KUBERNETES COMMUNITY DAYS BEIJING 2025

面向LLM的高效计算：基于昇腾硬件和Volcano的软硬协同优化

Shuqiao Li (Huawei, Senior Engineer)

Zicong Chen(Huawei Cloud, Member of Volcano, R&D Engineer)

Content

目录

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 00
背景介绍

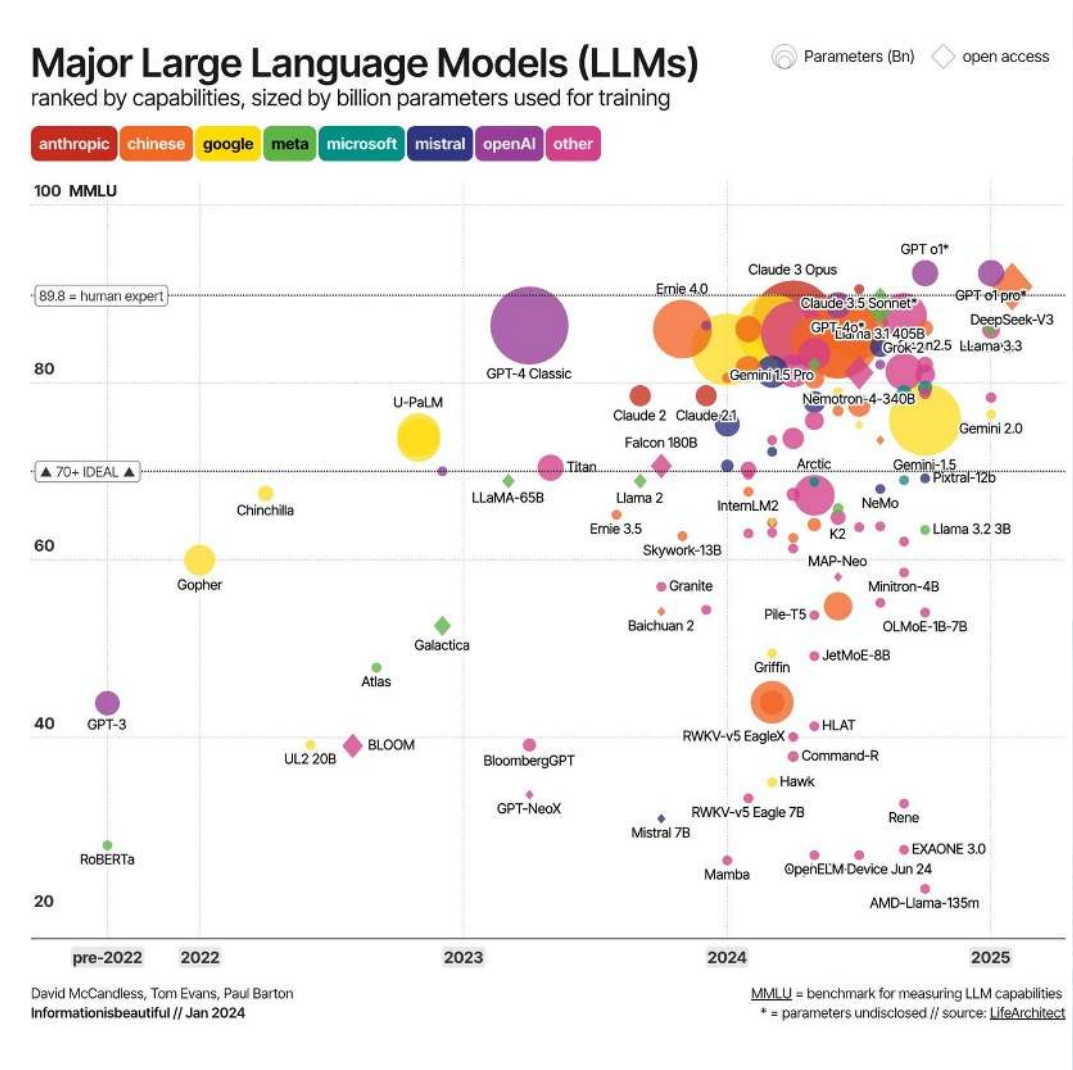
LLM发展趋势

大模型层出不穷：
ChatGPT、Claude、Gemini、Qwen、DeepSeek等

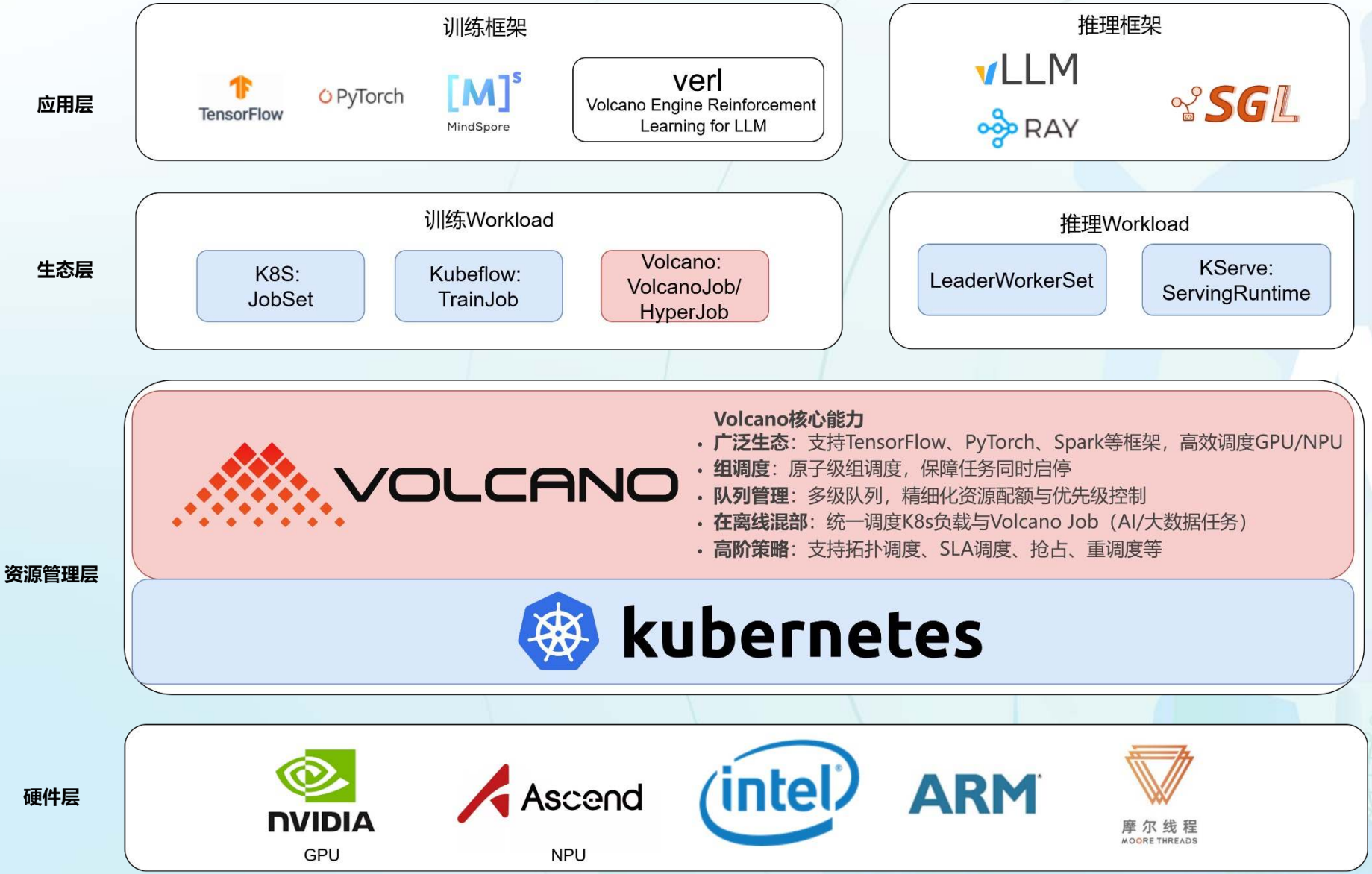
参数规模持续增长：
DeepSeek达671B，GPT-4超万亿

计算需求激增:
单机单卡已无法满足大模型的训练/推理需求，分布式训练和推理中存在的数据并行、模型并行、流水线并行、专家并行、Prefill与Decode分离等技术已成为关键

