

Building Cost-Efficient Data Processing on k8s at Pinterest

Rainie Li, Ang Zhang

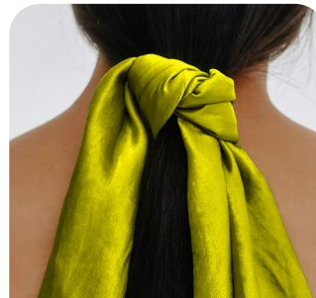


KubeCon



CloudNativeCon

Europe 2026



About Us



Rainie Li

Senior Engineering Manager

Data Processing Infra & Big Data Storage



Ang Zhang

Senior Director of Engineering

Big Data Platform



KubeCon



CloudNativeCon

Europe 2026

Agenda

- Pinterest Data Infra Overview
- Key Challenges of Resource Management
- How We Utilize K8s to Manage Resource Efficiently
- Utilizing Unused Reserved Capacity for Batch Processing
- Q&A



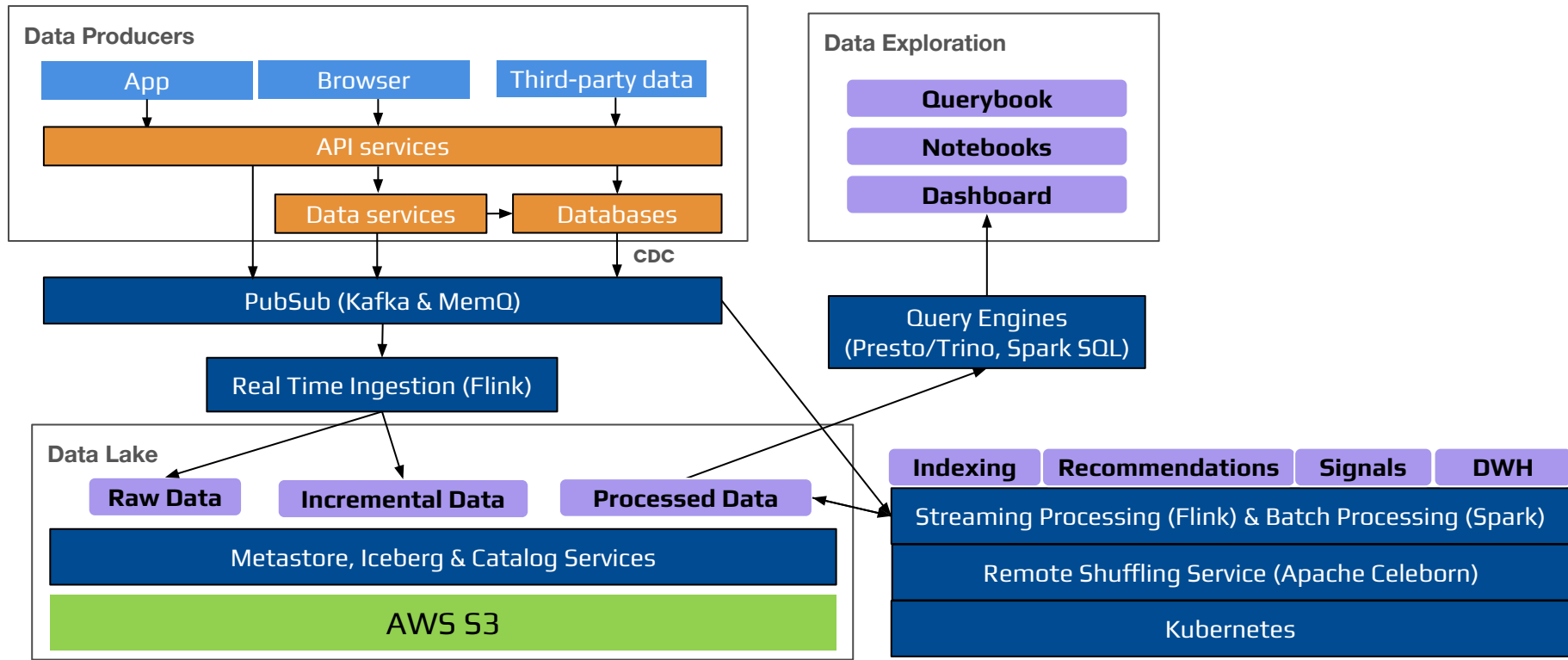
KubeCon



CloudNativeCon

Europe 2026

Pinterest Data Infra Overview



KubeCon



CloudNativeCon

Europe 2026



Platform Scale

- We run **thousands** of scheduled batch workflows/DAGs in prod.
- Process **hundreds of petabytes** data per day.
- Key use cases:
 - AI/ML pipelines: content & discovery, trust & safety, ads, etc.
 - BI & data warehouse: metrics, experimentation, etc.
 - Model development: research scientists, data scientists



