



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Volcano

**Orchestrating the full AI lifecycle
From Training to Inference and Agents**

**Zicong Chen (Huawei Technologies)
Máté Hajnal (Aumovio)**



Orchestrating the full AI lifecycle

From Training to Inference and Agents



KubeCon



CloudNativeCon

Europe 2026



Zicong Chen

R&D Engineer, Volcano Maintainer
Huawei Technologies Co., Ltd
<https://github.com/JesseStutler>



Máté Hajnal

Senior MLOps Engineer, Volcano Contributor
Aumovio SE.
<https://github.com/hajnalmt>

Agenda

- Big Picture - From Batch System To A Unified Scheduling Platform
- Scalable and Efficient Foundation - A robust core for ALL workloads (Microservices, Batch, AI, HPC) at ultra-large scale.
- What's new in Volcano Network Topology Aware Scheduling
- Kthena - Enterprise-Grade LLM Serving
- AgentCube - Cloud-Native Agent and Code Interpreter Platform
- Volcano Global - Multi-Cluster Federation and Roadmap



KubeCon



CloudNativeCon

Europe 2026

From Batch System To A Unified Scheduling Platform



Volcano Overview

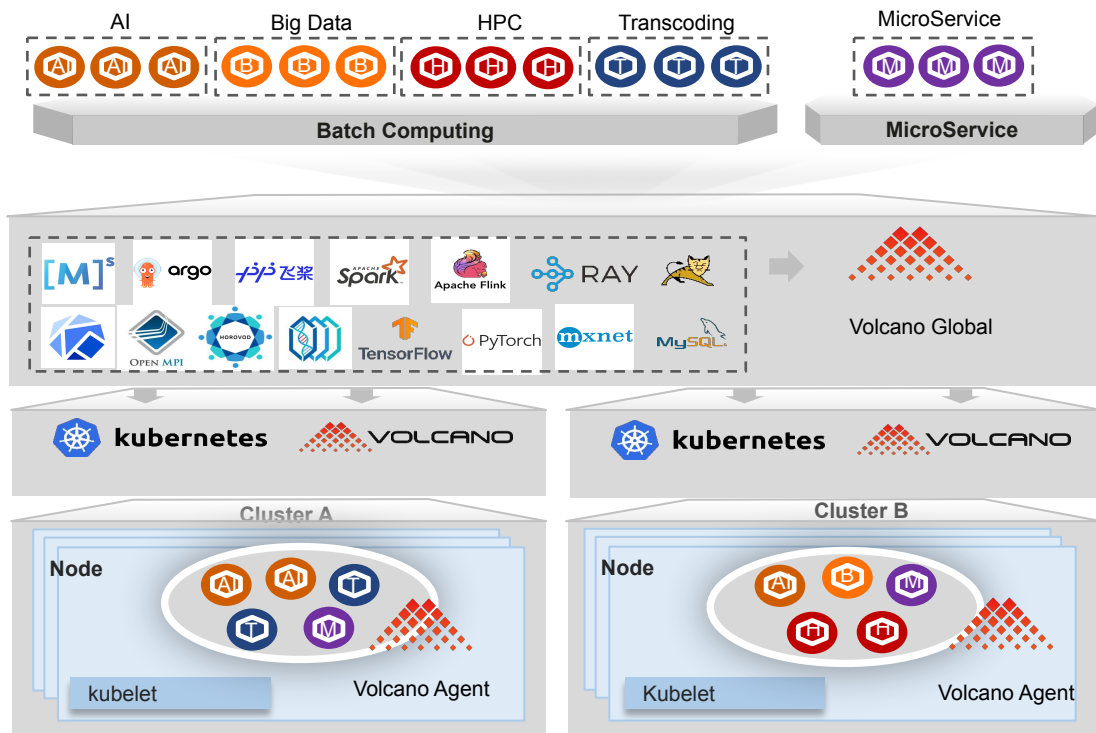


KubeCon



CloudNativeCon

Europe 2026



Unified Scheduling

Supports integrated job scheduling for both Kubernetes native workloads and mainstream computing frameworks (such as TensorFlow, Spark, PyTorch, Ray, Flink, etc.).

Queue Management

Provides multi-level queue management capabilities, enabling fine-grained resource quota control and task priority scheduling.

Workloads Colocation

Enables online and offline workloads colocation, improving cluster resource utilization through intelligent scheduling strategies.

Multi-cluster Scheduling

Supports cross cluster job scheduling, improving resource pool management capabilities and achieving large scale load balancing.

Heterogeneous Device Support

Efficiently schedules heterogeneous devices like GPU and NPU, fully unleashing hardware computing potential.

Multiple Scheduling Policies

Supports various scheduling strategies such as Gang scheduling, Network Topology Aware Scheduling, Fair-Share, Binpack, DeviceShare, NUMA-aware scheduling, Task Topology, etc.

Why now: the workload drift



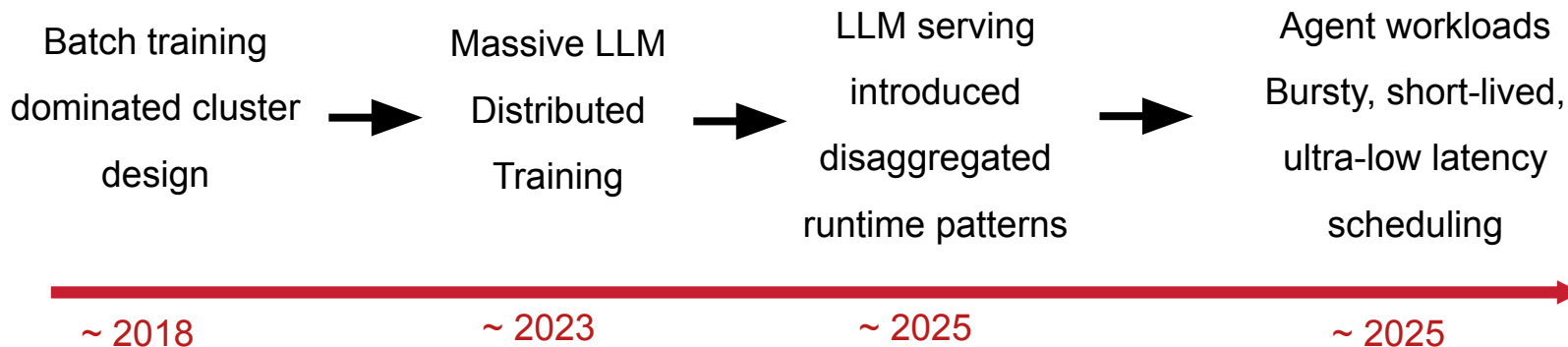
KubeCon



CloudNativeCon

Europe 2026

AI workloads are diverging fast.



