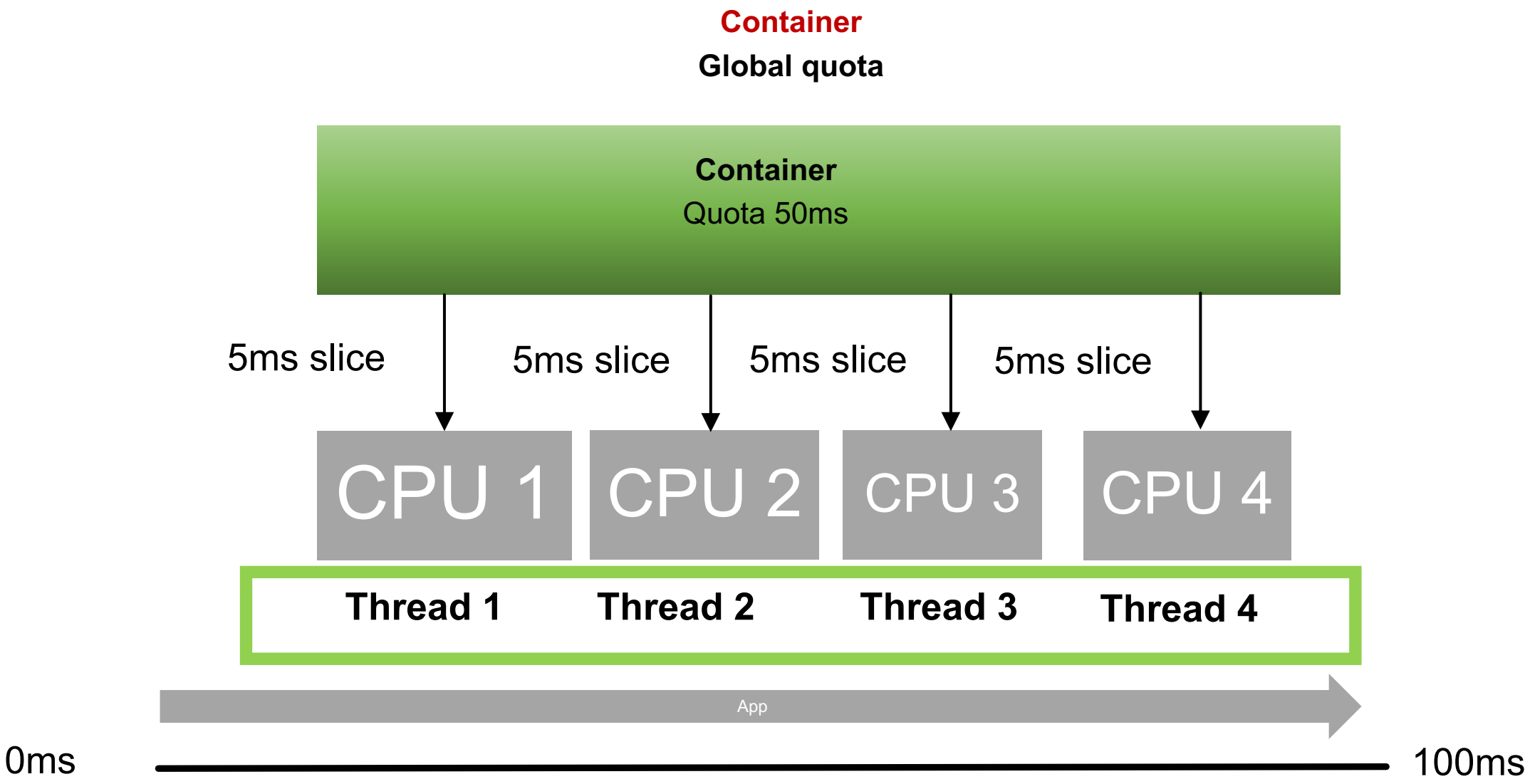


Thinking in Time

- **container_cpu_throttled_seconds_total**



** CPU Usage Per Period*


Container
Global quota
Container
Quota 50ms
5ms slice
5ms slice
5ms slice
5ms slice
CPU 1
CPU 2
CPU 3
CPU 4
Thread 1
Thread 2
Thread 3
Thread 4
App
0ms
100ms