Key Challenges of Resource Management

- Spark alone accounts for 1/3 of compute cost
- High priority / business critical jobs might not get enough resource to meet SLO when clusters are busy
- Cluster utilization is low during off peak hours
- Unused reserved capacity from online services are wasted and not easy to utilize for batch processing



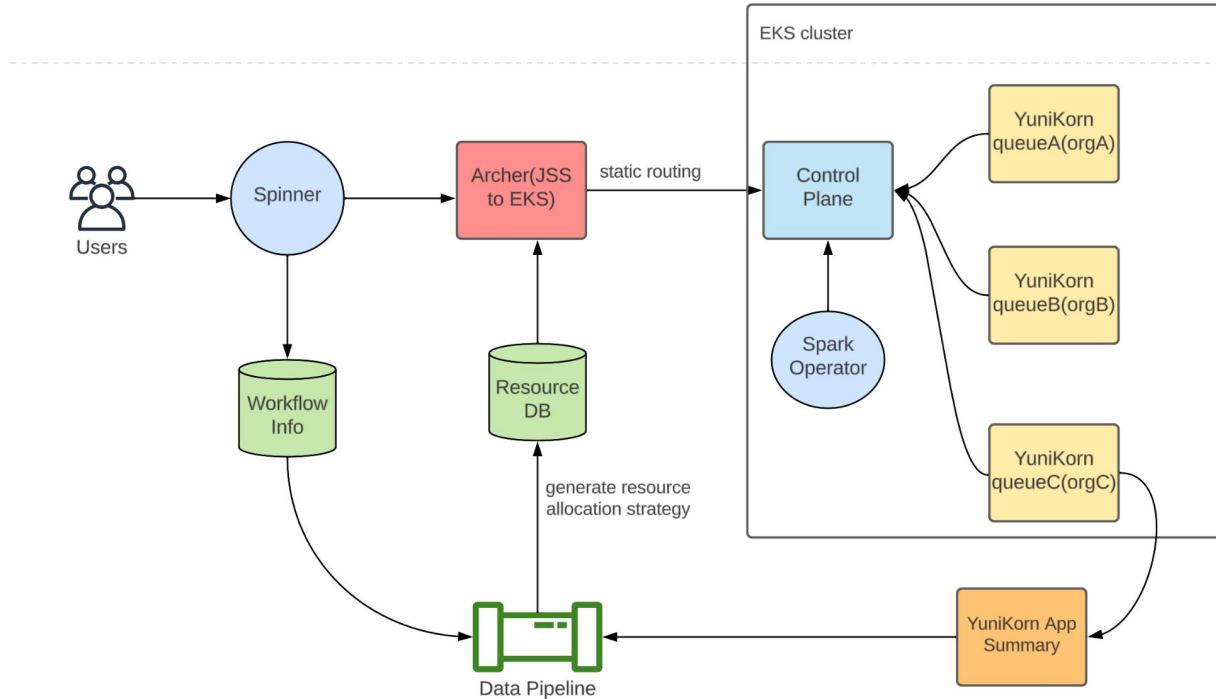
KubeCon



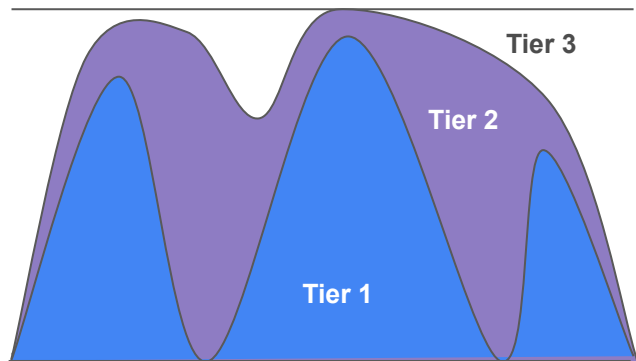
CloudNativeCon

Europe 2026

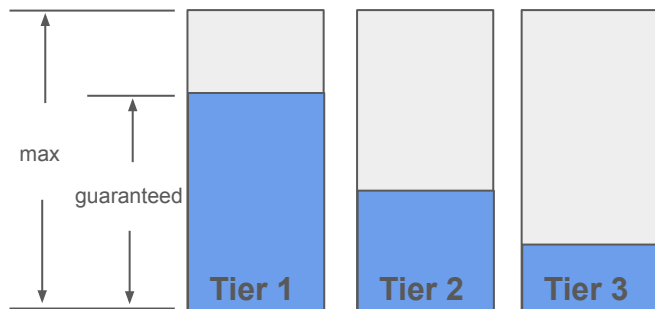
Resource Allocation During Submission



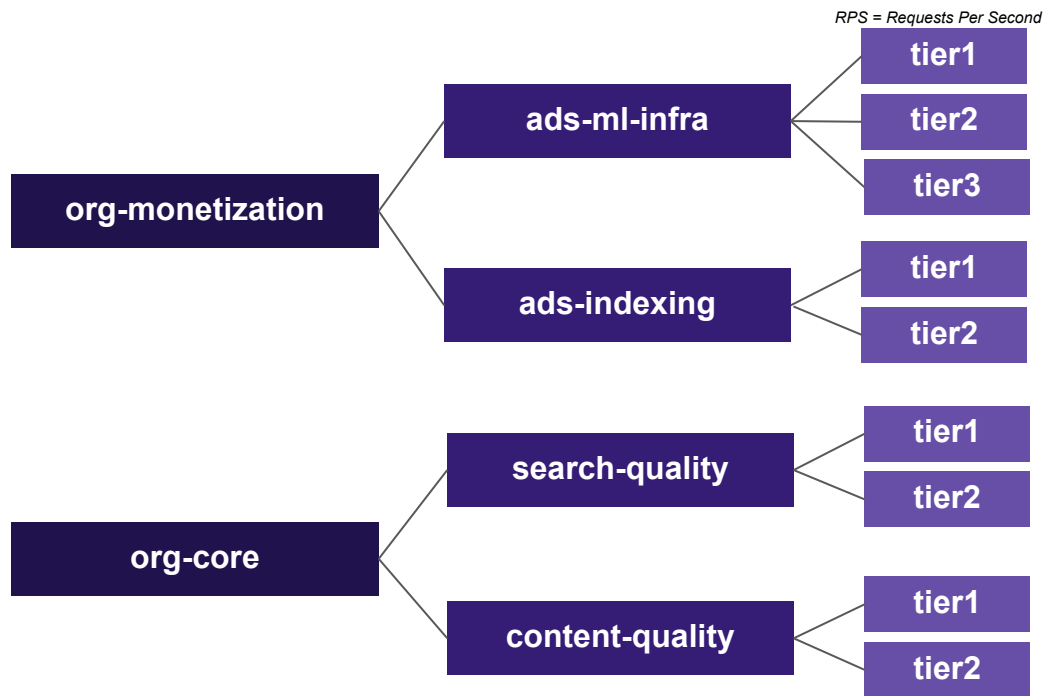
Tidered Resource Management



Prioritize resource based on tiers



Guaranteed vs max for different tiers