New requirement: one control plane, multiple execution patterns

One platform shall optimize for throughput and latency together

Volcano Technical Roadmap for 2026 and Beyond



KubeCon



CloudNativeCon

Europe 2026

Application & Workload Ecosystem

AI Training & RL

Distributed Training, Fine-tuning, RLHF

Megatron | DeepSpeed | vRL | AReAL

Distributed Inference

PD-disaggregated Inference

vLLM | SGLang | MindIE | TensorRT

AI Agent

Agent Workflow & RAG

LangGraph | Dify | A2A

Big Data & HPC

Batch / Scientific Computing

Spark | Flink | Horovod | Ray

Observability & Ops Ops & Tooling

Observability

Job Metrics, Tracing, Dashboard,
Resource Heatmap

CLI & SDK

Volcano CLI, JobFlow SDK, Python
Client

Ecosystem Integration

Kubeflow, Spark/Flink Operator, Argo
Workflow, Kserve, AIBox, LLM-D,
AgentSandbox and more...

Workloads & Job Orchestration

HyperJob

LLM Training Parallelism

Model/Parameter/Data/ Parallelism

Kthena

Elastic Inference Service

PD Disaggregation / Large EP

AgentCube

Multi-Agent Orchestration

Hibernate/Resume / Fast-Startup

VolcanoJob

Unified Job Abstraction

Batch / MPI / Pytorch / TF

...

Volcano Core Scheduler

Multi-tenant Resource Sharing

- Hierarchical Queue Management
- SLO-aware Preemption
- Dynamic Resource Oversubscription

High Throughput Scheduling

- Gang Scheduling / Binpack
- Fair Scheduling (FairShare / DRF)
- AI Topology Awareness (All-Reduce Opt)

Multi-cloud & Federation

- Multi-cluster Job Orchestration
- Adaptive Decomposition for Hyper-scale Training
- Cross-domain Resource Sharing & Training Over Heterogeneous Clusters

Unified High-Performance Scheduling Framework

Unified high-throughput scheduling for Microservices, Batch, Distributed Inference, AI Agents, Serverless workloads, etc.

Resource Management

Heterogeneous Pool

Fine-grained GPU/NPU Sharing & Isolation

Topology Abstraction

Intra-node / Inter-node (HyperNode) Topology

Device Access

Device Plugin | DRA | NRI | CDI

Infrastructure & Runtime

Sandboxer

Kuasar | Kata | WASM | gVisor
Firecracker | CloudHypervisor

OS (Colocation)

openEuler | SUSE | Ubuntu | RHEL | more...

Hardware

CPU / GPU / NPU / TPU
RoCE / InfiniBand / FPGA



KubeCon



CloudNativeCon

Europe 2026

Scalable and Efficient Foundation



Agent scheduler: Fast Scheduling for AI Agent Workloads



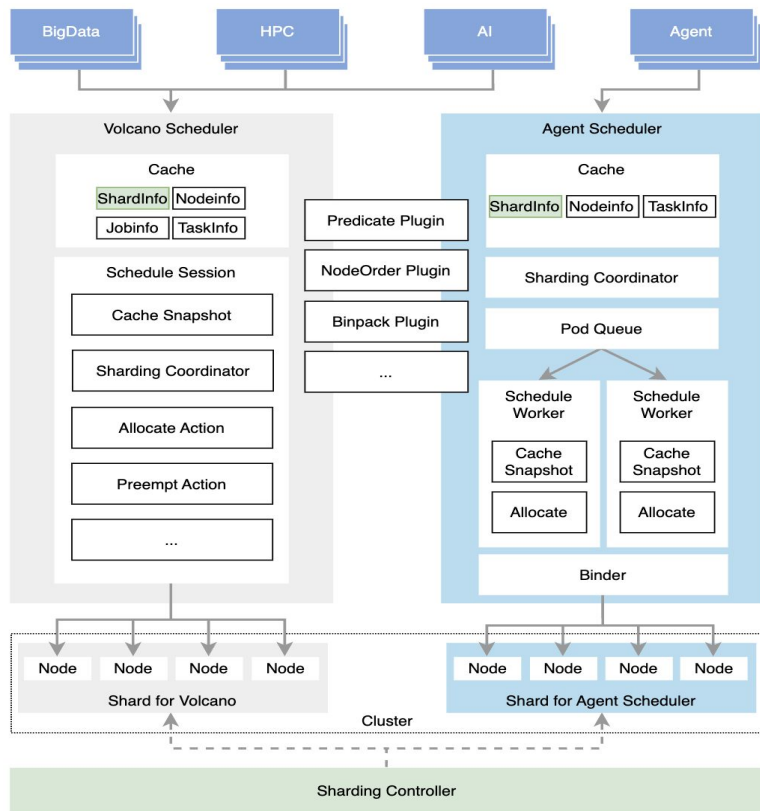
KubeCon



CloudNativeCon

Europe 2026

- **Dedicated fast scheduling** path for latency-sensitive workloads (e.g. agent workloads)
- Multi-worker parallel scheduling
- Optimistic concurrency + conflict-aware binding
- Optimized Scheduling Queue: Enhanced queue mechanism with urgent retry support
- Seamless coordination with Volcano batch scheduler via Sharding Controller



Scalable Multi-Scheduler with Dynamic Node Scheduling Shard



KubeCon

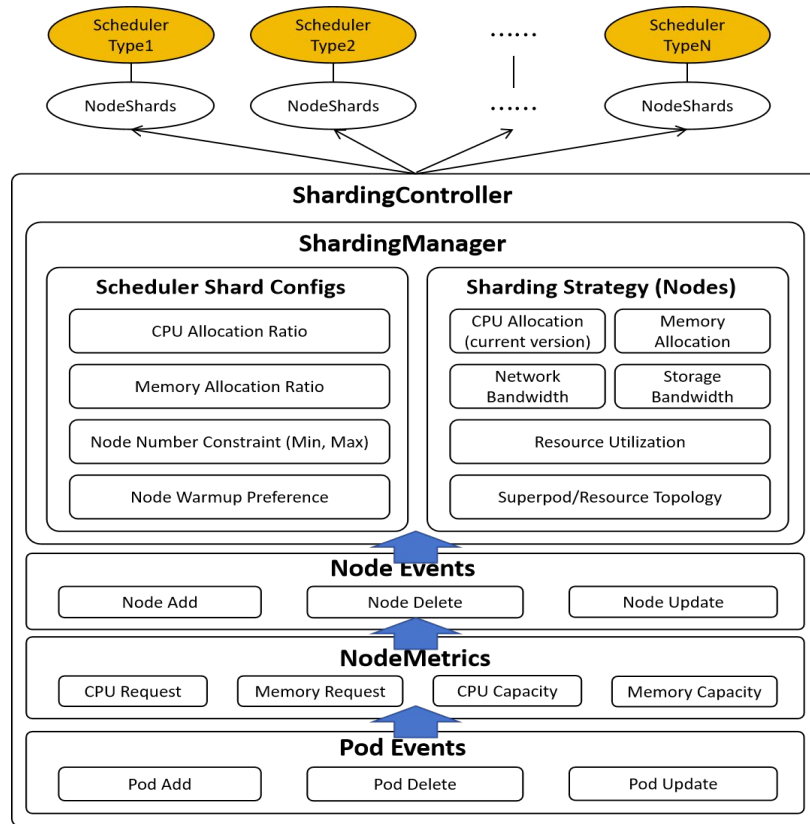


CloudNativeCon

Europe 2026

Specialized Scheduling, Resource Optimization Scalability, Workload Isolation

- Sharding Controller dynamically computes node pools based on extensible policies rather than inefficient static hard partitioning.
- Enables flexible scheduler combinations (e.g., multiple Batch Schedulers, or Agent + Batch) coordinating via the NodeShard CRD.