MaaS商业化困境与性能博弈：
模型即服务（MaaS）模式面临盈利难题，但行业竞争迫使企业持续投入。模型性能直接影响用户体验和运营成本，优化计算效率成为关键。



LLM全栈架构：分层协同赋能高效训练与推理



Volcano AI生态

北向AI框架支持：

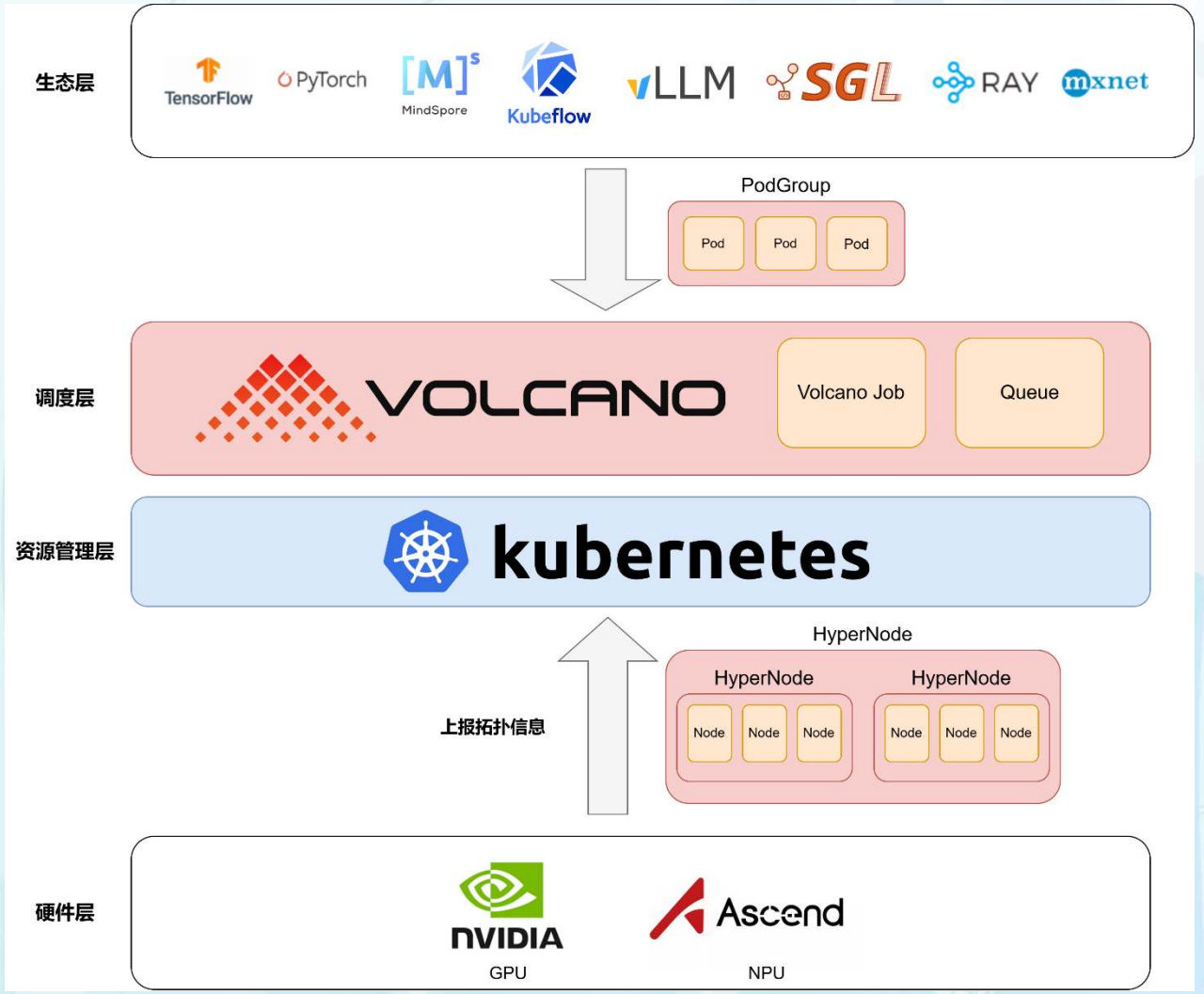
• 支持TensorFlow、 PyTorch等主流训练框架，支持vLLM等主流推理框架。生态层框架可通过Volcano PodGroup实现Gang Scheduling，并结合Volcano Queue进行精细化资源配额控制