aws

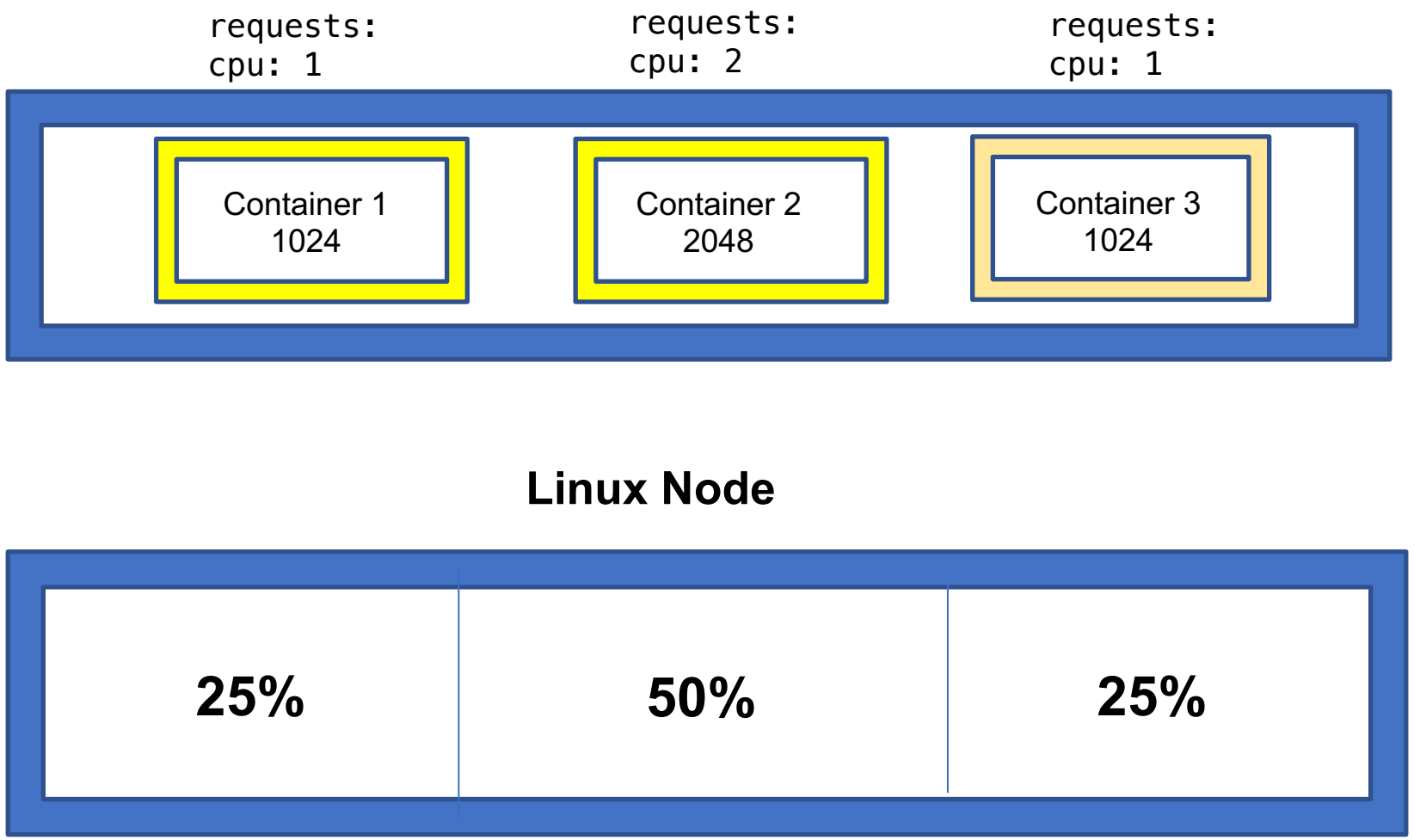
CONTAINERS

Using Prometheus to Avoid Disasters with Kubernetes CPU Limits

Read the blog post ›

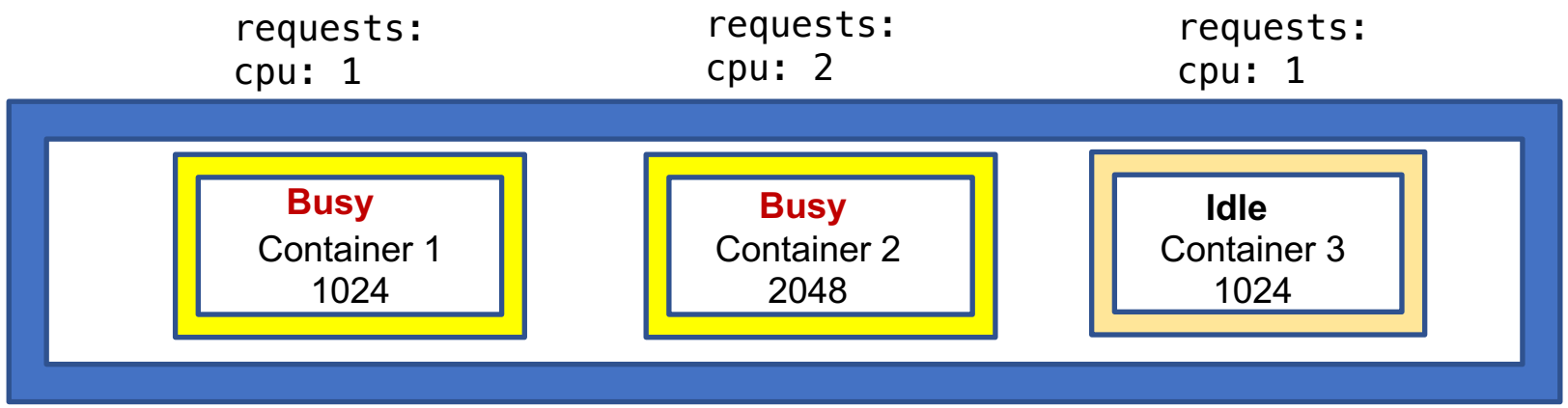
Static vs. Dynamic

Mixing Models



Dynamic Shares

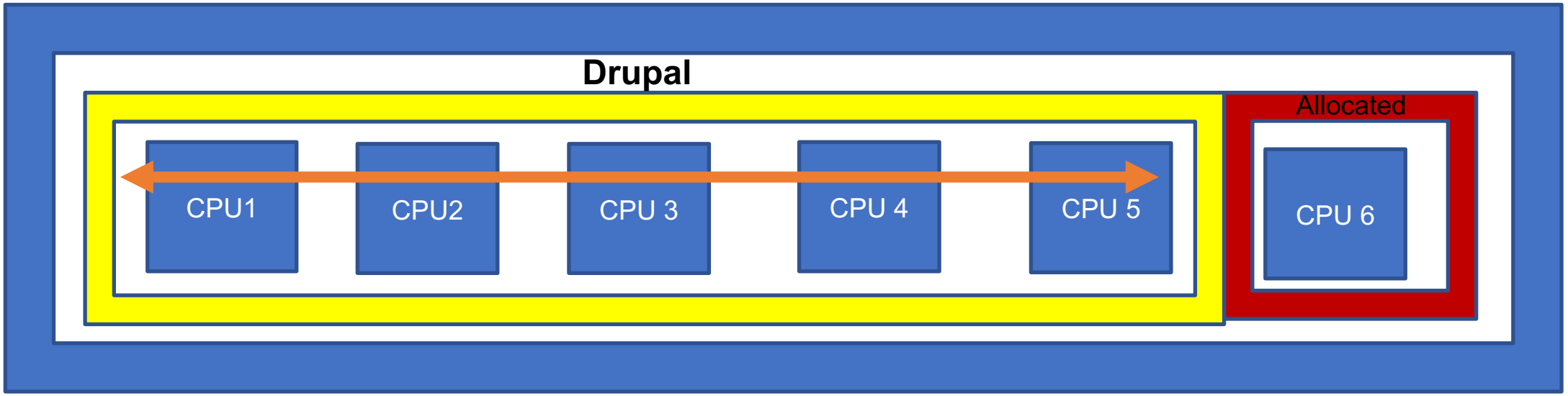
CPU * Container Shares / Busy shares



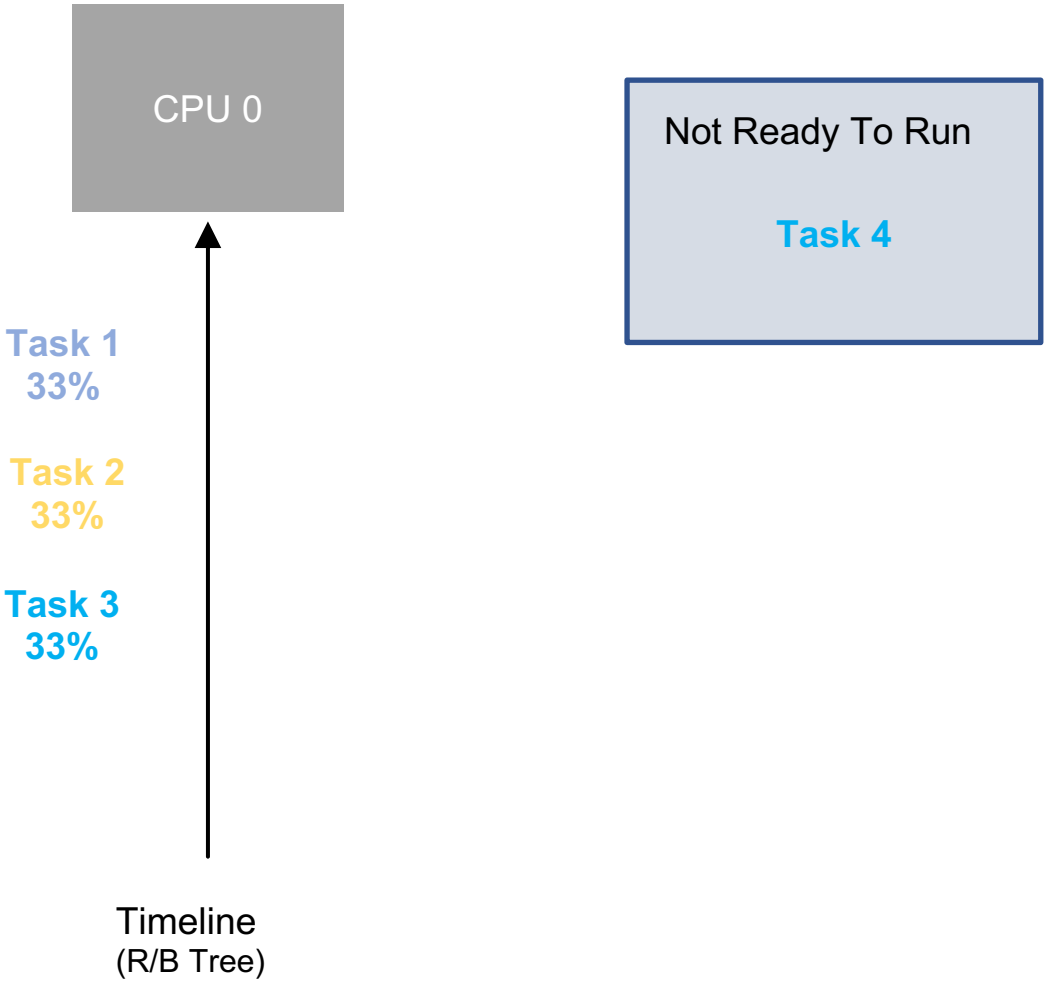
Linux Node

Only If It’s Busy

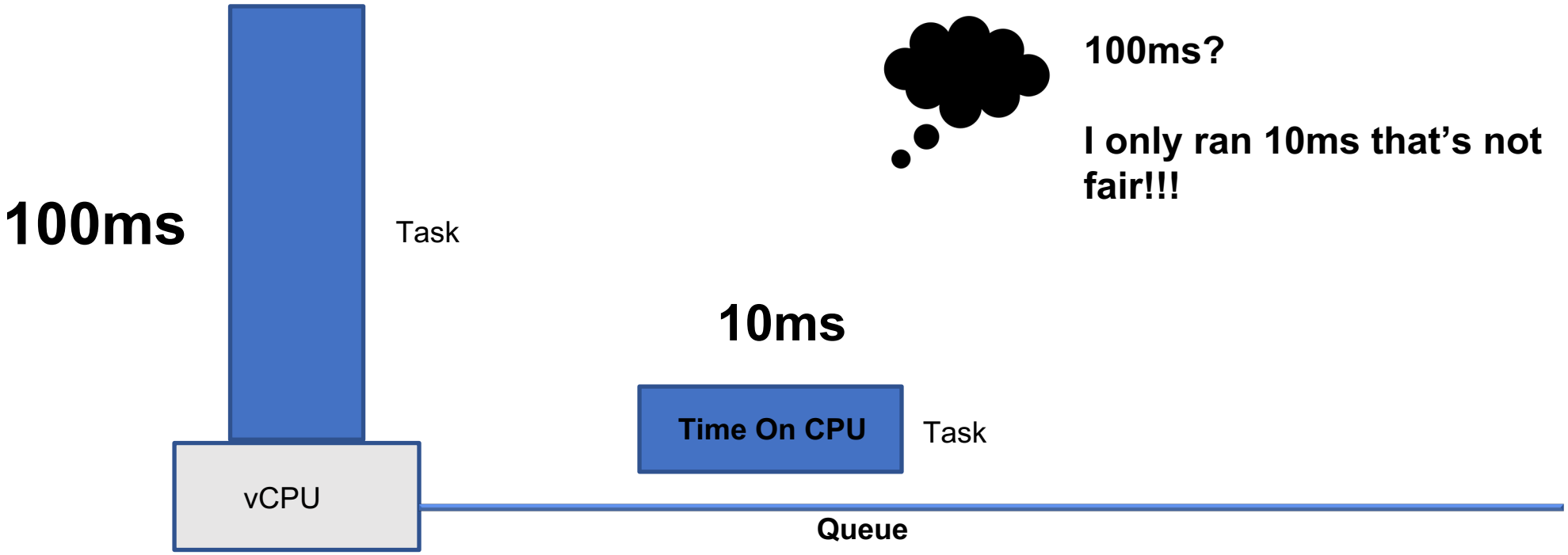
```
requests:
cpu: 1000m
```



Why?

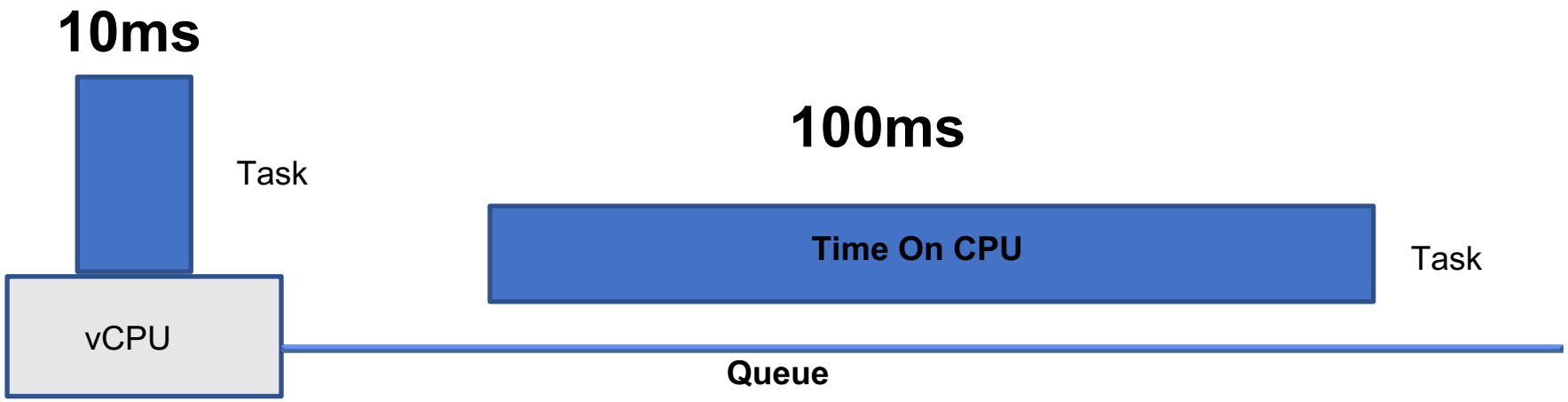

CPU 0
Not Ready To Run
Task 4
Task 1
33%
Task 2
33%
Task 3
33%
Timeline
(R/B Tree)

Linux is Fair



Linux is Fair

Lowest Time on CPU goes first!