NodeShard CRD

- NodeShard CRD becomes the coordination contract
- **Dynamic Candidate Node Pools:** Replaces static hard-partitioning with dynamic node pools, improving cluster utilization.
- **Pluggable Sharding Strategies:** Extensible policy design to compute candidate nodes (e.g., CPU utilization strategy available now).

```
scheduler-configs:
- name: agent-scheduler
  type: agent
  cpu-utilization-min: 0.7
  cpu-utilization-max: 1.0
  prefer-warmup-nodes: true
  min-nodes: 1
  max-nodes: 100
```

```
scheduler-configs:
- name: volcano-scheduler
  type: volcano
  cpu-utilization-min: 0.0
  cpu-utilization-max: 0.69
  prefer-warmup-nodes: false
  min-nodes: 1
  max-nodes: 100
```

Sharding Configuration

```
apiVersion: shard.volcano.sh/v1alpha1
kind: NodeShard
metadata:
  name: volcano
spec:
  # Nodes should be used within this shard.
  nodesDesired:
    - node1
    - node2
    - node3
status:
  lastUpdateTime: "2025-12-08T06:00:00Z"
  nodesInUse:
    - node0
    - node1
    - node2
  # Node should be remove from scheduler.
  nodesToRemove:
    - node0
  # Node should be added to scheduler.
  nodesToAdd:
    - node3
```

Node Shard Example

What's new in Volcano Colocation



KubeCon



CloudNativeCon

Europe 2026

Different QoS Model:

LC (Latency Critical)
HLS (Highly Latency Sensitive)
LS (Latency Sensitive)
BE (Best Effort)

} Online
→ Offline

Multi-Workload support

Microservices | AI | Big Data | HPC

Comprehensive QoS guarantees

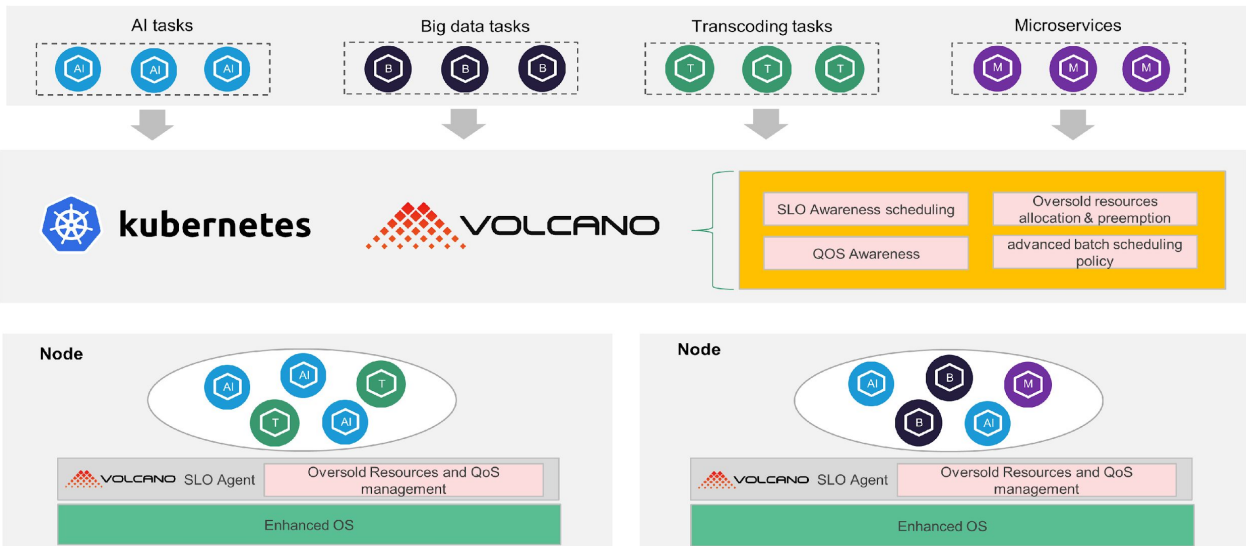
Protects online services by suppressing or evicting offline jobs during resource contention

Dynamic Oversubscription

+30% utilization via real-time reclaim

CPU Burst

Allow to temporarily exceed CPU limits



What's new in Volcano Colocation



KubeCon



CloudNativeCon

Europe 2026

- Expanded OS Support: Fully decoupled from specific distros, **now supporting generic operating systems (Ubuntu, CentOS, etc.)**.
- Dynamic CPU Throttling & Burst: Adaptive CPU quota adjustment for offline pods to protect online workloads, plus CPU Burst for latency-sensitive tasks.
- Full Cgroup v2 Integration: Auto-detects cgroup versions and natively implements v2 resource managers.
- Fine-grained Memory QoS
- Declarative Policy API: Introduced ColocationConfiguration CRD

```
apiVersion: config.volcano.sh/v1alpha1
kind: ColocationConfiguration
metadata:
  name: colo-config1
spec:
  selector:
    matchLabels:
      app: offline-test
  memoryQos:
    # Memory throttling ratio
    highRatio: 100
    # Memory priority protection ratio
    lowRatio: 0
    # Absolute memory protection ratio
    minRatio: 0
```



KubeCon



CloudNativeCon

Europe 2026

Volcano Topology Aware Scheduling: What's new?



Volcano Topology Aware Scheduling



KubeCon



CloudNativeCon

Europe 2026

Large-scale AI clusters lack unified topology abstraction,
requiring optimized intra- and inter-node configurations for
maximum performance

- v1.12: HyperNode-based topology-aware scheduling reaches alpha
- v1.13: label-based HyperNode auto-discovery added
- v1.14: richer subgroup-level constraints and gang coordination;
HyperNode-level binpacking reduces fragmentation

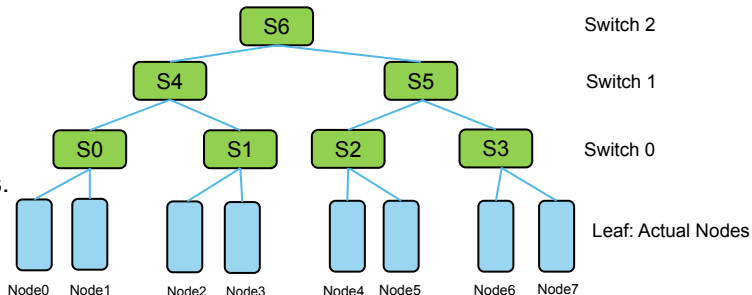
Non-Leaf HyperNodes (s4, s5, s6):

Represent other HyperNodes.