南向硬件支持：

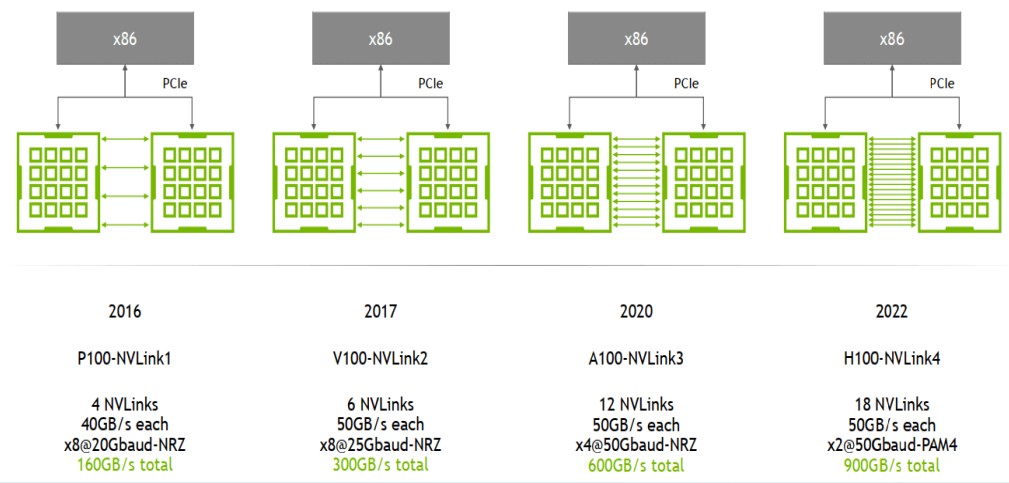
• vGPU/MIG调度

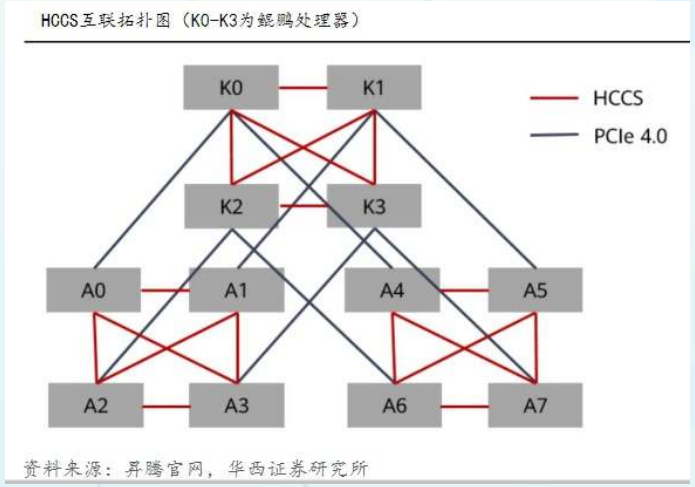
• 昇腾NPU调度

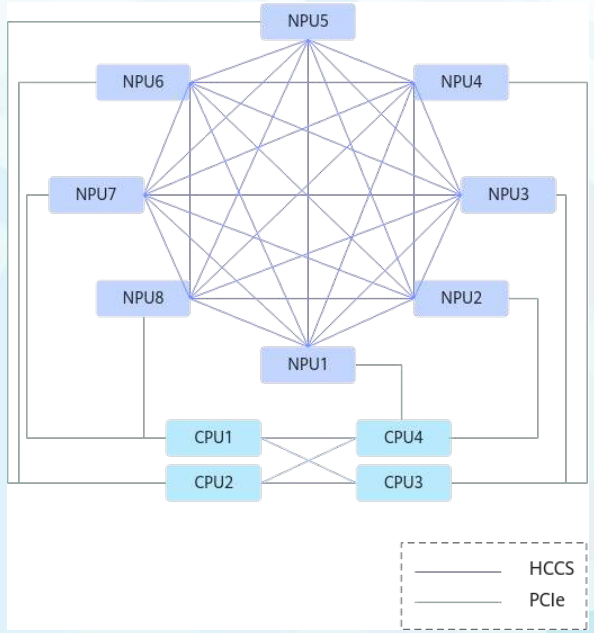
• 支持使用网络拓扑信息发现工具，提供HyperNode CRD，提升训练/推理任务网络通信效率



