



Europe 2025





KubeCon



CloudNativeCon

Europe 2025

Optimizing Training Performance for LLM in Kubernetes

Klaus Ma
Peng Gu



Who We Are



Klaus Ma

NVIDIA

Senior Software Manager
Beijing, China



Peng Gu


Tech Startup

Software Architect

A distributed system helps us to manage the workload to match the design performance of hardware.

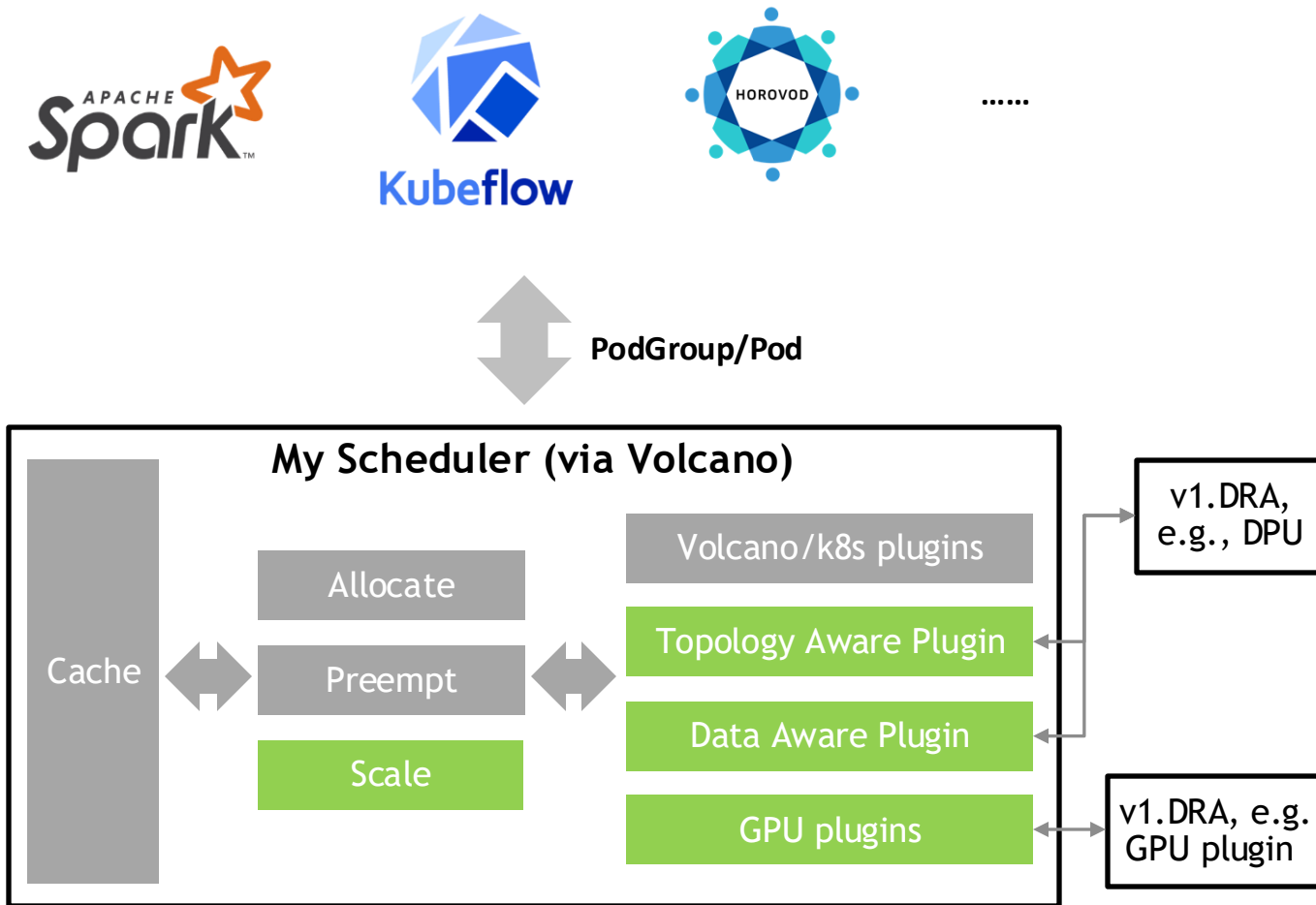
- **Computing** resources, e.g., CPU, GPU, DPU and other accelerators
- **Networking** resources, e.g., Eth/IB/RoCE, bandwidth, health, latency
- **Storage** resources, e.g., cache, local disk, remote/distributed storage