Leaf HyperNodes (Node[01]):

Whose members are actual cluster nodes.

Standardizing Cluster Topologies via
Performance Domains



```
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s0
spec:
  tier: 1
  members:
  - type: Node
    selector:
      regexMatch:
        pattern: node-[01]
```

```
apiVersion: batch.volcano.sh/v1alpha1
kind: Job
metadata:
  name: job-name
spec:
  networkTopology:
    mode: hard
    highestTierAllowed: 2
  ...
```

Multi-Level Network Topology Scheduling



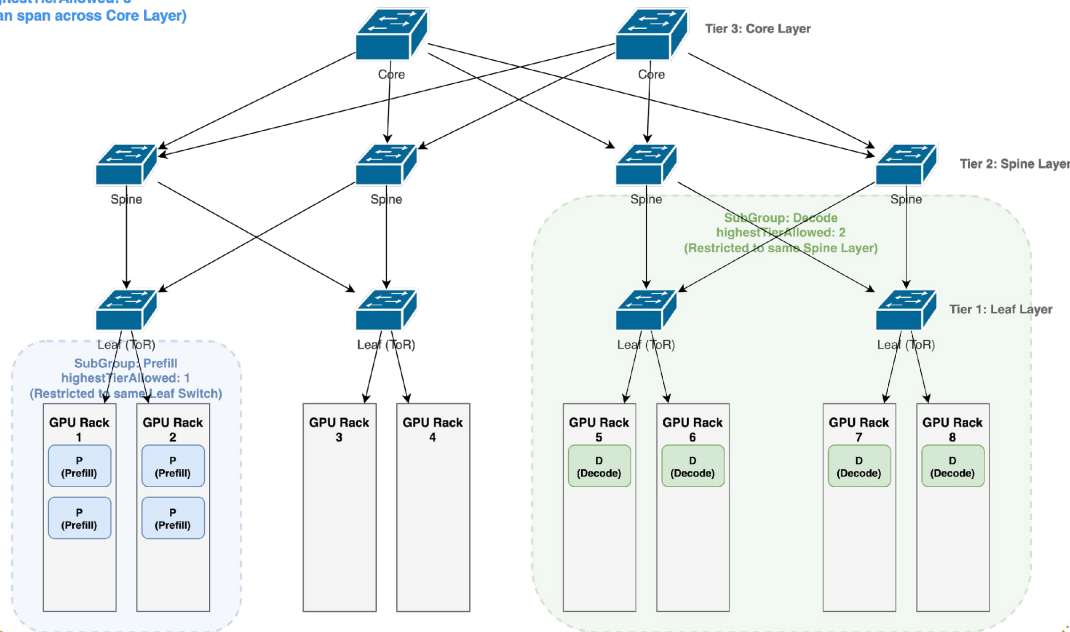
KubeCon



CloudNativeCon

Europe 2026

PodGroup (LLM Inference Job)
highestTierAllowed: 3
(Can span across Core Layer)



```
apiVersion: scheduling.volcano.sh/v1beta1
kind: PodGroup
metadata:
  name: llm-inference-pg
spec:
  minMember: 8
  networkTopology:
    mode: hard
    highestTierAllowed: 3 # Job-level: Core Layer
  subGroupPolicy:
    - name: "prefill"
      subGroupSize: 4
      networkTopology:
        mode: hard
        highestTierAllowed: 1 # SubGroup 1: Leaf Layer
    - name: "decode"
      subGroupSize: 4
      networkTopology:
        mode: hard
        highestTierAllowed: 2 # SubGroup 2: Spine Layer
```

PodGroup
(highestTierAllowed: 3
minMember:8)

SubGroup-prefill
(highestTierAllowed: 1
minMember:4)

SubGroup-Decode
(highestTierAllowed: 2
minMember:4)

Volcano Topology Aware Scheduling: HyperNode Pluggable Auto-Discovery



KubeCon



CloudNativeCon

Europe 2026

AI Cluster & Cloud Environment



NVIDIA



On-Premise:

1. NVLink Fabric Manager API
2. IB Subnet Manager (OpenSM)
3. RoCE LLDP/CDP



Google Cloud

Cloud:

1. Cloud Provider APIs
2. DPU/SmartNIC Telemetry

Report
Network
Topology



VOLCANO
Volcano Control Plane

Discovery
Manager

UFM
Discoverer

NodeLabels
Discoverer

RoCE
Discoverer

other
Discoverers

Update
HyperNode
Hierarchy
Tree

HyperNode
Controller

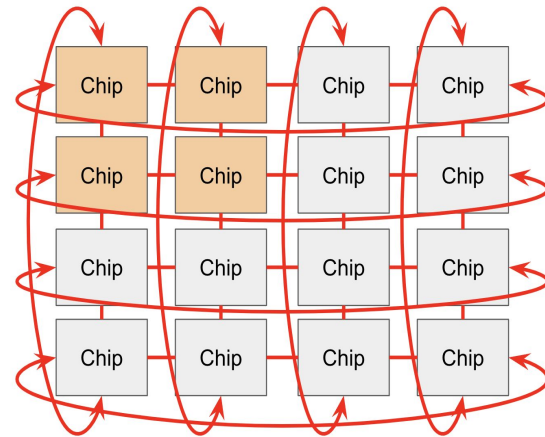
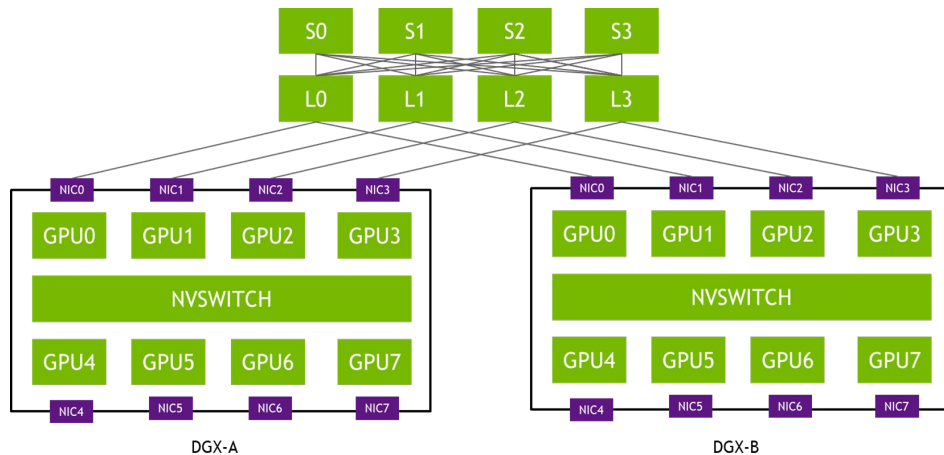
Create
/Update
HyperNode

HyperNode API Objects

Example Hypernode

```
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3
  members:
  - type: HyperNode
    selector:
      exactMatch:
        name: s4
status:
  conditions:
  type: Ready
  status: "True"
  lastTransitionTime: 2025-10-31 10:08:08
```

Volcano Topology Aware Scheduling: What's Next?