Time Not Cores

TASK RUNTIME

100ms

```
requests:
cpu: 1
```

```
requests:
cpu: 2
```

TASK RUNTIME

TASK RUNTIME

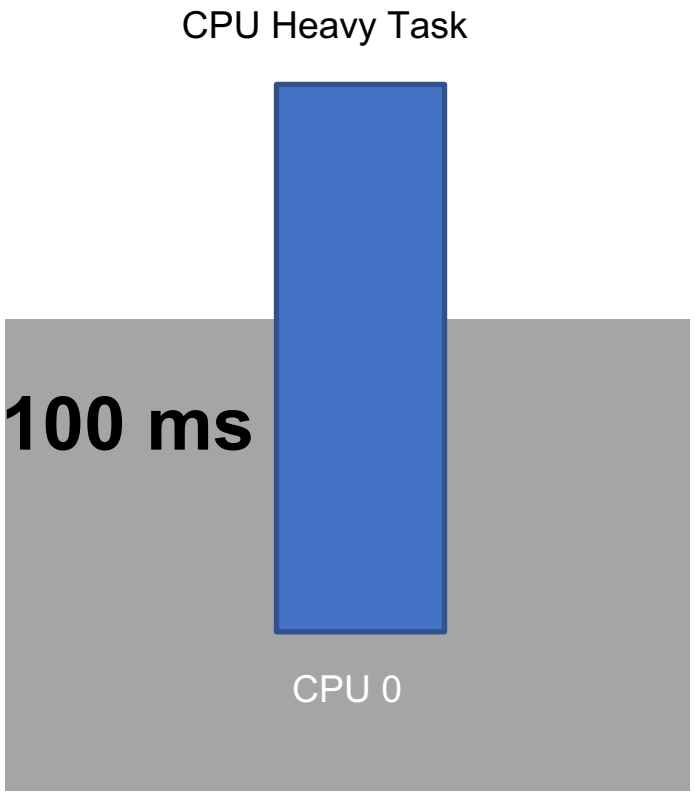
Runtime / 1024 shares

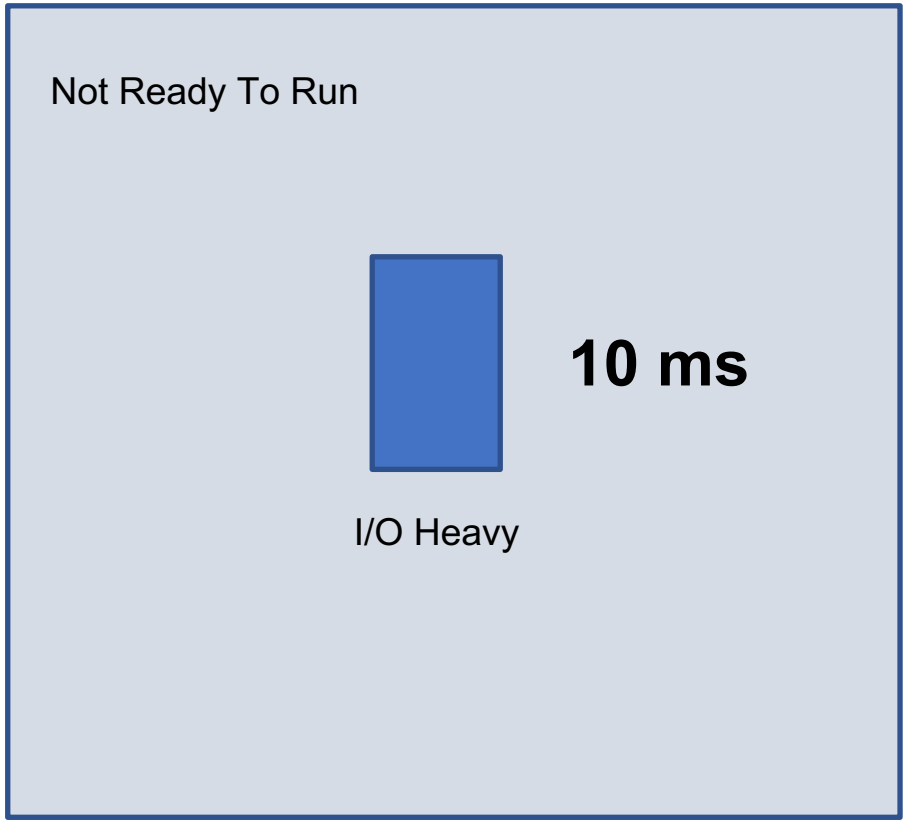
Runtime / 2048 shares

Shorter Runtime

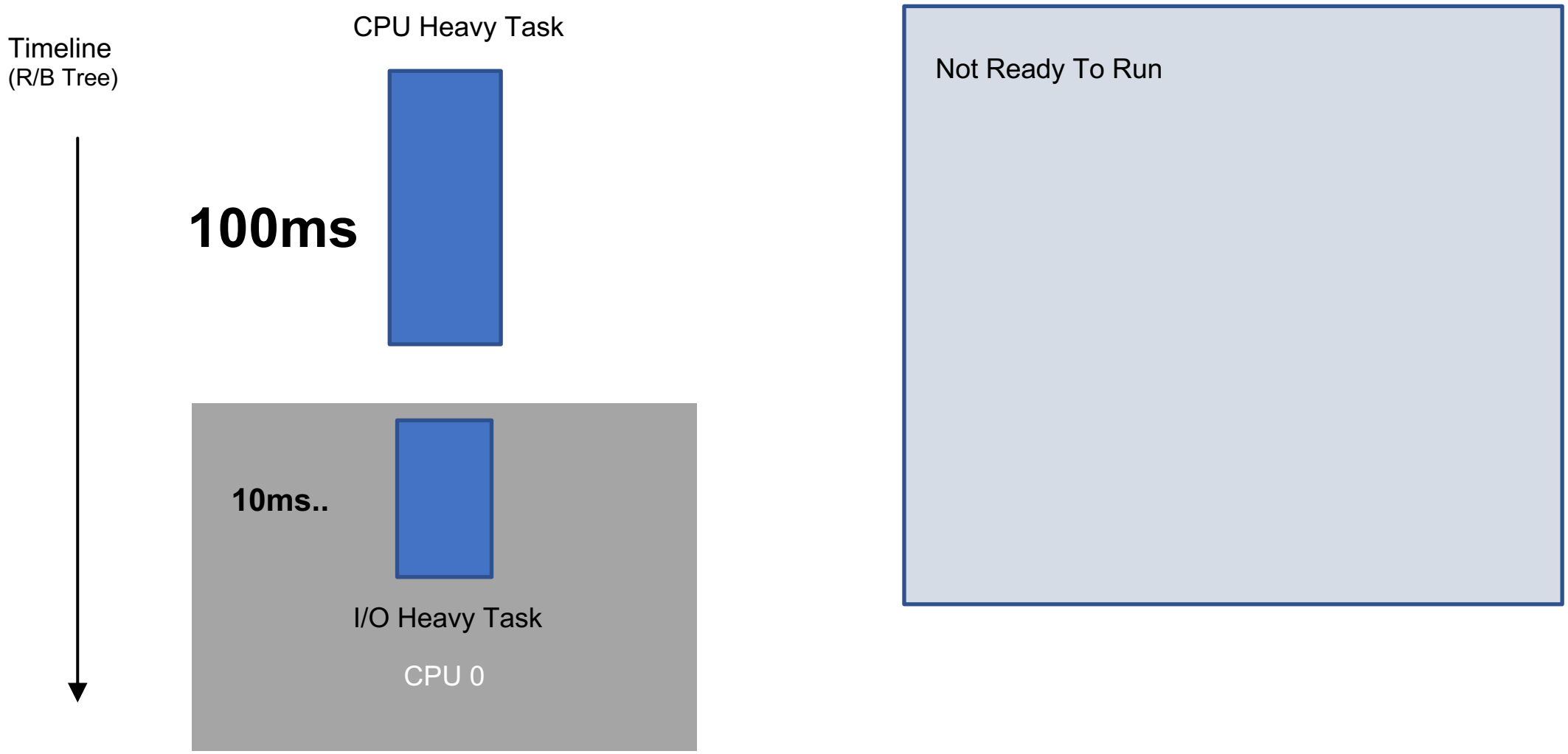