Hierarchical queue structure and resource allocation



KubeCon



CloudNativeCon

Europe 2026

Batch k8s Scheduler

Benefits

- Efficient resource utilization: borrow resources between queues
- Advanced scheduling features: fair scheduling, preemption, Gang etc.

Spark - Apache YuniKorn

- Easy transition from YARN
- Good UI and API
- Full queue capability: e.g max application
- Appsummary feature

Ray/PyTorch Training - Volcano

- Topology aware scheduling
- Colocation



KubeCon



CloudNativeCon

Europe 2026

Utilizing Unused Reserved Capacity for Batch Processing

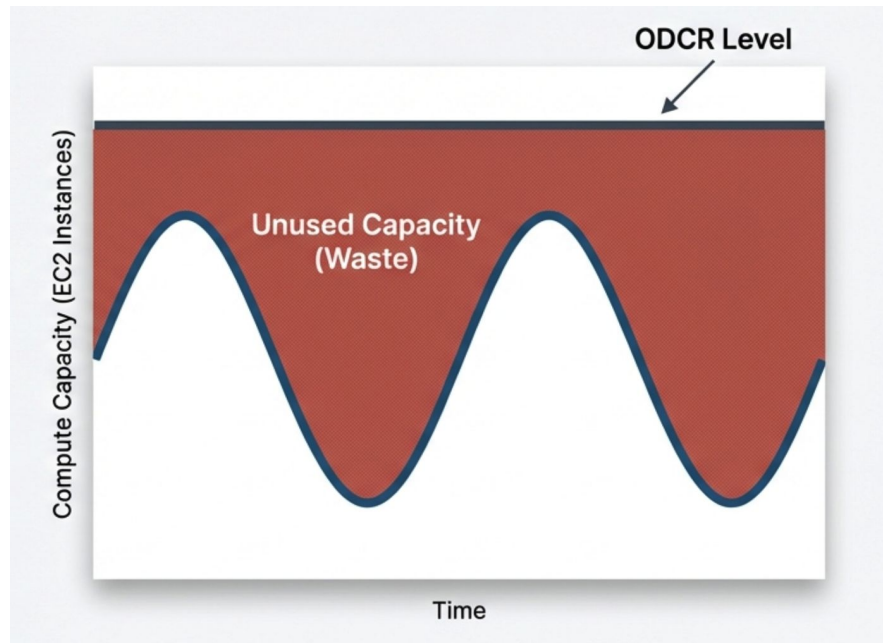
RPS = Requests Per Second

Why reserving capacity?