1. Expand fields and capabilities, and collaborate with other resources (e.g. DRA) to support more networking schemes, such as multi rail, 3D torus topologies

Volcano Topology Aware Scheduling: What's Next?



KubeCon



CloudNativeCon

Europe 2026

2. Integrate more network topology discovery tools to fully automate HyperNode and reduce the configuration pressure of admins

- Standard Network Protocols:

LLDP / SNMP for generic Ethernet and RoCEv2 networks.

- Open Source Providers:

NVIDIA Topograph, and DCIM (Data Center Infrastructure Management) systems.

- Cloud & Fabric Managers:

Public cloud topology APIs (AWS, GCP, etc.).

What's Next: The HyperNode Initiative



KubeCon



CloudNativeCon

Europe 2026

3. Promote HyperNode as a **K8s First-Class Citizen**

- Extract HyperNode:
Decouple Volcano's HyperNode into an independent, vendor-neutral API standard.
- Call for Co-creation HyperNode Initiative:
Co-create HyperNode initiatives (e.g., via CNCF TOC) to jointly define and manage HyperNode standards.

Node Label



Kueue:
Topology
CRD+
Node Label



KAI Scheduler:
Topology CRD+
Node Label



Koordinator:
ClusterNetwork
kTopology
CRD+
Node Label

HyperNode CRD



VOLCANO

- (1) Both Node Labels and auto-discovery are supported
- (2) HyperNode level health monitoring, network congestion status monitoring
- (3) HyperNode level bin-pack and other customize scheduling policies



KubeCon



CloudNativeCon

Europe 2026

Kthena: Kubernetes-Native LLM Serving Platform



Kthena: Kubernetes-Native LLM Serving Platform

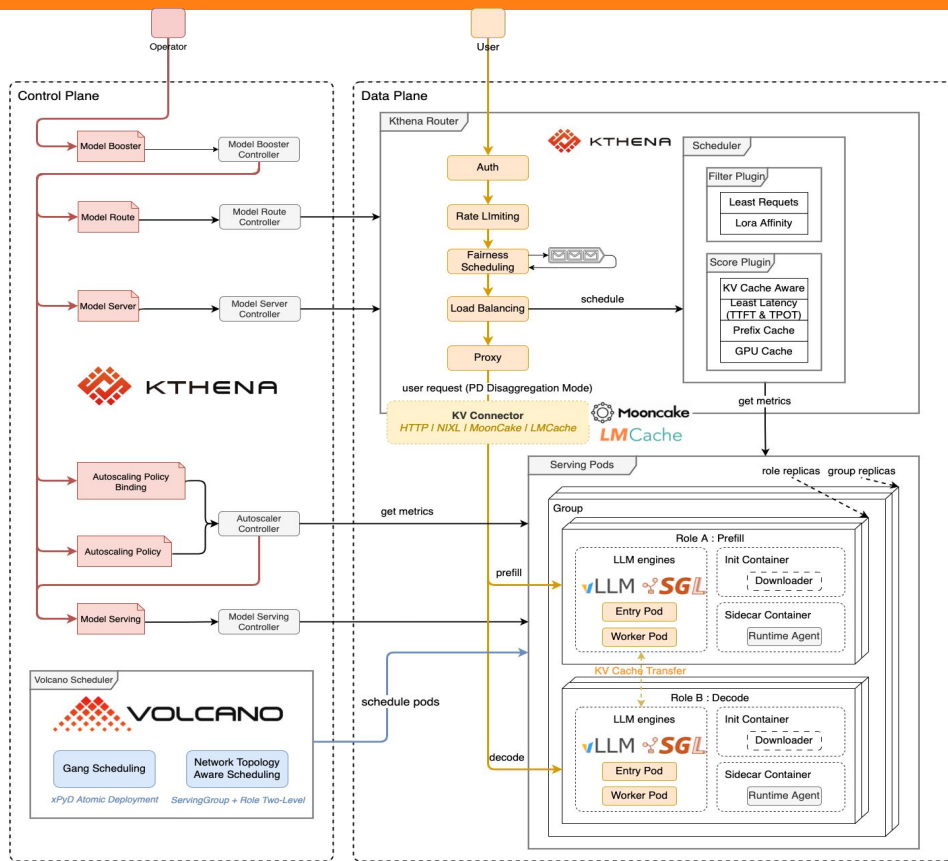


KubeCon



CloudNativeCon

Europe 2026



1. Declarative Control Plane

- Supports vLLM, SGLang, Triton, TorchServe, MindIE (Support both standard and prefill-decode disaggregated)
- Autoscaling on token throughput, queue depth, CPU/GPU, and budget
- Rolling updates, canary, failover, dynamic LoRA management

2. Intelligent Data Plane

- KV-cache, prefix-cache, PD-group aware routing via Redis
- Pluggable KV connectors (LMCache, NIXL, Mooncake) for PD disaggregation
- Token rate limiting, weighted traffic

3. Volcano-Native Scheduling

- Gang scheduling for atomic xPyD deployment
- Two-level network topology-aware scheduling (ServingGroup and Role)