Cloud native batch scheduling system for compute-intensive workloads

- **Computing** resources: DRF, fair-sharing, Queue, SLA, backfill, preempting, reclaiming,


We're here now!
- **Networking** resources, e.g., Task-Topology Scheduling, **Networking Topology Aware Scheduling**
- **Storage** resources, e.g., Data aware scheduling, Data pre-loading

Volcano Plugin Machinery



```
1 package main
2
3 import (
4     ...
5
6     "volcano.sh/volcano/cmd/scheduler/app"
7
8     // Import default actions/plugins.
9     _ "volcano.sh/volcano/pkg/scheduler/actions"
10    _ "volcano.sh/volcano/pkg/scheduler/plugins"
11
12    // Import my-scheduler actions/plugins
13    _ "xxx.com/my-scheduler/pkg/volcano/actions"
14    _ "xxx.com/my-scheduler/pkg/volcano/plugins"
15 )
16
17 func main() {
18     ...
19     if err := app.Run(s); err != nil {
20         fmt.Fprintf(os.Stderr, "%v\n", err)
21         os.Exit(1)
22     }
23 }
```

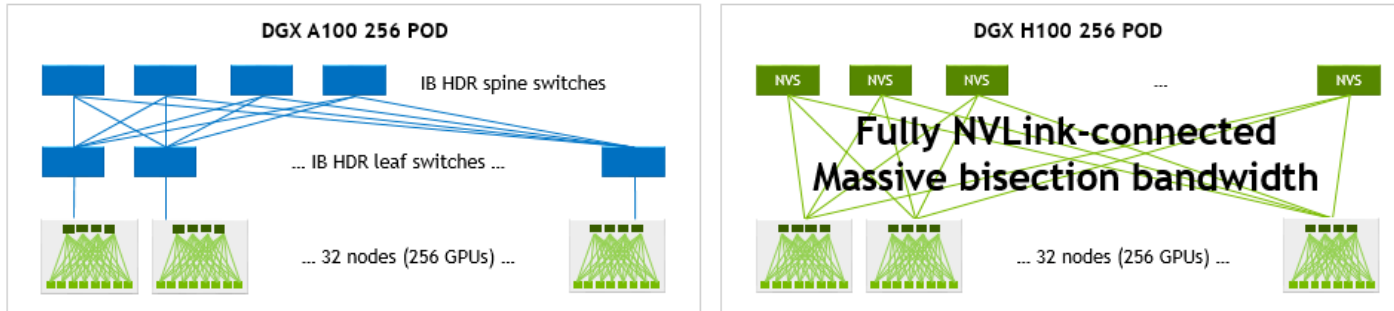
Challenges

- High GPU density
- High Bandwidth & Low latency

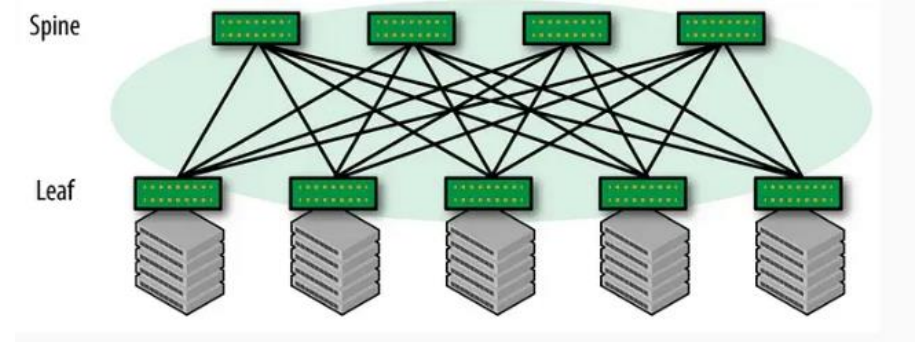
Volcano

- **Computing** resources: DRF, fair-sharing, Queue, SLA, backfill, preempting, reclaiming,
- **Networking** resources, e.g., Task-Topology Scheduling, **Networking Topology Aware Scheduling**
- **Storage** resources, e.g., Data aware scheduling, Data pre-loading