Shortest Runtime

Least Amount of Time

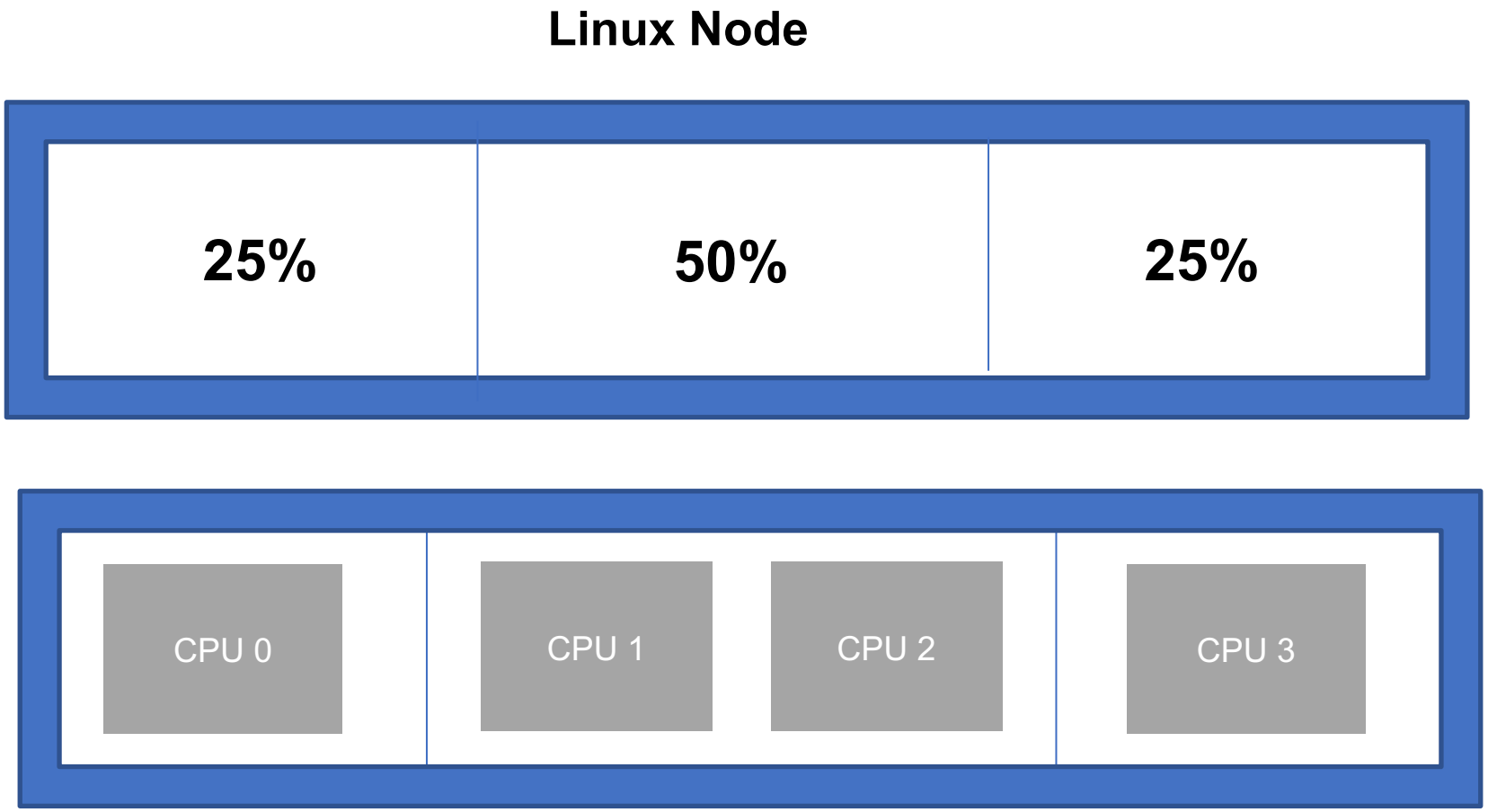




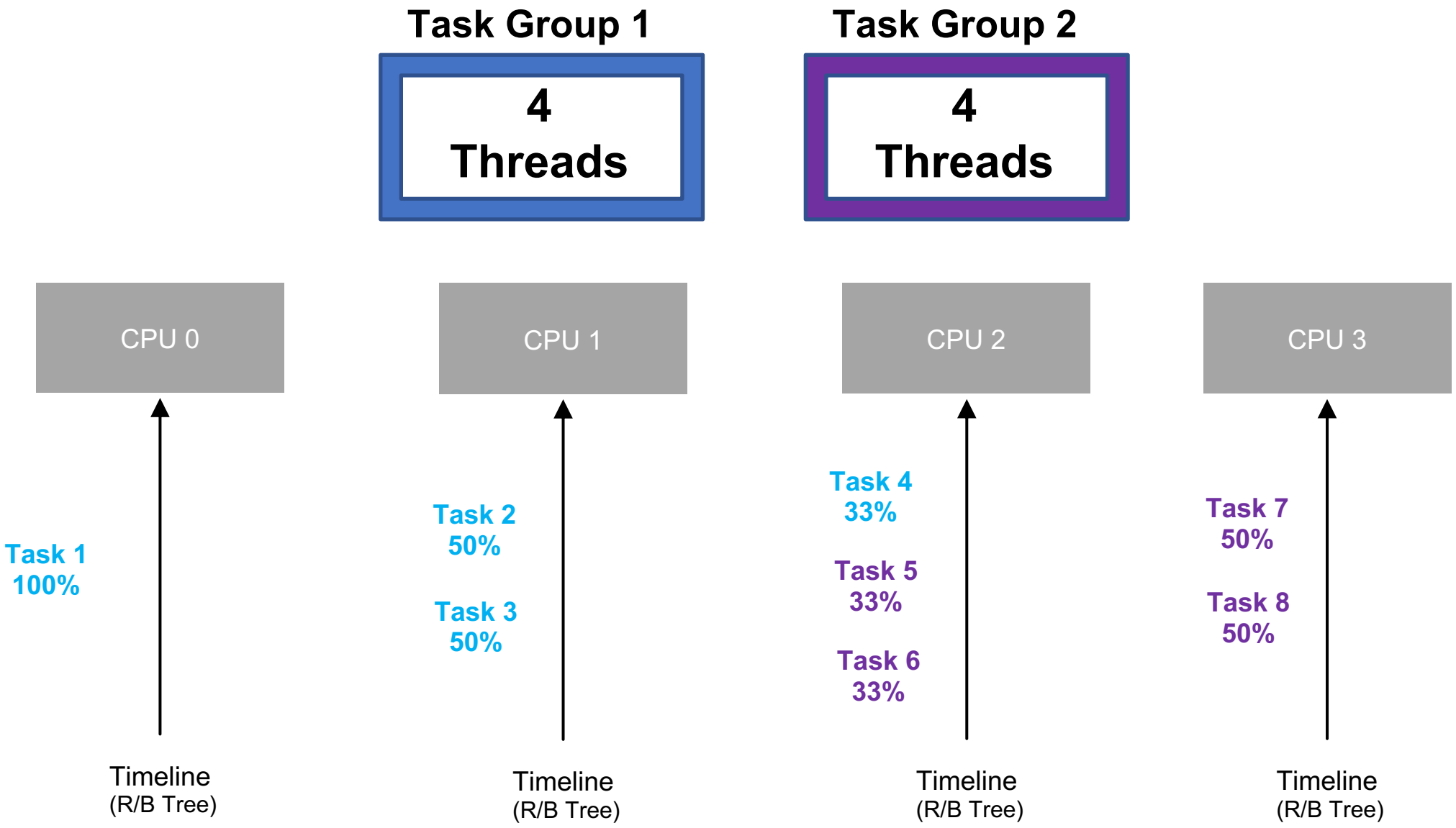
Don’t Forget the I/O Bound



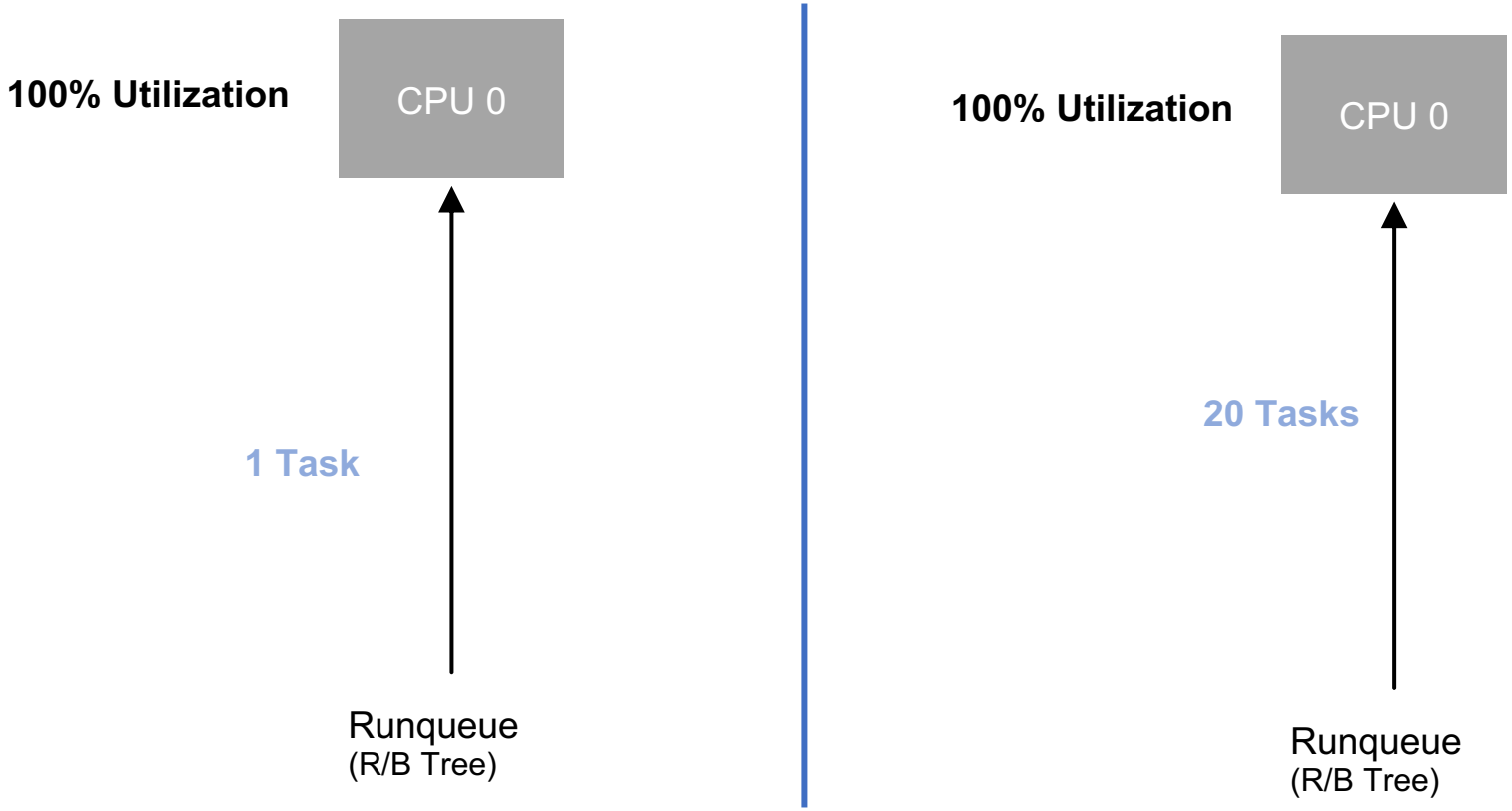
Not Cores



The New Picture

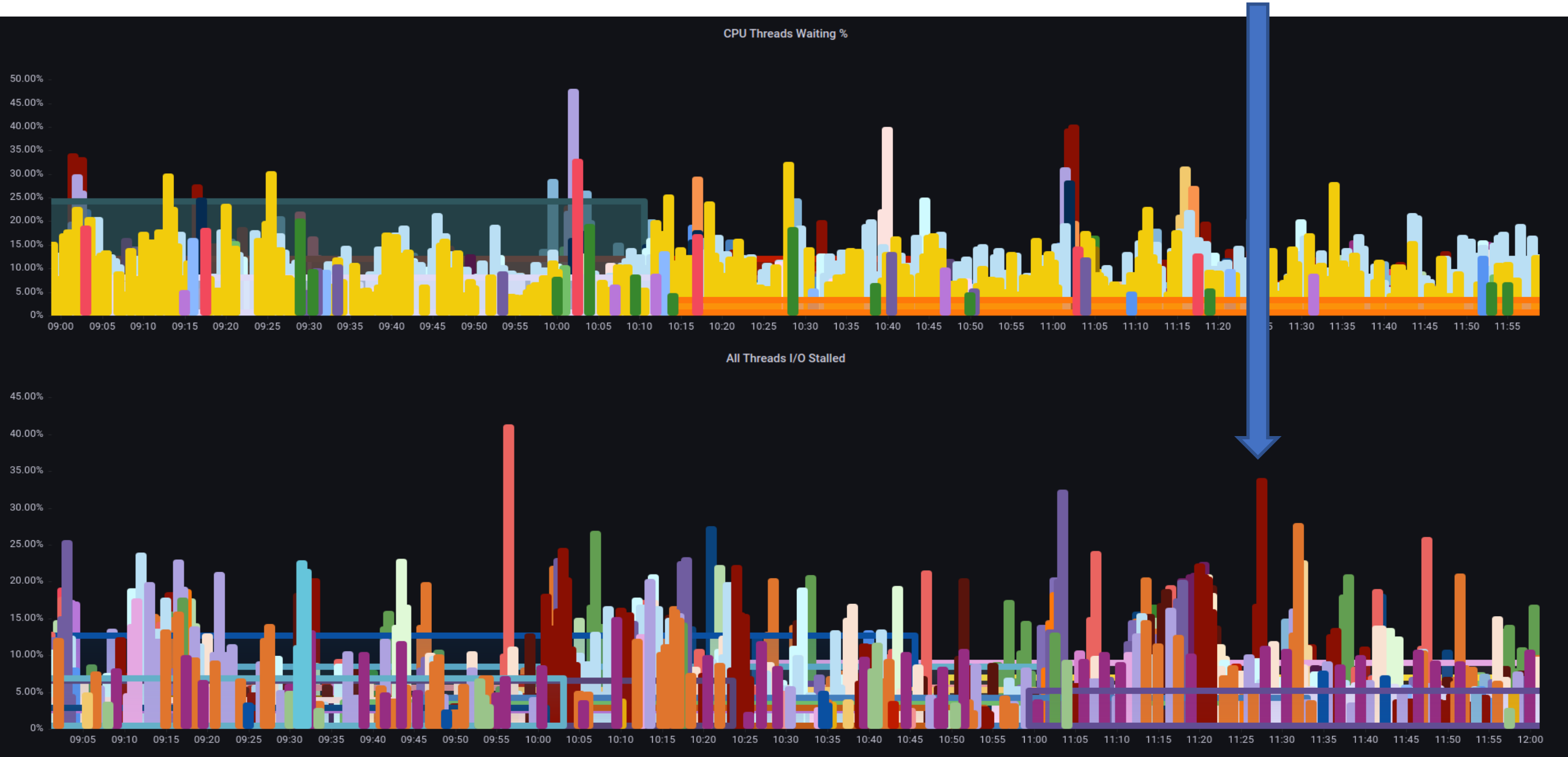


Utilization vs. Saturation