昇腾NPU/GPU节点内拓扑



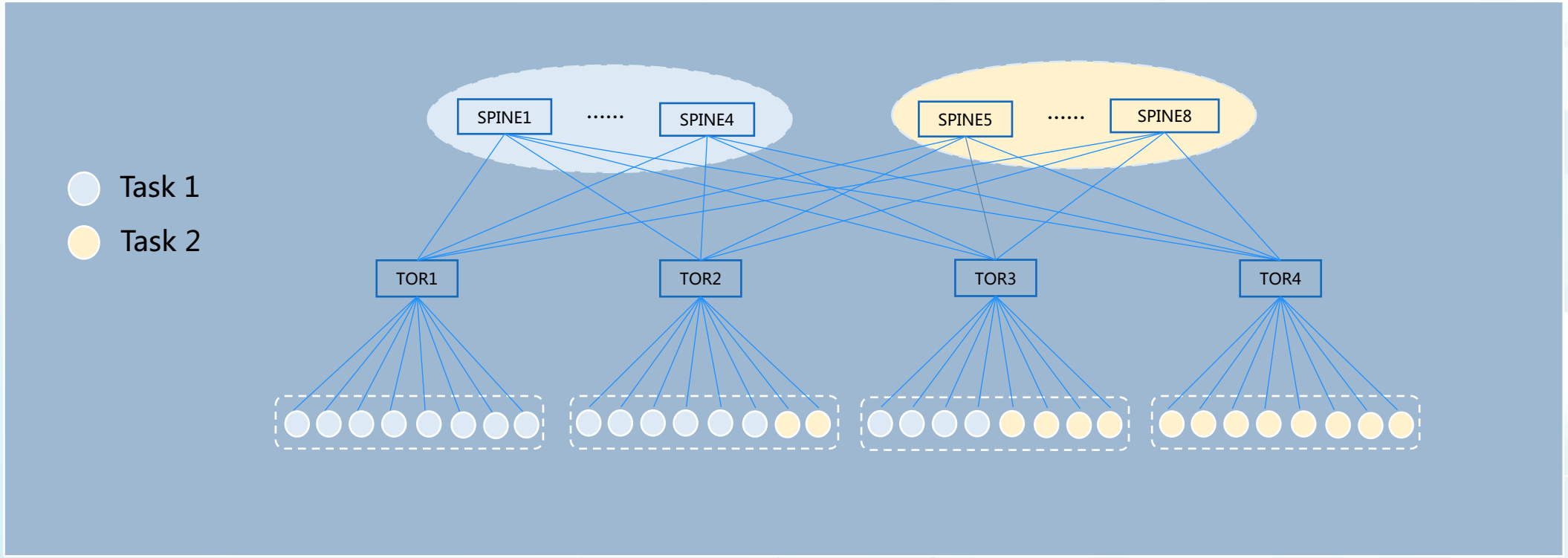




需要支持：

- 昇腾NPU亲和性调度
- HCCS拓扑感知调度
- GPU拓扑感知调度
- 拓扑感知抢占

跨节点网络拓扑感知缺失



当前现状：

- **跨节点网络拓扑感知缺失**：

调度器无法识别网络拓扑中的高效通信区域（如同一机架内），频繁通信的任务组（PodGroup）可能被分散到不同机架或节点，可能导致任务间的数据交换路径过长，增加延迟，拖慢训练/推理效率。