Kthena: Native Prefill-Decode Disaggregation

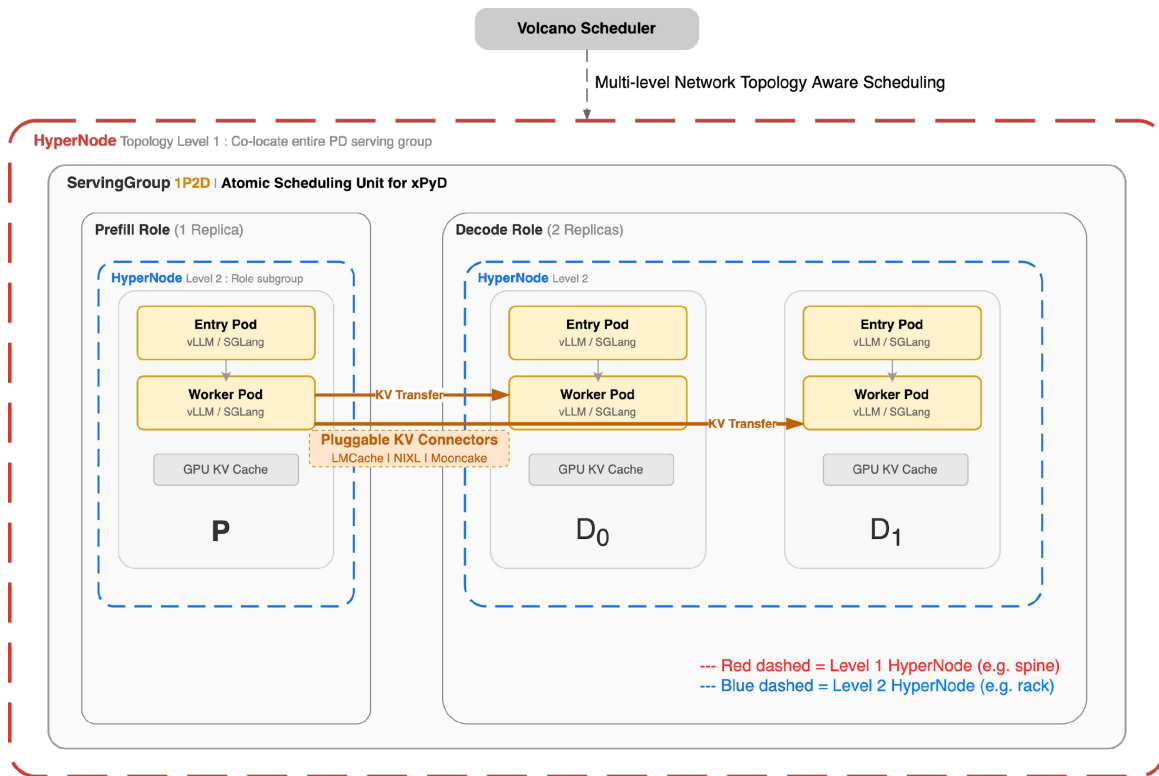


KubeCon



CloudNativeCon

Europe 2026



- ServingGroup as atomic scheduling unit for xPyD (1P1D, 1P3D, etc.)
- Gang Scheduling: all pods in a PD group scheduled together or not at all
- Multi-level network topology-aware scheduling:
 - ServingGroup level:** co-locate entire PD group in same HyperNode
 - Role level:** co-locate Entry + Worker within each Prefill/Decode role
- Pluggable KV Connectors (LMCache, NIXL, Mooncake) for KV cache transfer between P and D

Kthena Router: LLM-Native Intelligent Routing

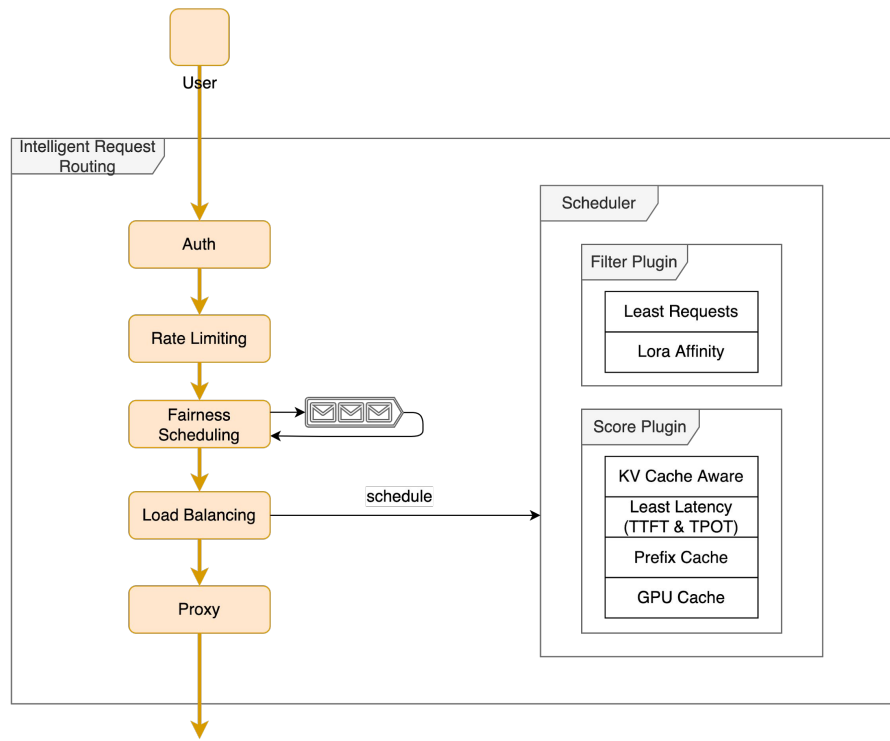


KubeCon



CloudNativeCon

Europe 2026



Smart Pluggable Routing Engine

- *prefix-cache* — route to pod with matching prompt prefix
- *kvcache-aware* — Redis-based distributed KV block matching
- *least-request* — avoid pods with long queues
- *least-latency* — prefer lower TTFT / TPOT
- *lora-affinity* — route to LoRA-loaded pod
- *gpu-usage* — prefer free GPU KV cache

Token-Level Traffic Management

- Token-based rate limiting (input + output, local + global Redis)
- Per-user fairness on historical token usage
- Weighted splitting for canary / A-B / multi-LoRA

Native PD Disaggregation Routing

- PD-group aware request splitting (Prefill then Decode)
- KV Connectors: HTTP / NIXL / LMCache / Mooncake
- Independent scoring per phase, automatic KV transfer