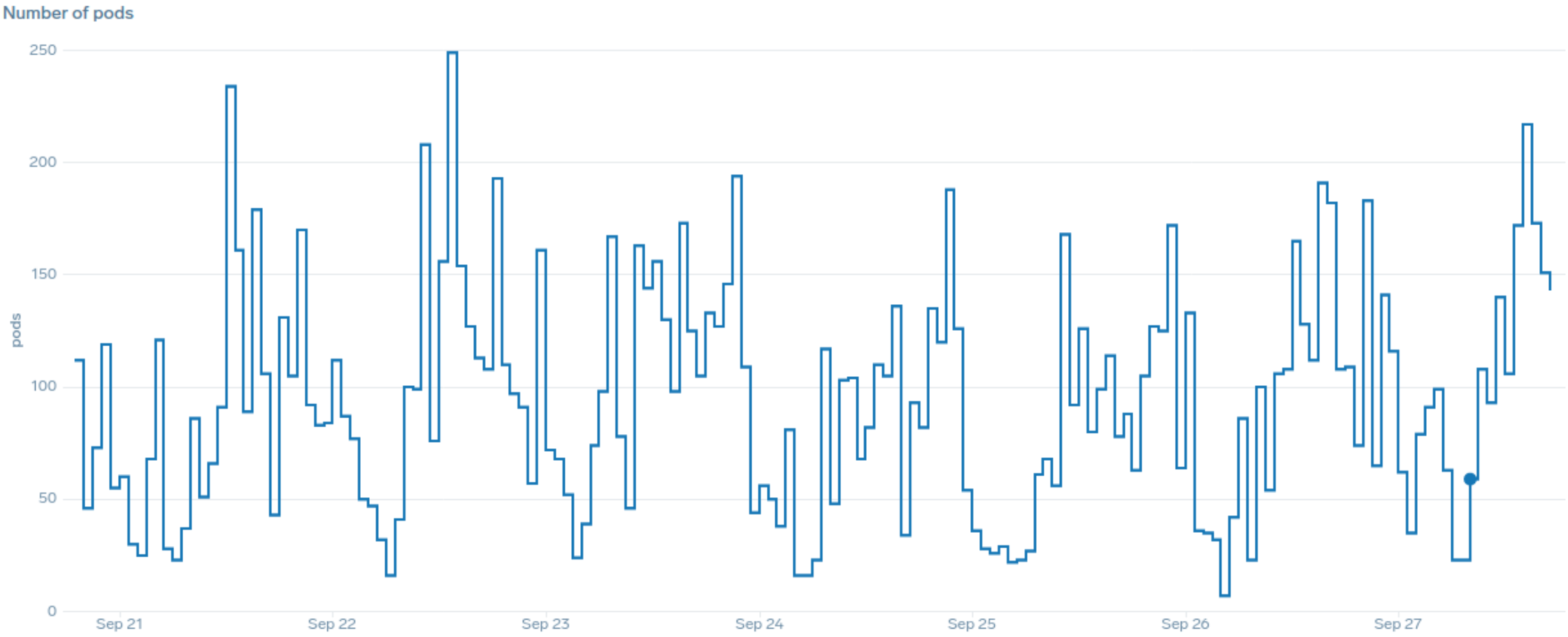


Pressure Stall Information

35% Stalled




IMPORTANT MESSAGE



Unnecessary Churn

Percent of Requests?

% Of Requests

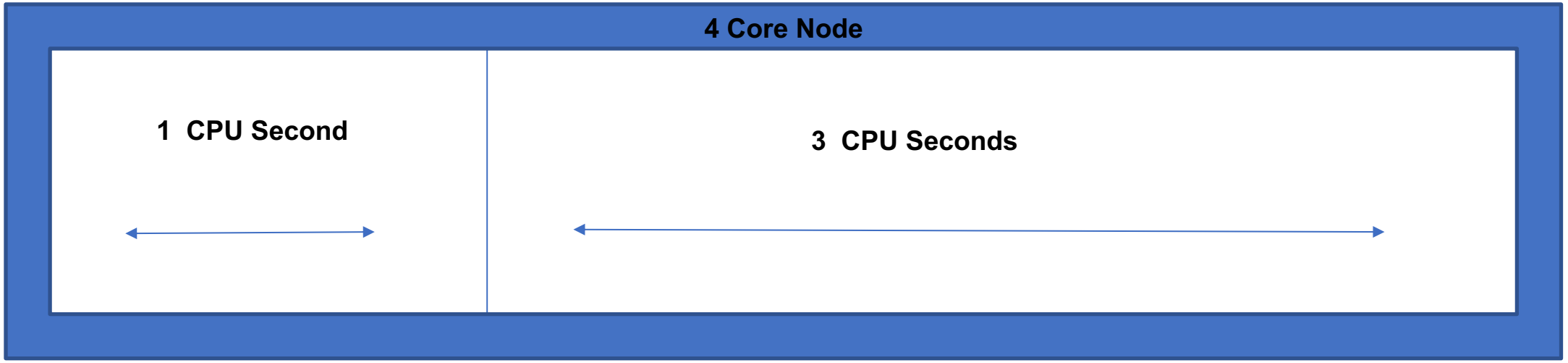
$$\frac{\textit{Actual CPU usage per second}}{\textit{CPU requests}} = \text{\% of Requests}$$