Backgrounds of Networking Aware Scheduling



NVIDIA Super Pod

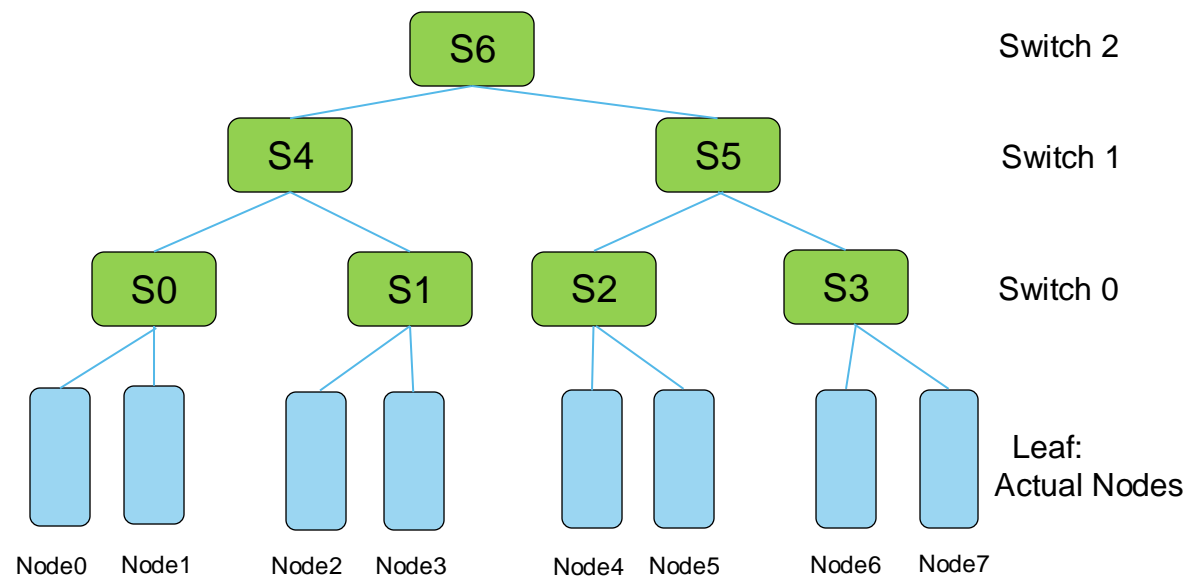


DIY Spine-leaf Network

- **Heterogeneous Data Center Network Architectures**
Protocol and Topology Diversity from IB to RoCE, NVLink
- **Inter-Node Communication in Multi-Tier Networks**
Performance Varies Across Network Hops

- **Affinity Scheduling for Compute-Intensive Workloads:**
Affinity scheduling optimizes placement for topology-sensitive workloads, e.g. model parallelism, disaggregated storage/compute
- **Topology-Agnostic Scheduling:**
The Hidden Cost of Generic Schedulers

HyperNode: Standardizing Cluster Topologies via Performance Domains



- **Leaf HyperNodes** (s0, s1, s2, s3): Whose members are actual cluster nodes.
- **Non-Leaf HyperNodes** (s4, s5, s6): Represent other HyperNodes.
- **node0** and **node1** within s0 have the highest communication efficiency.
- **node1** and **node2** spanning two HyperNode layers (s0→s4→s1) have lower efficiency.
- **node0** and **node4** spanning three HyperNode layers (s0→s4→s6) have the lowest efficiency.