- Guarantee capacity when needed for critical online services either in the cloud or on-prem
- Long term commitment save cost significantly

Problem / Opportunity

- Unused capacity during non-peak hours
- Some batch applications, e.g. Spark, can run in a flexible schedule



KubeCon



CloudNativeCon

Europe 2026

Options & Tradeoffs

Option 1: running everything on same k8s clusters

Pros:

- Faster preemption and smoother transition

Cons:

- k8s scalability limitation for running offline/Spark and online applications in a large k8s cluster
- Lack of resource isolation like network and disk space/io isolation

Option 2: rebalancing the EC2 capacity from online for offline/Spark applications

Pros

- Can work using current technologies (k8s etc.)
- Fully isolate online and offline/Spark applications

Cons:

- Need to ensure returning capacity is almost instantaneous



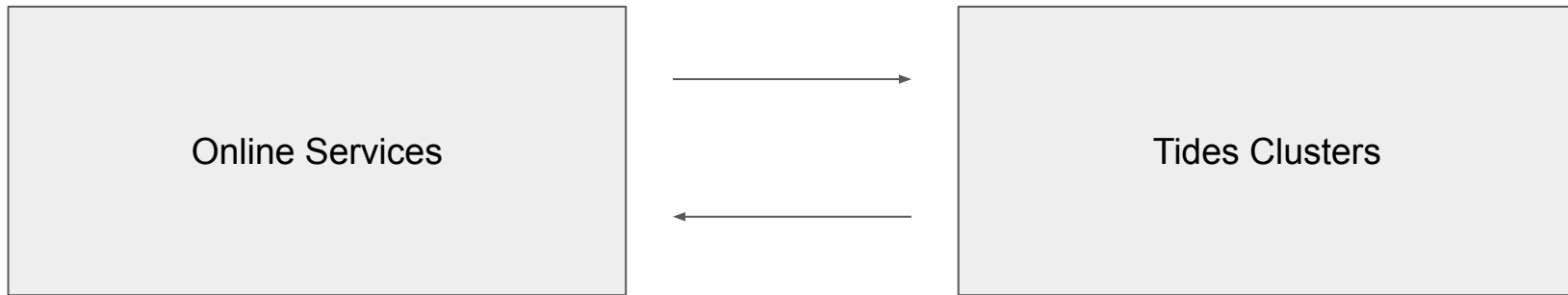
KubeCon



CloudNativeCon

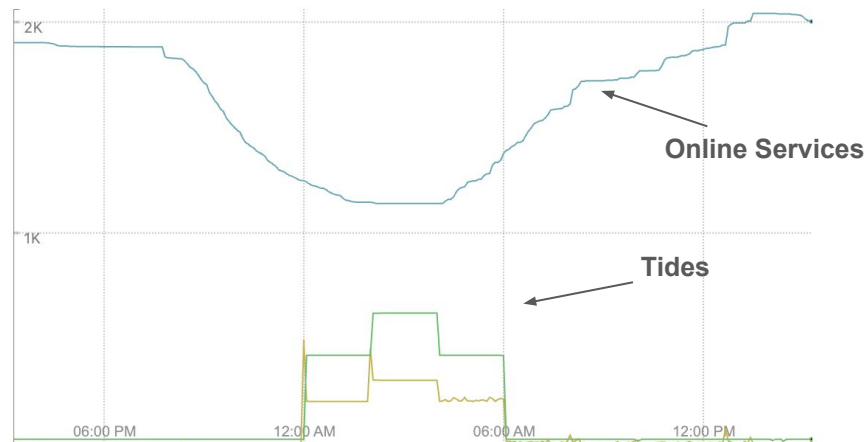
Europe 2026

Introducing Project Tides



Moving capacity from online to offline and returning it back by controlling size of Auto Scaling Groups in Tides Cluster