$$\frac{500\text{ms per second}}{1000\text{m}} = 50\text{\% of Requests}$$

100% of Request

```
requests:
cpu: 1
```

```
requests:
cpu: 3
```



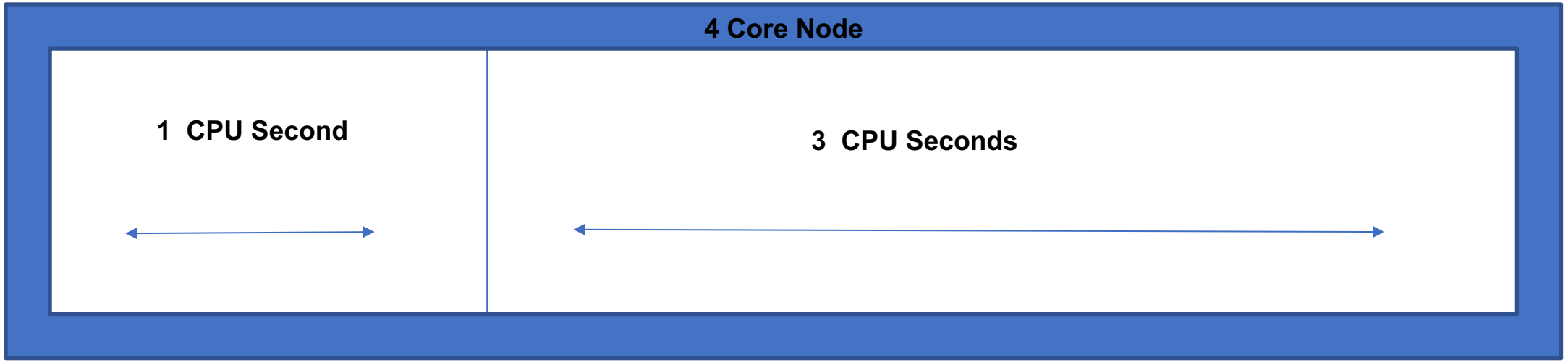
100% of Request

100% of Request

1,000% of Request

```
requests:
cpu: 100m
```

```
requests:
cpu: 300m
```

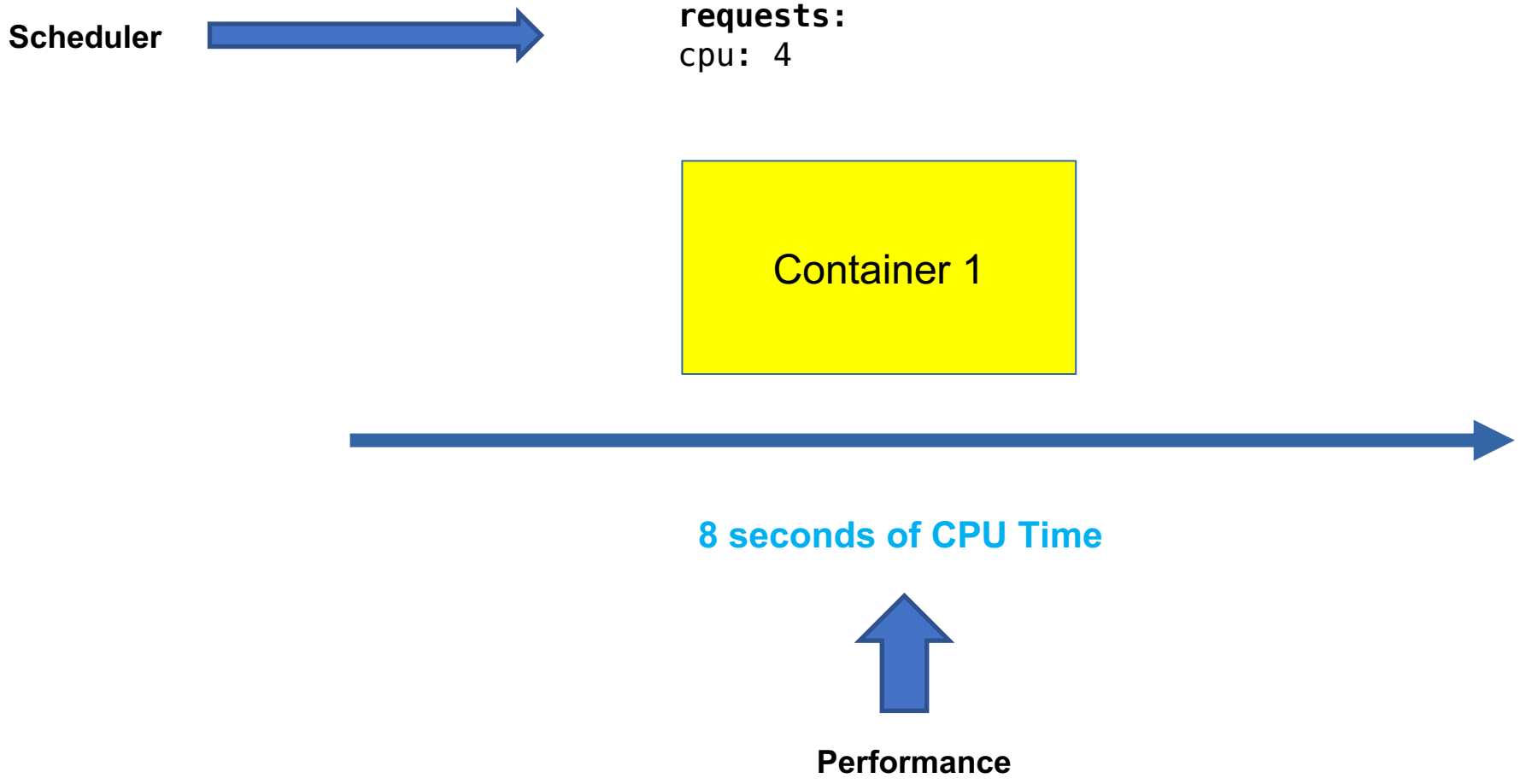


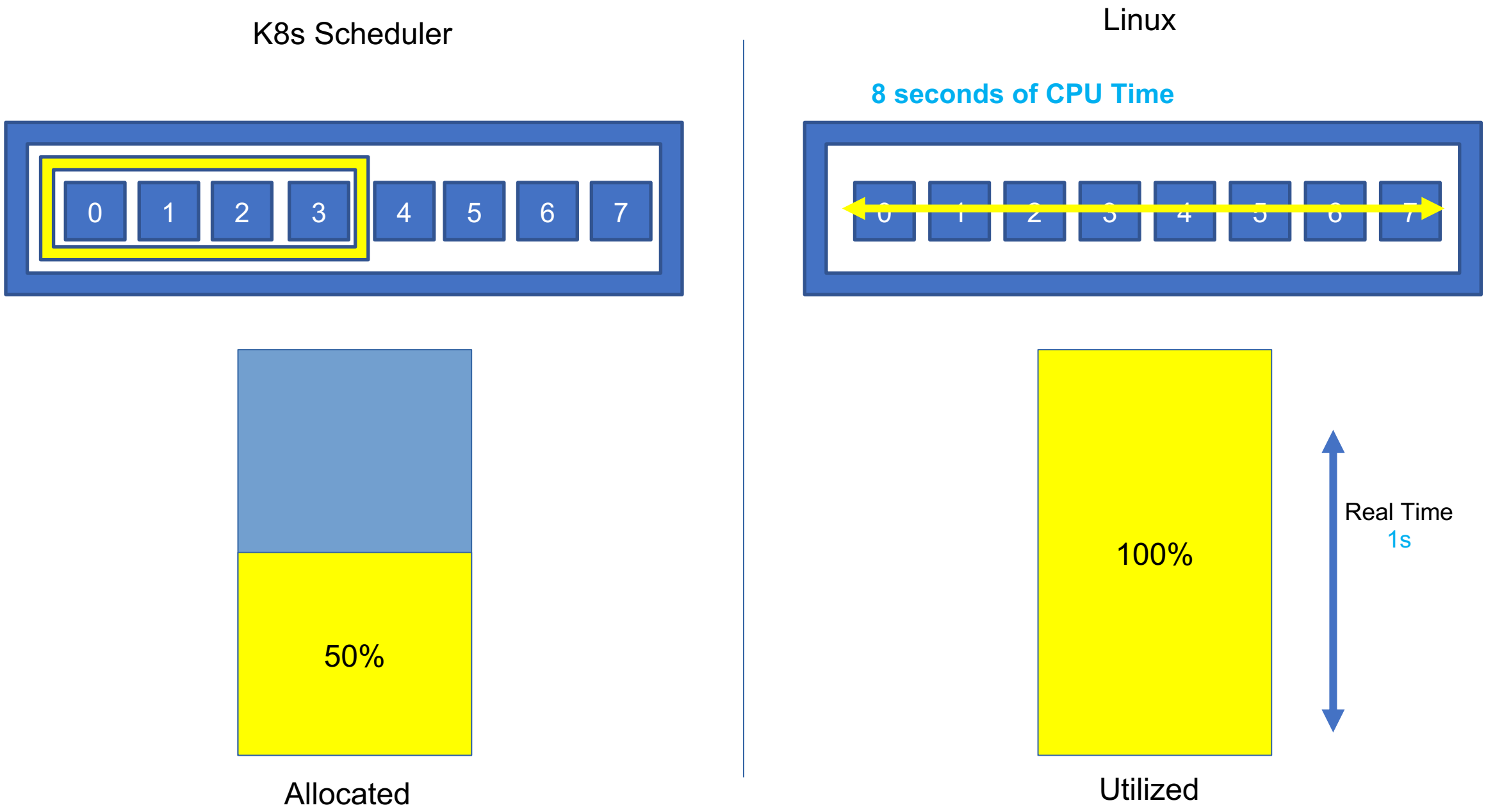
1,000% of Request

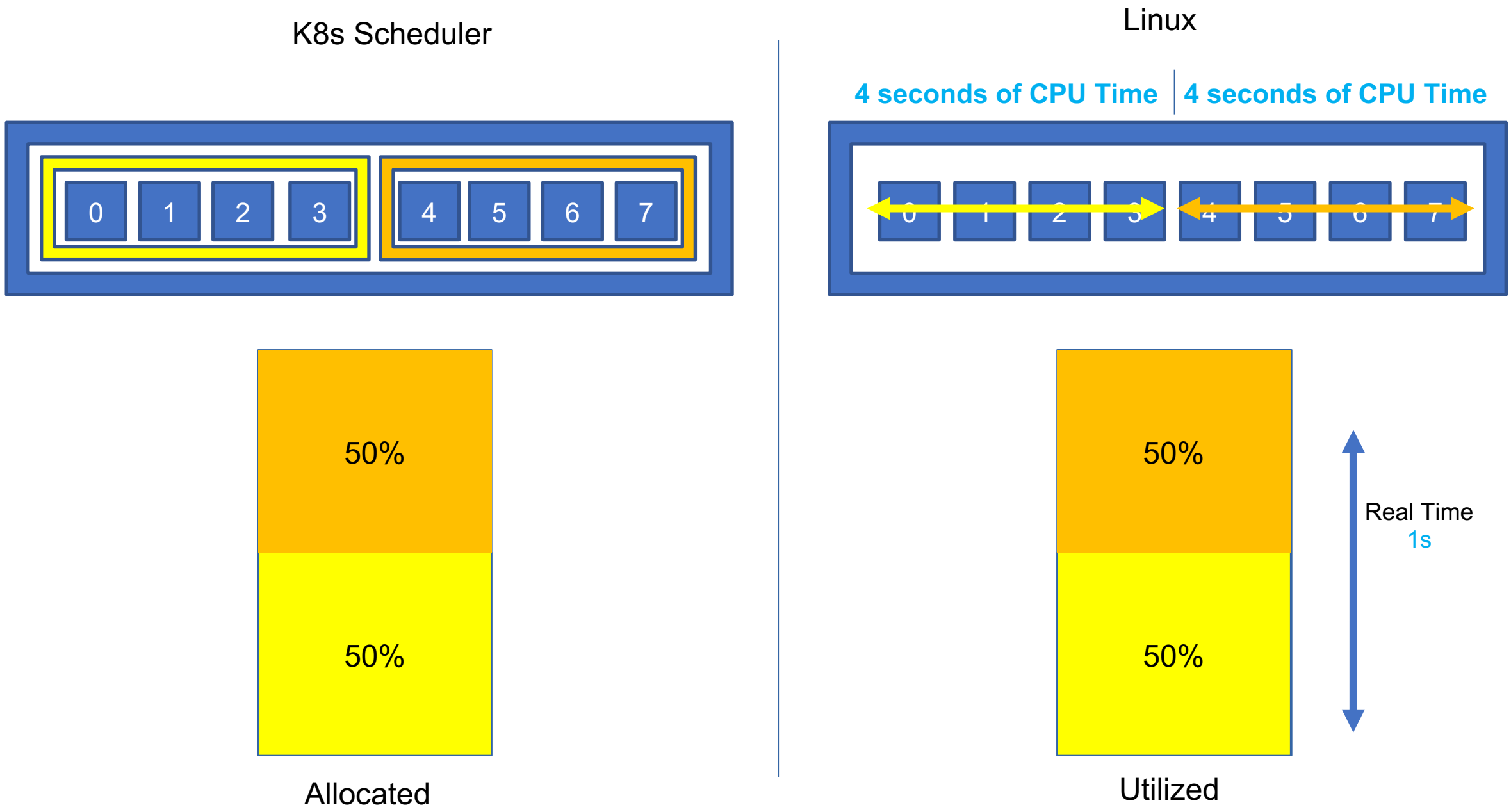
1,000% of Request

The Plot Thickens...

Scheduler vs. Performance




K8s Scheduler
0
1
2
3
4
5
6
7
50%
Allocated
Linux
8 seconds of CPU Time
0
1
2
3
4
5
6
7
100%
Real Time
1s
Utilized


K8s Scheduler
0
1
2
3
4
5
6
7
50%
50%
Allocated
Linux
4 seconds of CPU Time
4 seconds of CPU Time
0
1
2
3
4
5
6
7
50%
50%
Real Time
1s
Utilized

Did we help?