应用层框架与底层硬件的现状与挑战

推理框架

训练框架

verl

Volcano Engine Reinforcement Learning for LLM

当前，有一部分的上层应用层框架（如推理框架和训练框架）仍无法完全对底层硬件无感，导致硬件性能无法充分发挥。

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 01

节点内拓扑感知调度

节点内昇腾NPU亲和性调度

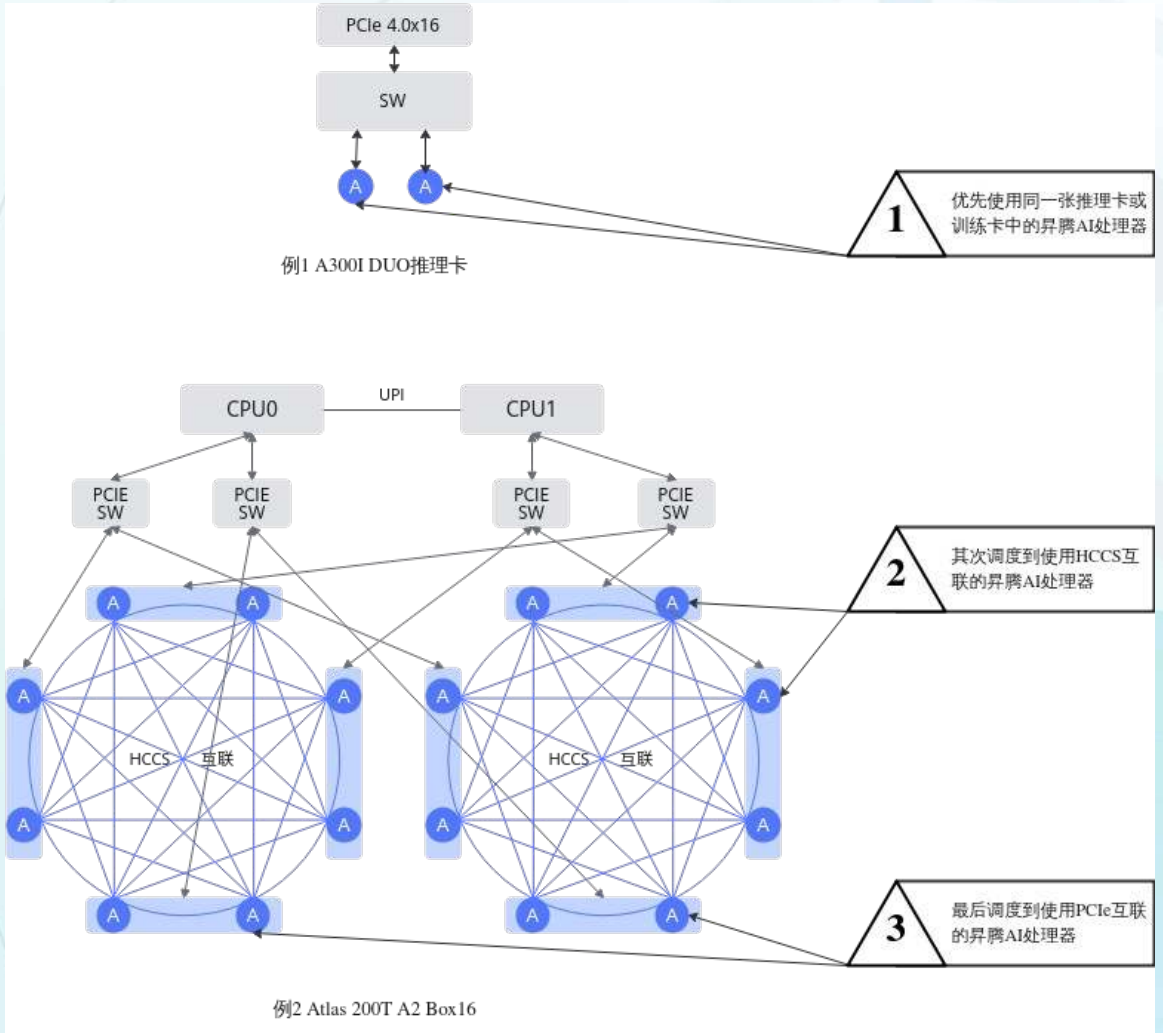
在昇腾硬件产品内部，有三种芯片链接方式。他们的调度优先级为：

① 优先将任务调度到同一张推理卡或者训练卡内的昇腾AI处理器中；

② 其次调度到使用HCCS互联的昇腾AI处理器中；

③ 最后调度到使用PCIe互联的昇腾AI处理器中。

HCCS（Huawei Cache Coherence System）是HCCL（Huawei Collective Communication Library）的硬件形态，HCCL提供了深度学习训练场景中服务器间高性能集合通信的功能。