 **Online vs Tides Instance Counts** [Add description](#)



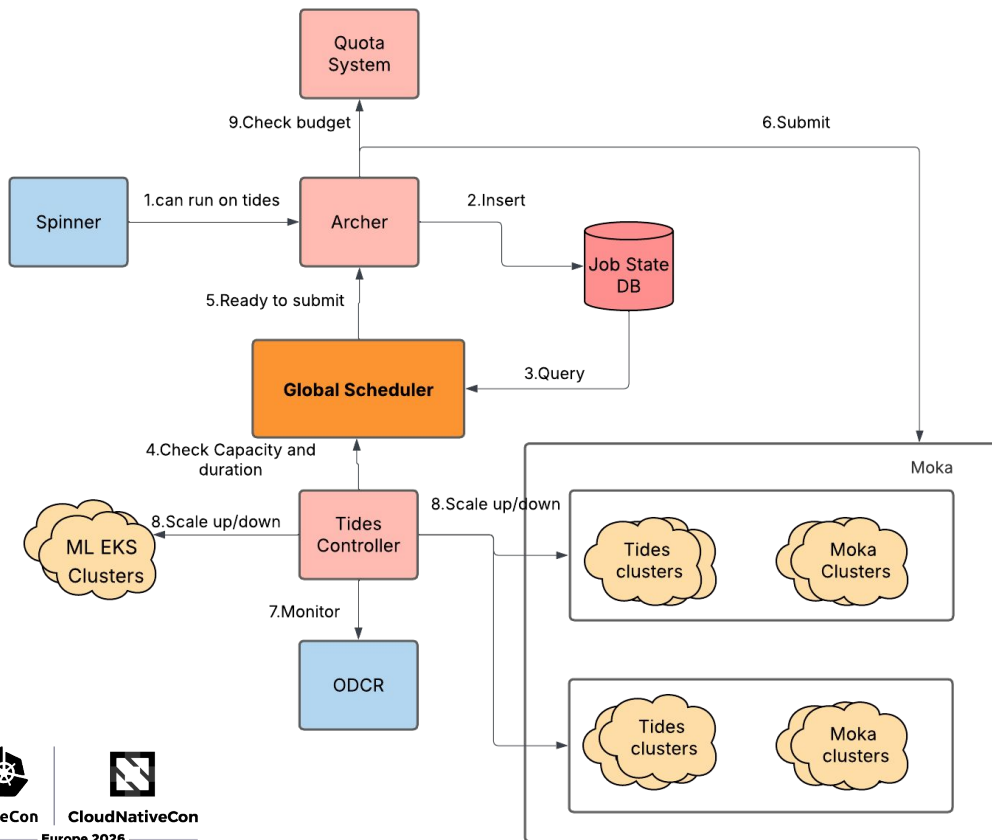
KubeCon



CloudNativeCon

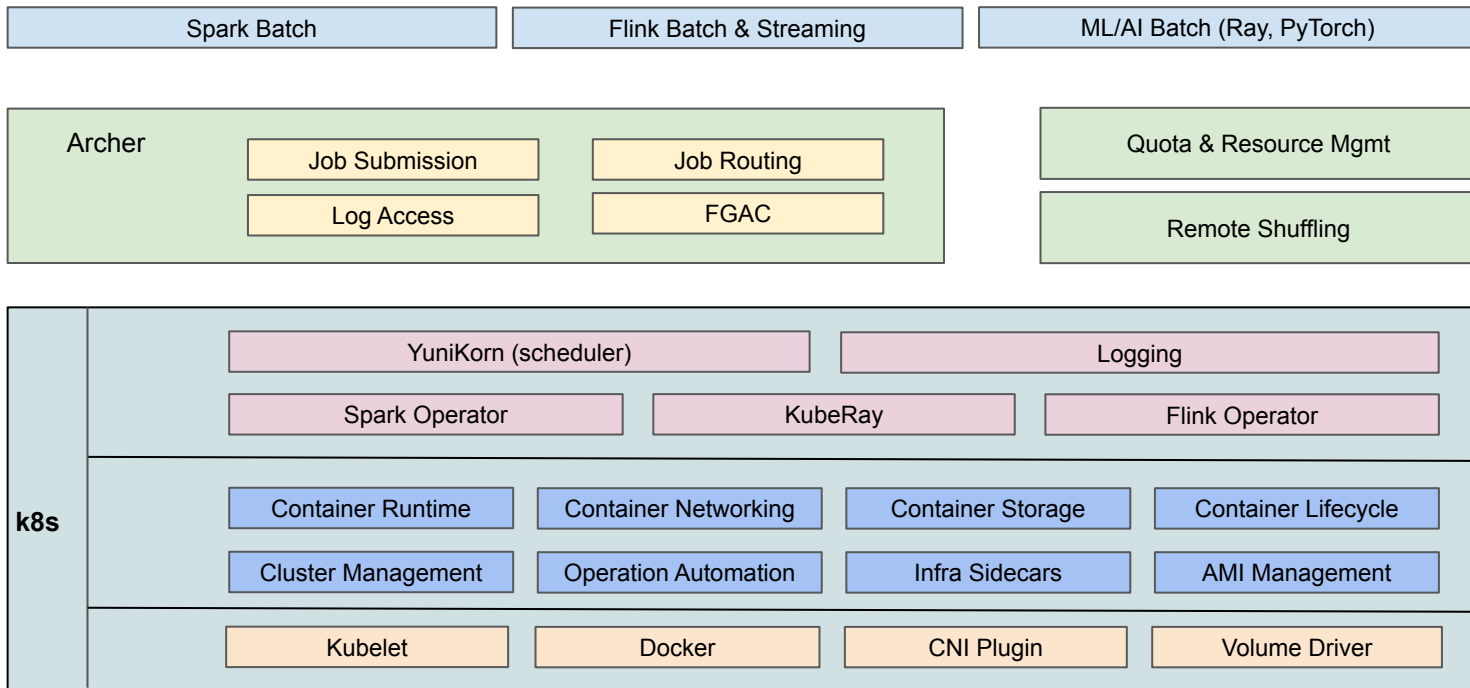
Europe 2026

Global Scheduler



- Archer: job submission service
- Global scheduler: decides when and where to schedule jobs
- Dynamic Queue Template
- Tides Controller: manage Moka clusters capacity
- Quota: manage global quota with budget