8 Seconds of Work

1 second of Real Time

200% of requests

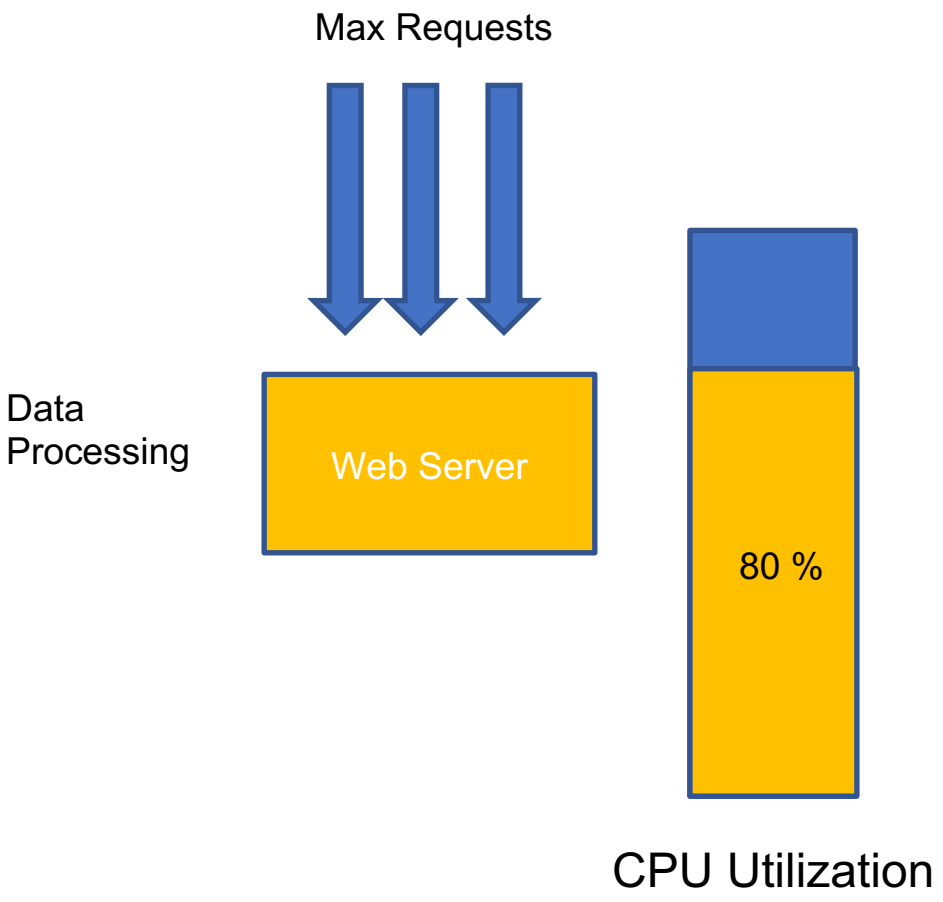
8 Seconds of Work

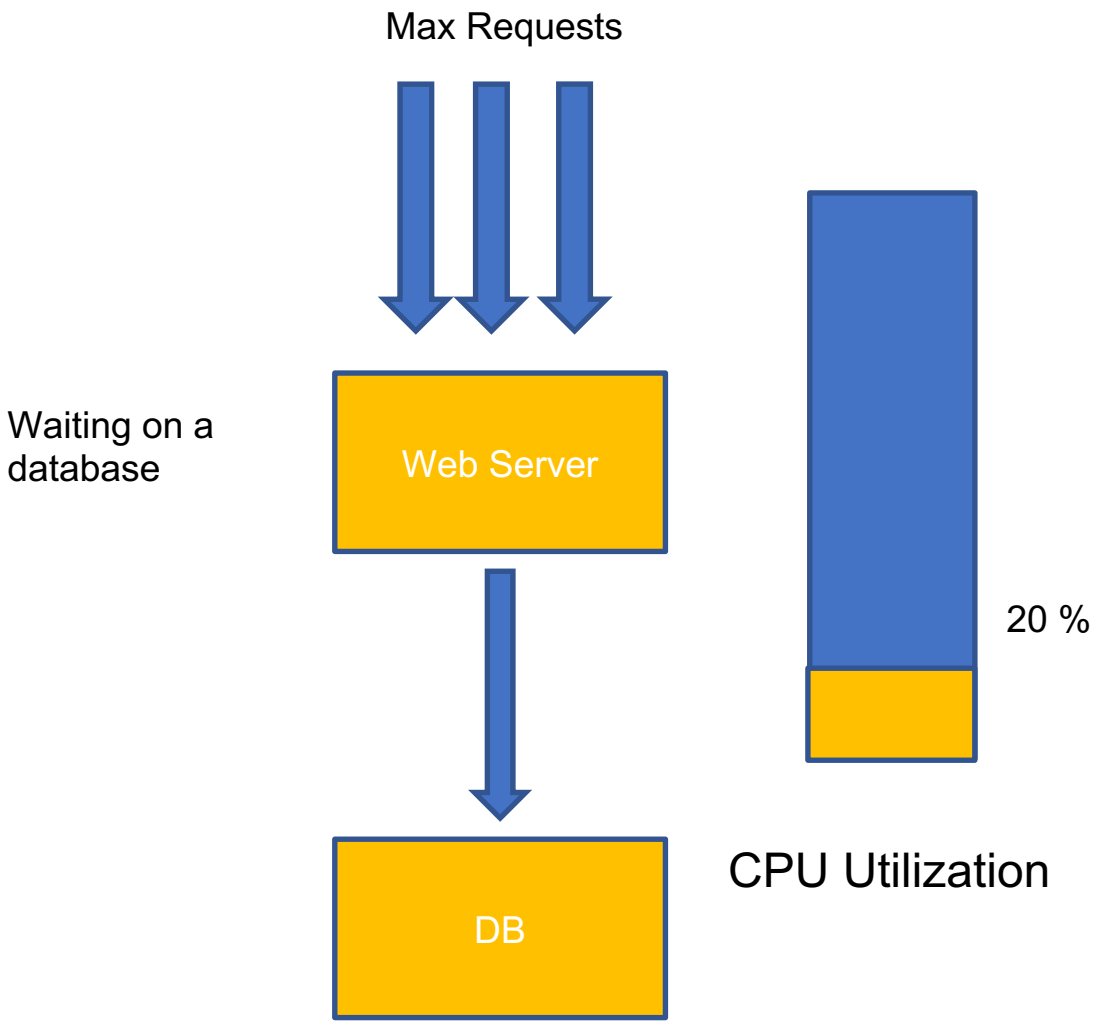
2 seconds of Real Time

100% of requests

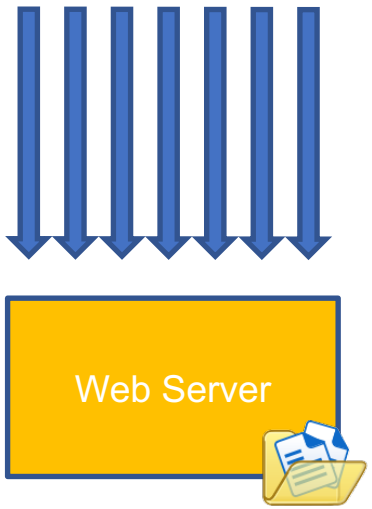
Is CPU The Right Metric At All?