Feature Preview in v1.11.0-network-topology-preview.0

HyperNode example

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

```
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3
  members:
  - type: HyperNode
    selector:
      exactMatch:
        name: s4
```

Volcano Job with network Topology

```
...
spec:
  networkTopology:
    mode: hard
    highestTierAllowed: 2
...
```

Pipelines for Networking scheduling

- Unified API for NVLink/IB/RoCE via Restful API and OpenAPI
- Device or Service auto-discovery, e.g., Nvidia DPF (HBN/OVN for networking, SNAP for storage)
- Topology/Healthy auto-discovery
- Bandwidth/Latency auto-discovery

```
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3
  members:
  - type: HyperNode
    selector:
      exactMatch:
        name: s4
```



```
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3
  members:
  - type: HyperNode
    selector:
      exactMatch:
        name: s4
status:
  conditions:
    type: Ready
    value: true
    timestamp: 2025-03-01 1:11:11
```

Evolution of Network Topology Awareness in Kubernetes



Standard topology labels introduced

Volcano Topology Aware Scheduling
Feature preview



PodTopologySpread feature released

Why we choose Volcano Scheduler

- ✓ **Gang Scheduling**
- ✓ **Heterogeneous Device Scheduling**
- ✓ **Task Topology Support**
- ✓ **Flexible Resource Sharing/Preemption/Reclaim**