Our Data/AI on k8s Platform (Moka)



KubeCon

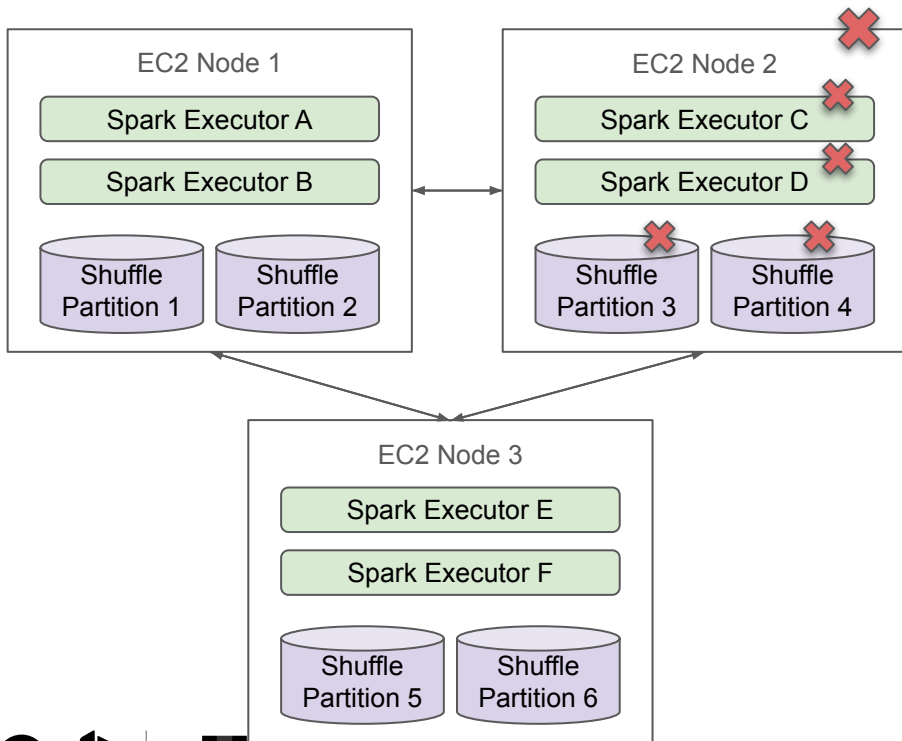


CloudNativeCon

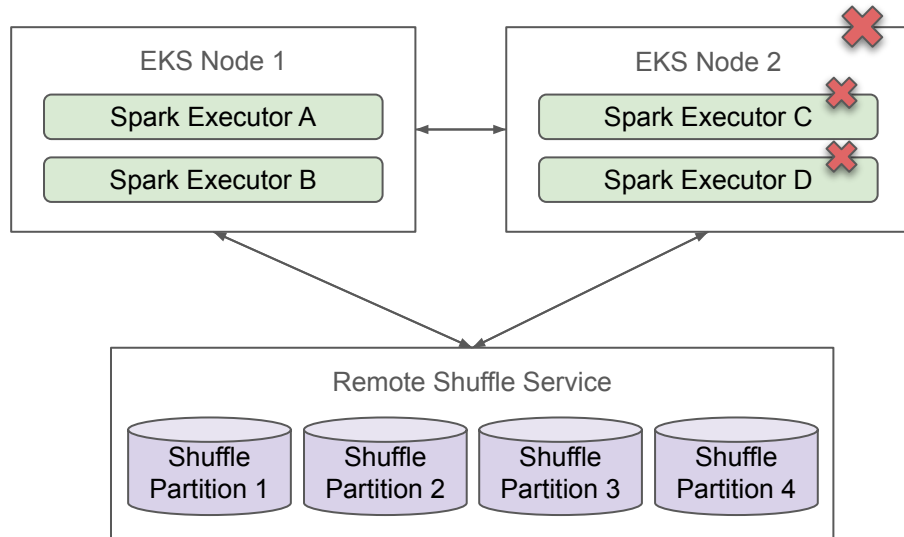
Europe 2026

How Moka Platform Makes Sharing Possible

Hadoop



k8s



KubeCon



CloudNativeCon

Europe 2026

Other Benefits of Moka That Enables Tides

- No more reliance on HDFS so that scale down won't lose shuffle data etc.
- Faster node setup comparing with Hadoop (w/o setting up Hadoop daemons)
- Unified clusters for fine grained access control
- Can utilize instances types that only use EBS because of remote shuffling