Saturated


Max Requests
Data Processing
Web Server
80 %
CPU Utilization


Max Requests
Waiting on a database
Web Server
DB
20 %
CPU Utilization

Not All Requests Created Equal

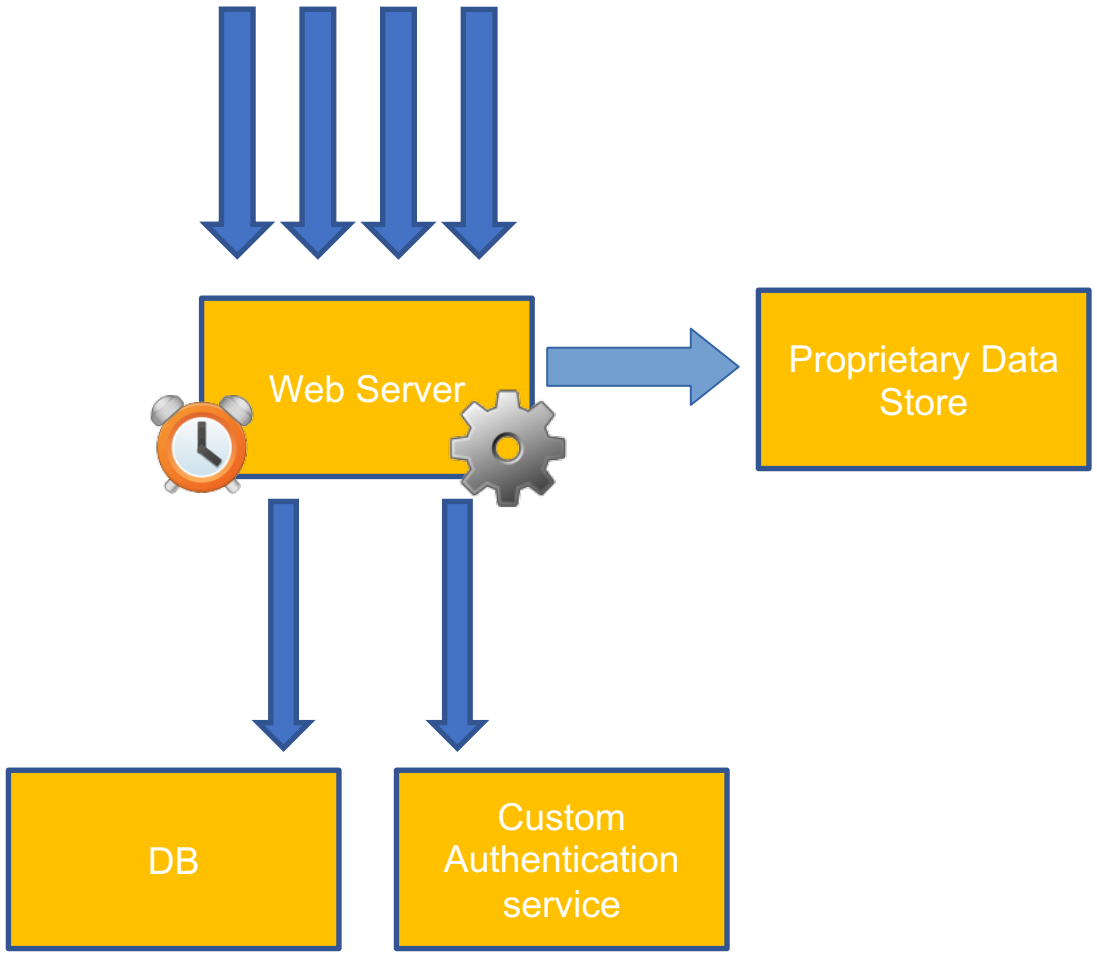


Simple Site

100 RPS

Complex site

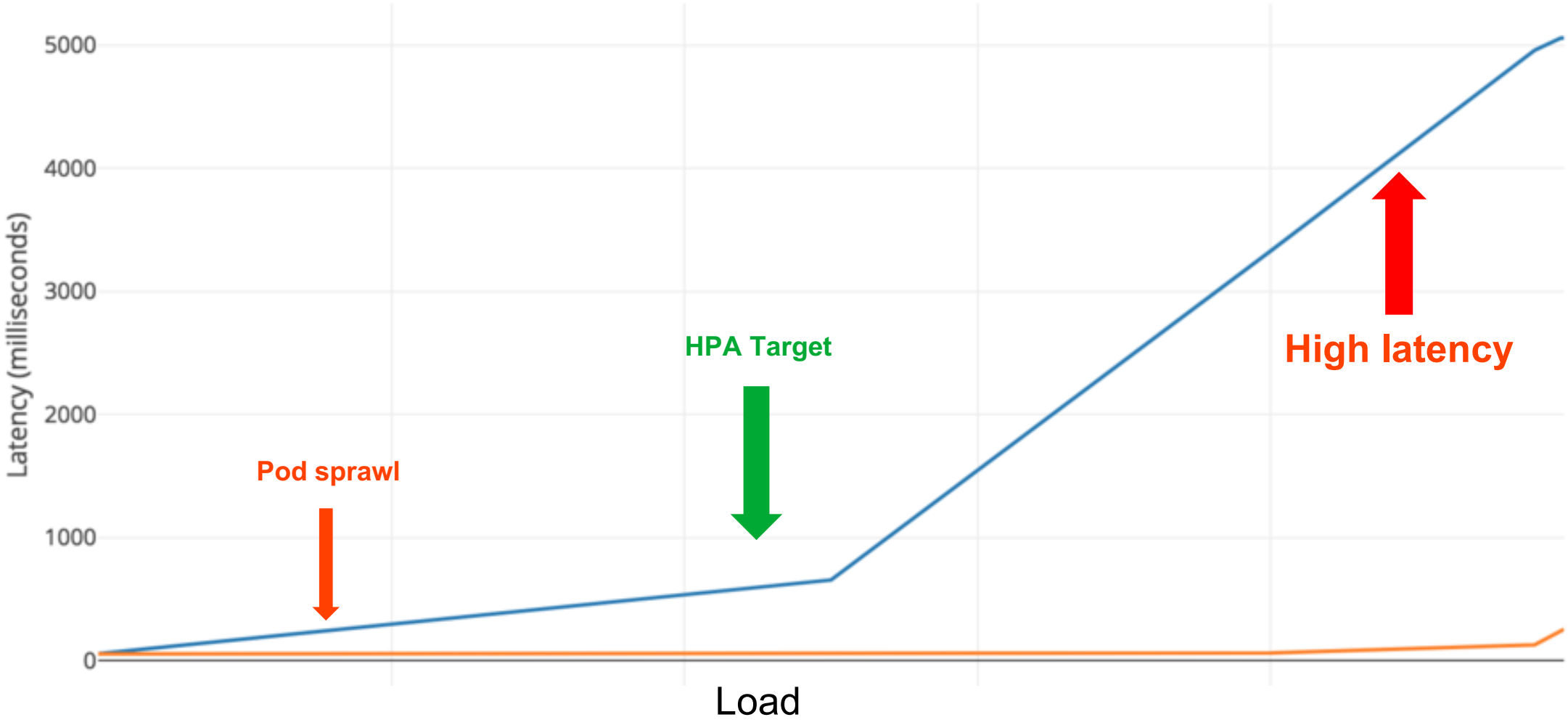
4 RPS



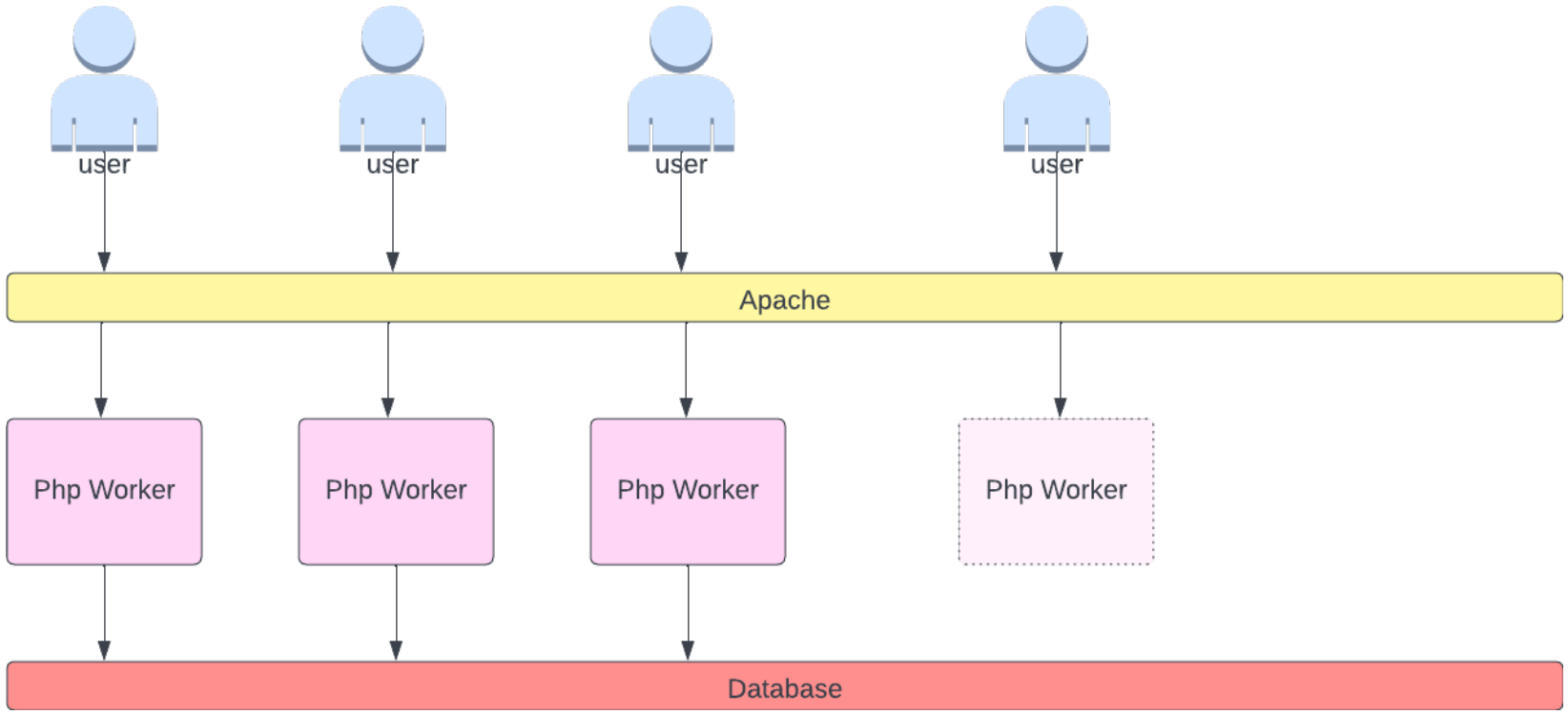
What is the right metric?

- Highly correlated
- Smooth
- Early Signal

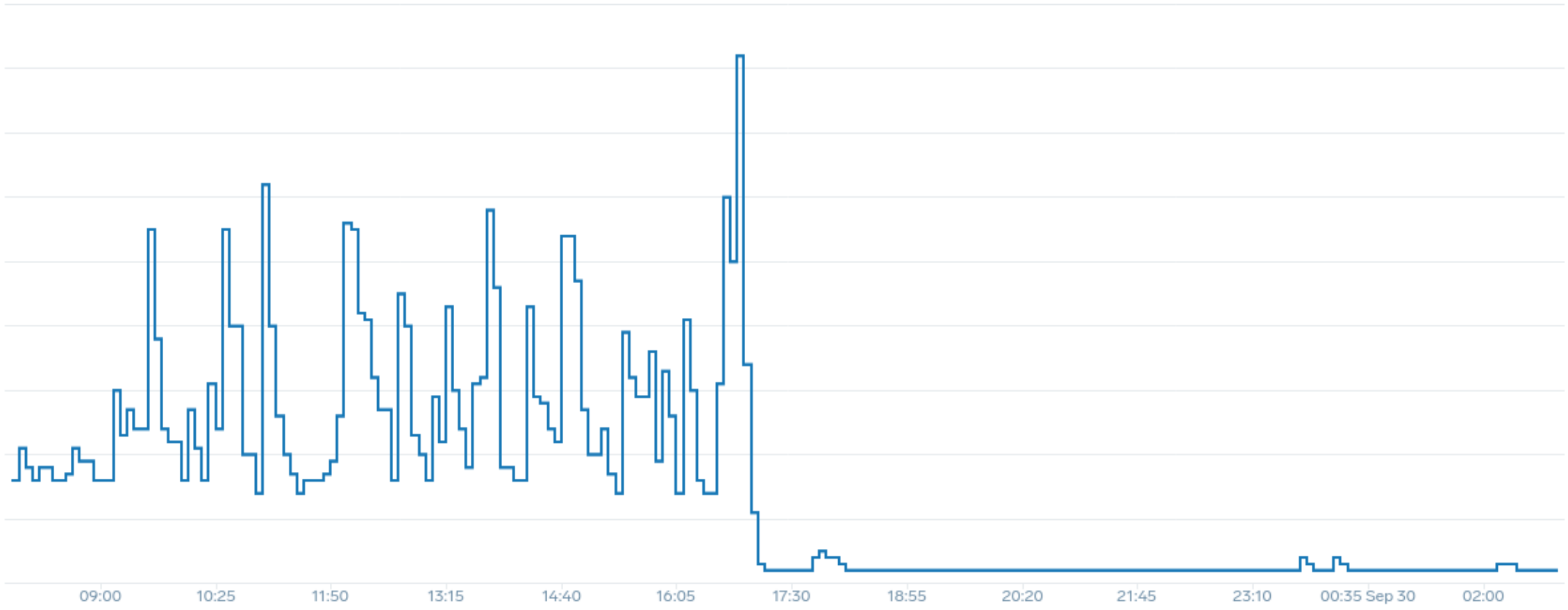
Workload Rightsizing



Active Threads



Success


09:00
10:25
11:50
13:15
14:40
16:05
17:30
18:55
20:20
21:45
23:10
00:35 Sep 30
02:00

Lessons

- Focus on the Fundamentals
- The Easy Answer is the Wrong One
- Measure Everything
- Don’t Think in “Cores”
- Don’t Scale on CPU/Memory
- Understand the Metrics

KubeCon | CloudNativeCon
North America 2022

BUILDING FOR THE ROAD AHEAD

DETROIT 2022

Session QR Codes will be sent via email before the event

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