KubeCon



CloudNativeCon

Europe 2026

AgentCube: A Kubernetes-Native Platform for AI Agent Workloads



AgentCube: A Kubernetes-Native Platform for AI Agent Workloads



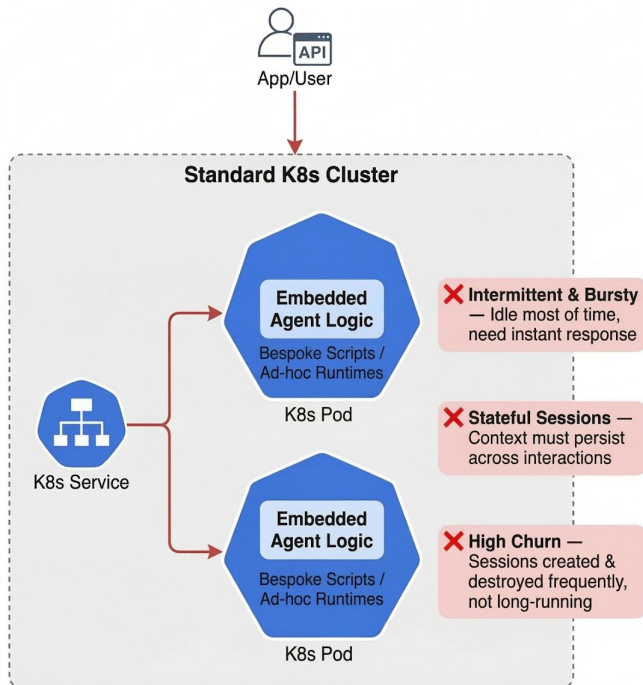
KubeCon



CloudNativeCon

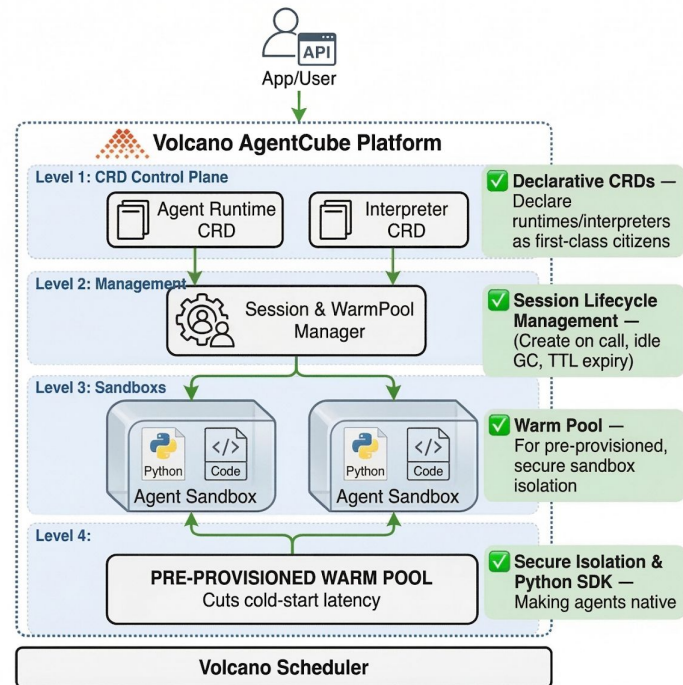
Europe 2026

Status Quo: Kubernetes Today - THE PROBLEM



COMPARISON: AGENT WORKLOADS ON KUBERNETES

Volcano AgentCube: The Solution



AgentCube: A Kubernetes-Native Platform for AI Agent Workloads

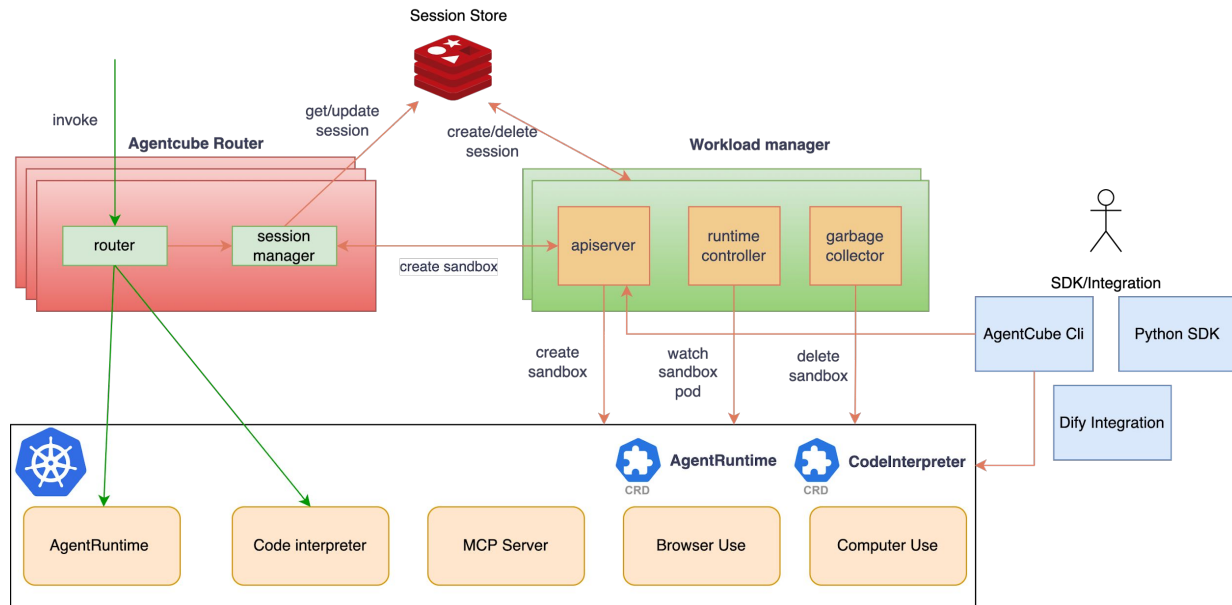


KubeCon



CloudNativeCon

Europe 2026



- **Session-aware** routing and lifecycle management
- Warm pools for **reusable, fast-starting sandboxes**
- **Out-of-the-box** SDKs, CLI, and extensible integrations

AgentCube: Out-of-the-Box SDK



KubeCon



CloudNativeCon

Europe 2026

```
from agentcube import CodeInterpreterClient

# Step 1: create a new sandboxed session
client1 = CodeInterpreterClient()

# Persist state in the workspace
client1.write_file("42", "value.txt")

# Step 2: reuse the same session in a new client
client2 = CodeInterpreterClient(session_id=client1.session_id)

# The file written in step 1 is still available
result = client2.run_code(
    "python",
    """
print("Reusing existing session...")
print(open("value.txt").read())
"""
)

print(result)

# Cleanup when done
client2.stop()
```

- Getting Started:

<https://github.com/volcano-sh/agentcube/blob/main/docs/getting-started.md>

- Python SDK Example:

https://github.com/volcano-sh/agentcube/blob/main/sdk-python/examples/basic_usage.py



KubeCon



CloudNativeCon

Europe 2026

Volcano Global: What's new?



Volcano Global: What's new?