KubeCon



CloudNativeCon

Europe 2026

Challenges and Solutions

- Spark driver reliability
- Disk size & IO throughput
- Job priorities & ordering
- Opt in vs auto routing
- Capacity prediction



KubeCon



CloudNativeCon

Europe 2026

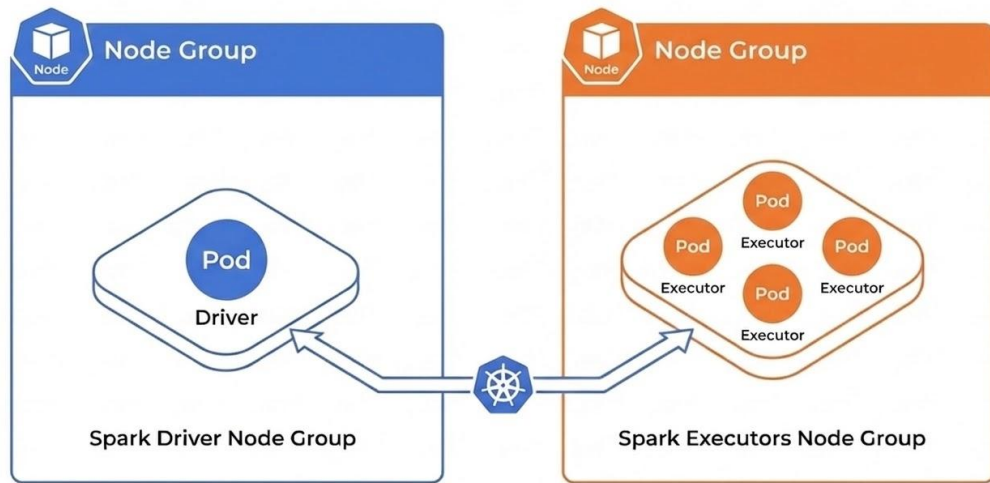
Spark driver reliability

Challenge

- Spark driver could be killed during scale down causing whole job failure

Solution

- Keep a separate small node group with static capacity for drivers



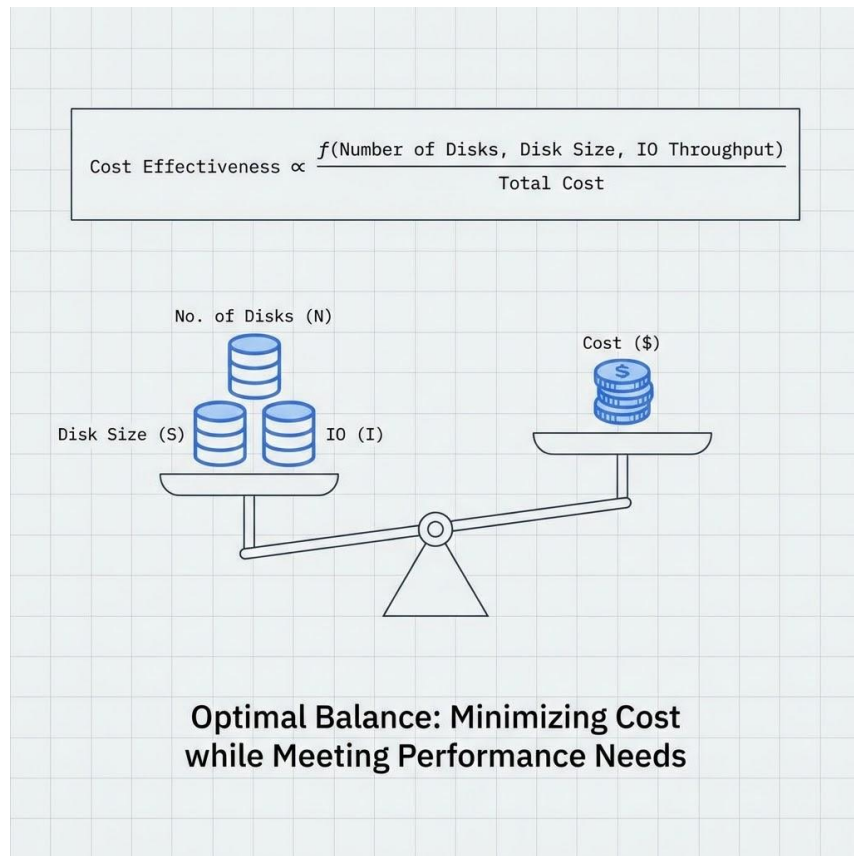
Disk size & IO throughput

Challenge

- Online service instances use block storage/EBS while Spark clusters use instance storage
- EBS can be expensive