A Typical GB300 Cluster Backend Network Topology

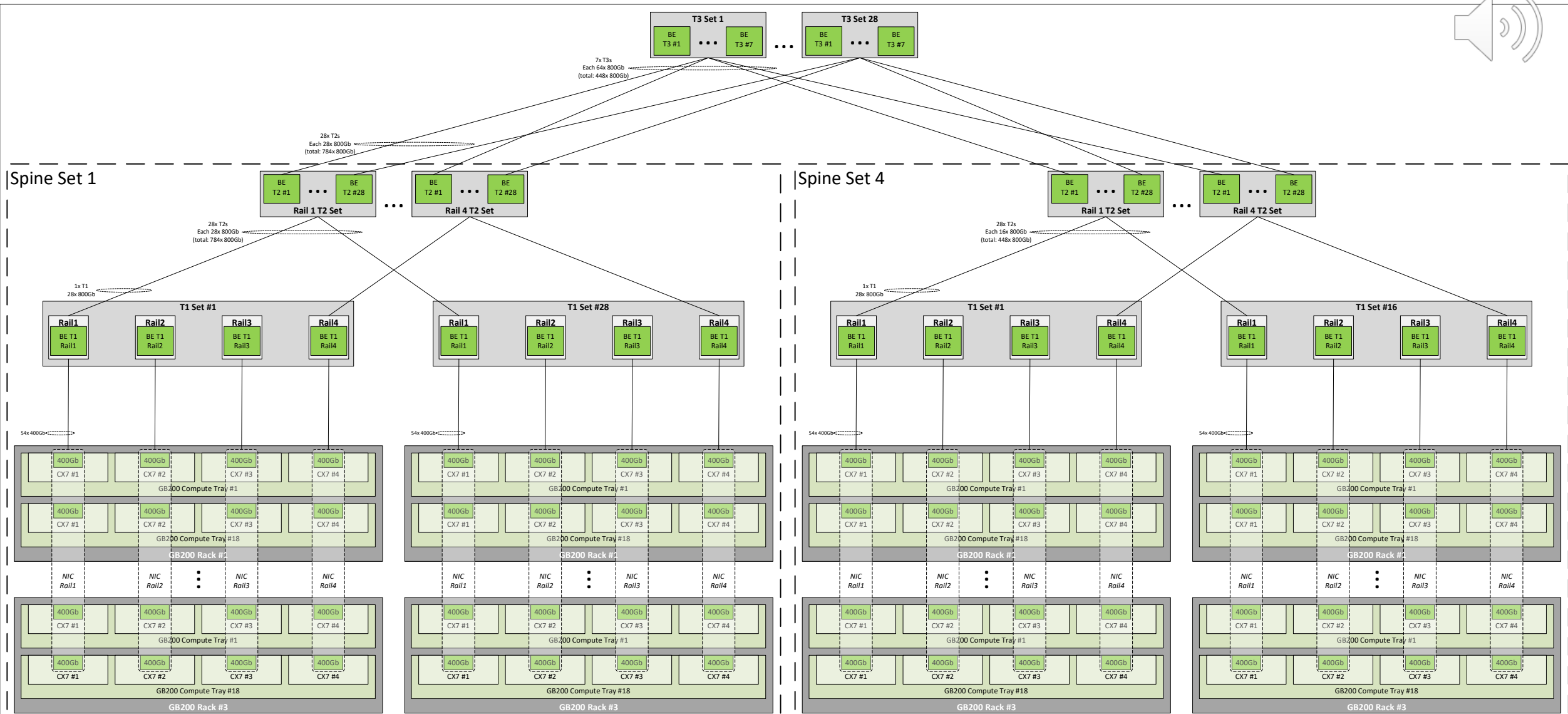


KubeCon



CloudNativeCon

Europe 2025



Logical HyperNode Topology

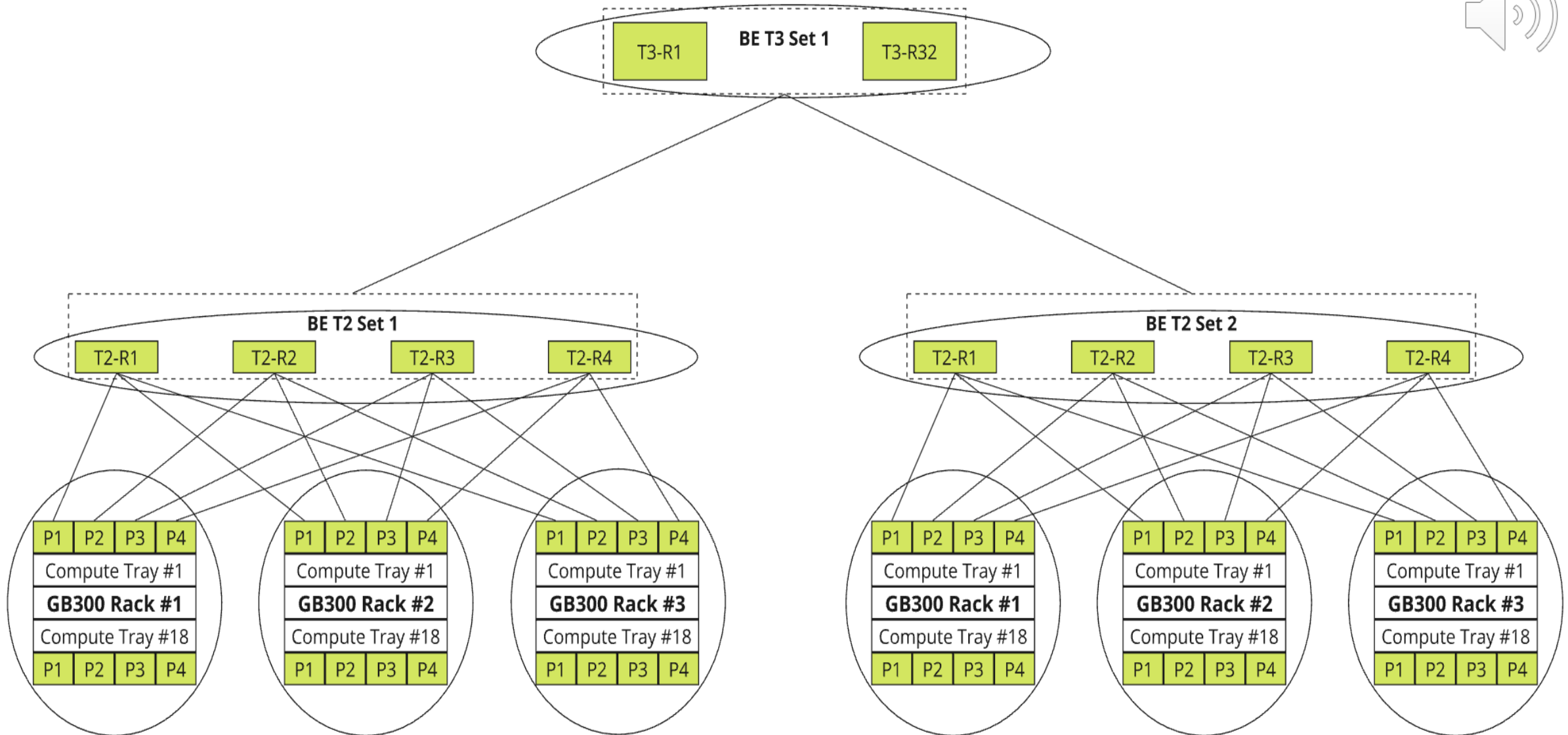


KubeCon



CloudNativeCon

Europe 2025



Demo Setup – Software Versions

- ✓ K8s: v1.32.2
- ✓ Kind: v0.27.0
- ✓ KWOK: v0.6.1
- ✓ Volcano: v1.11.0-network-topology-preview.0



DEMO

Cluster Topology

Job Submission

Scheduling Observation

- 1. Install Volcano v1.11.0 or later:** This version includes the Network Topology-Aware Scheduling feature.
- 2. Define HyperNodes:** Create HyperNode CRs to represent your cluster's network topology.
- 3. Configure Jobs with Network Topology Constraints:** Specify the desired network topology constraints in your job definitions to guide the scheduler's placement decisions.

```
helm install volcano volcano-sh/volcano \  
--version v1.11.0-network-topology-preview.0 -n volcano-system --create-namespace
```

HyperNode CRD Example

```
conf > hypernode > ! hypernode.yaml > ...
1  # HyperNode Custom Resources for Network Topology
2
3  # Tier 1: Rack level HyperNodes
4  ---
5  apiVersion: topology.volcano.sh/v1alpha1
6  kind: HyperNode
7  metadata:
8    name: rack-uk-london1-dc01-row01-rack01
9  spec:
10    tier: 1
11    members:
12    - type: Node
13      selector:
14        regexMatch:
15          pattern: "gb300-node-uk-london1-dc01-row01-rack01-node[0-9]+"
```

Job YAML Example



KubeCon



CloudNativeCon

Europe 2025

```
conf > jobs > ! llm-training-1.yaml > {} spec > {} networkTopology
```

```