- Feature 1: HyperJob

Multi-cluster Job Splitting for Large-scale AI Training

- Feature 2: Data Dependency Aware Scheduling

Data-Aware Placement for Compute-Storage Disaggregation

1. HyperJob for Multi-cluster Job Splitting

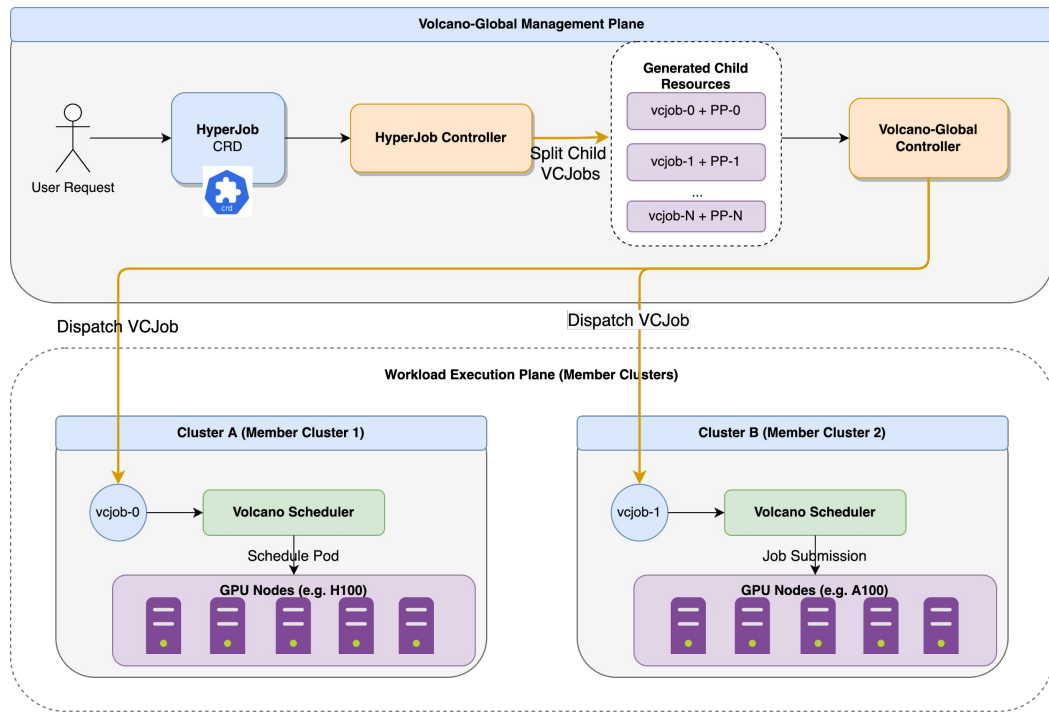


KubeCon



CloudNativeCon

Europe 2026



- **Break Scale Limits:** Train massive LLMs requiring thousands of GPUs that exceed a single cluster's capacity.
- **Embrace Heterogeneity:** Seamlessly split and coordinate training tasks across clusters with different hardware (e.g., A100 in Cluster A, Ascend in Cluster B).
- **Unified Experience:** Submit one job, get one aggregated status. No manual distribution or cross-cluster coordination required.

HyperJob Use Case



KubeCon



CloudNativeCon

Europe 2026

```
apiVersion: training.volcano.sh/v1alpha1
kind: HyperJob
metadata:
  name: llm-training
spec:
  minAvailable: 2
  maxDomains: 2
  replicatedJobs:
  - name: trainer
    replicas: 2
    templateSpec:
      tasks:
      - name: worker
        replicas: 128
        template:
          spec:
            containers:
            - name: trainer
              image: training-image:v1
              resources:
                requests:
                  nvidia.com/gpu: 1
```

Case 1: Large-scale Training Job Splitting

```
apiVersion: training.volcano.sh/v1alpha1
kind: HyperJob
metadata:
  name: ascend-heterogeneous-training
spec:
  minAvailable: 2
  replicatedJobs:
  - name: trainer-910b
    replicas: 1
    clusterNames: ["cluster-ascend-910b-1", "cluster-ascend-910b-2"]
    templateSpec:
      tasks:
      - name: worker
        replicas: 64
        template:
          spec:
            affinity:
              nodeAffinity:
                requiredDuringSchedulingIgnoredDuringExecution:
                  nodeSelectorTerms:
                  - matchExpressions:
                    - key: hardware-type
                      operator: In
                      values:
                      - Ascend910B
            containers:
            - name: trainer
              image: training-image:v1
              resources:
                requests:
                  ascend910c: 1
                limits:
                  ascend910c: 1
  - name: trainer-910c
    replicas: 1
    clusterNames: ["cluster-ascend-910c-1"]
    templateSpec:
      tasks:
      - name: worker
        replicas: 64
        template:
          spec:
            affinity:
              nodeAffinity:
                requiredDuringSchedulingIgnoredDuringExecution:
                  nodeSelectorTerms:
                  - matchExpressions:
                    - key: hardware-type
                      operator: In
                      values:
                      - Ascend910C
            containers:
            - name: trainer
              image: training-image:v1
              resources:
                requests:
                  ascend910c: 1
                limits:
                  ascend910c: 1
```

Case 2: Heterogeneous Clusters

2. Data Dependency Aware Scheduling

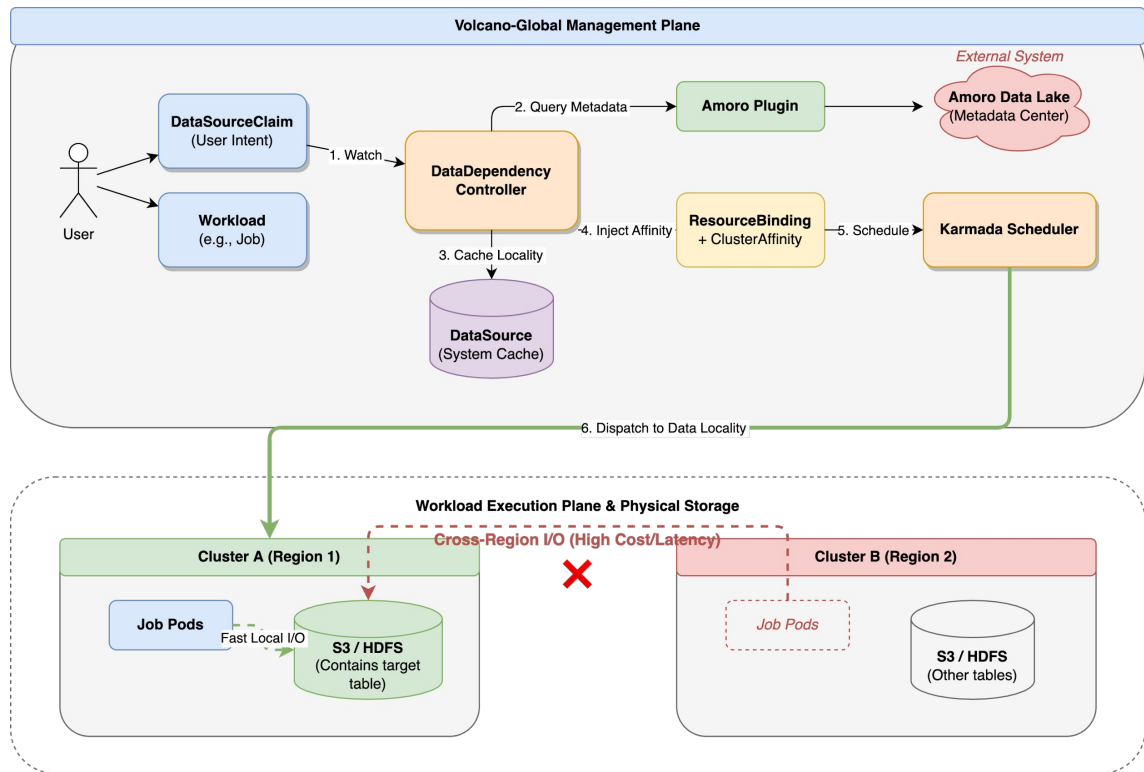


KubeCon



CloudNativeCon

Europe 2026



- **Pluggable Integration:** Connects with external Data Lakes (e.g., Amoro, Hive, S3) to discover physical data locality.
- **Zero-Touch Placement:** Automatically injects affinity rules to force workloads to the cluster closest to the target data.

Resources and Call for Contribution

Call For Use Case:

<https://github.com/volcano-sh/volcano/issues/3855>

Are you using Volcano?

First of all, thank you very much for using and supporting Volcano!
We highly appreciate every user's feedback, we will try our best to keep Volcano better.

The intention of this issue:

We'd like to establish a connection between the community and users, to listen to their voices, and to provide support when needed.

Call For Contribution:

<https://github.com/volcano-sh/volcano/blob/master/contribute.md>





KubeCon



CloudNativeCon

Europe 2026

Q&A





KubeCon



CloudNativeCon

Europe 2026

Thank you!