Solution

- Balance the number of disks, size and IO to be cost effective



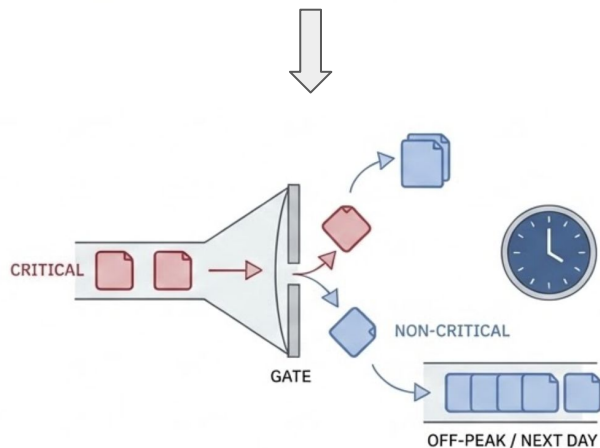
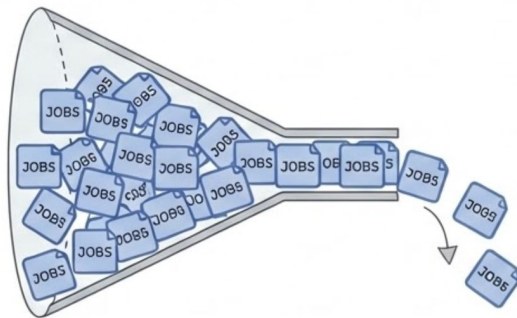
Job priorities & ordering

Challenge

- When capacity is tight, too many jobs could all compete for resource causing starvation

Solution

- Hold the jobs that are unlikely to complete before capacity is reduced and prioritize for next day



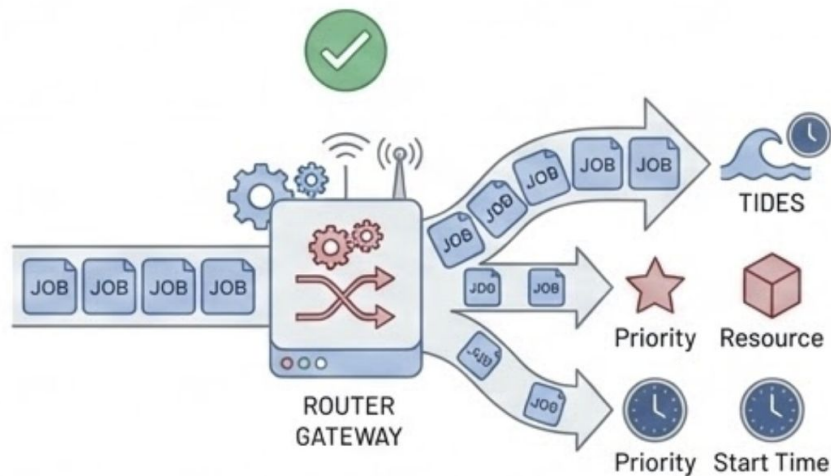
Opt in vs auto routing

Challenge

- Not many users have incentives to opt in

Solution

- Automatically select jobs to route to Tides based on priorities, resource requirements and start time



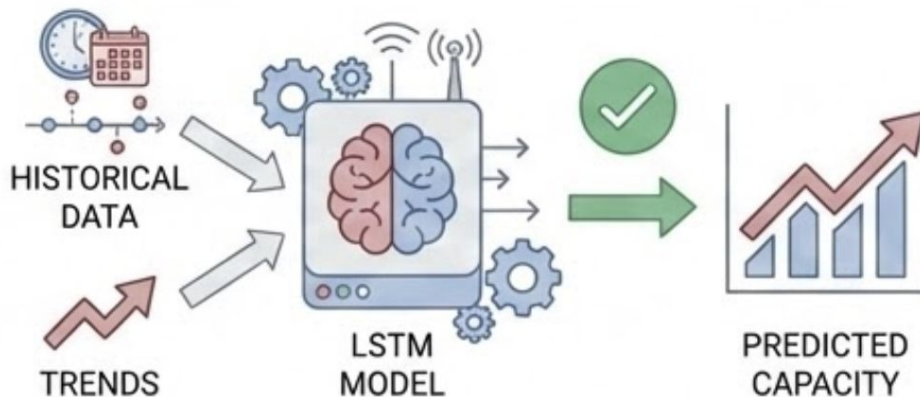
Capacity prediction

Challenge

- Unused capacity can change over time, e.g. during holidays

Solution

- Use ML model (e.g LSTM) to predict the capacity based on historical data and trend



Future Improvements

- Combine Tides scheduler with global scheduler to treat all resources and jobs the same way
- Utilize AI to further improve the capacity and resource requirement prediction



KubeCon



CloudNativeCon

Europe 2026

Q&A



KubeCon



CloudNativeCon

Europe 2026