1  apiVersion: batch.volcano.sh/v1alpha1
2  kind: Job
3  metadata:
4    | name: llm-training-1
5  spec:
6    | networkTopology:
7      | mode: hard
8      | highestTierAllowed: 1
9    minAvailable: 18
10   schedulerName: volcano
```

- **External Network Topology Discovery:** Volcano does not include an in-band network topology discovery service. Each cloud provider or cluster administrator needs to implement their own method for defining and maintaining the network topology information.
- **Accuracy of Topology Information:** The effectiveness of scheduling depends on the accuracy of the HyperNode definitions.
- **Dynamic Cluster Environments:** In rapidly changing clusters, maintaining up-to-date topology information can be challenging.
- **Still in Early Preview:** This feature will mature overtime, but today it is still in Preview stage.

- **Add External Network Topology Discovery:** For cloud providers, to add cloud provider specific network auto discovery features.
- **Network Topology Aware Scheduling in Production:** We plan to upgrade to the latest Volcano version when this feature reaches GA.
- **Scheduler Performance Tuning/Validation:** We did not get a chance to test the performance of the scheduler as of today. We plan to do a scheduling performance analysis and workload performance analysis as well.

- Peng Gu
 - LinkedIn: <https://www.linkedin.com/in/peng-g-9ab93211/>
- Klaus Ma:
 - LinkedIn: <https://www.linkedin.com/in/k82cn/>
 - Github: <http://github.com/k82cn>

Thank you



KubeCon



CloudNativeCon

Europe 2025

Q & A