昇腾AI处理器互联方式

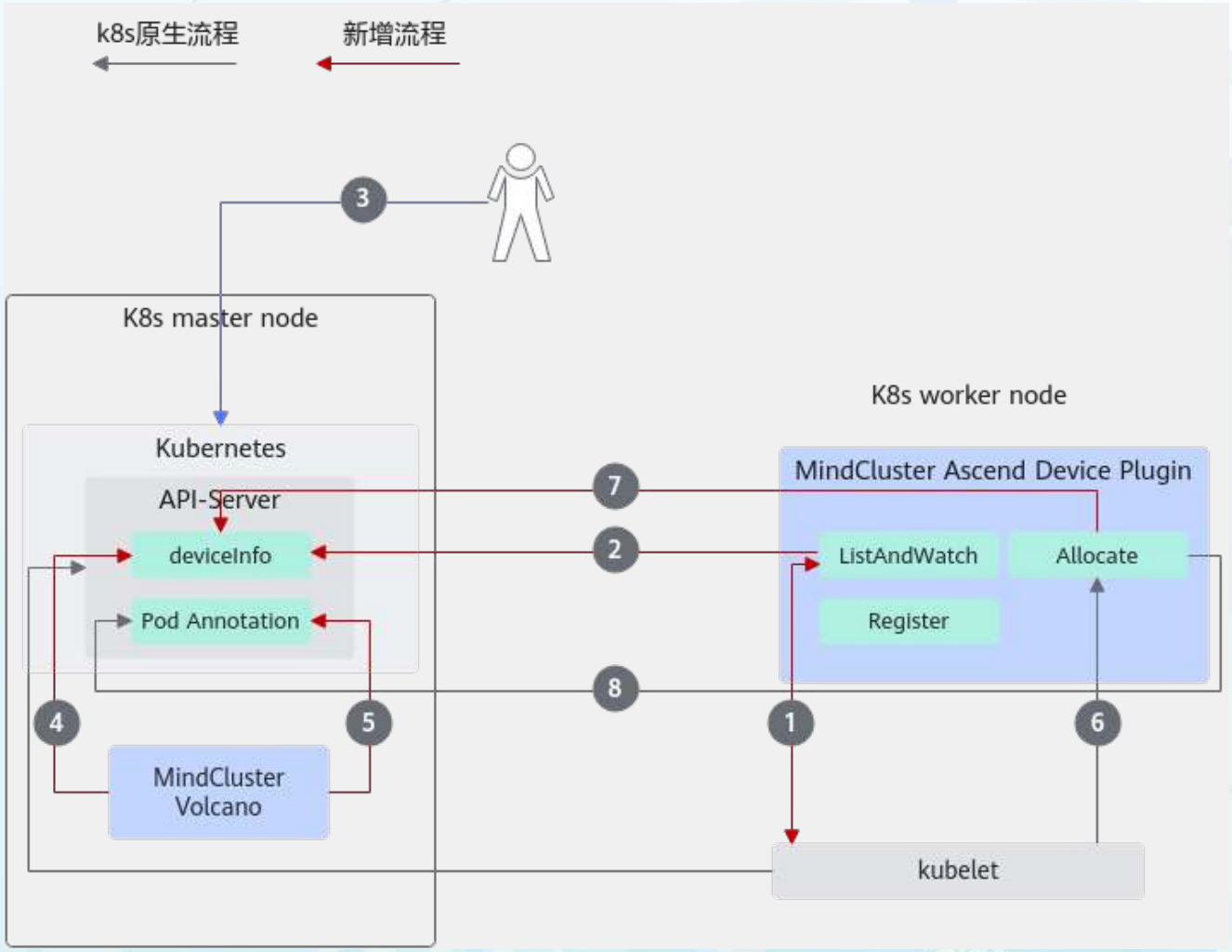
昇腾NPU调度流程

①② NPU Device Plugin组件上报NPU健康状态和拓扑信息，更新到configMap deviceinfo-{*nodeName*}中。

③ 用户创建业务job。

④ Volcano组件通过configmap获取当前可用的NPU

NPU调度流程详情见：
[昇腾AI处理器的调度流程]



NPU调度流程

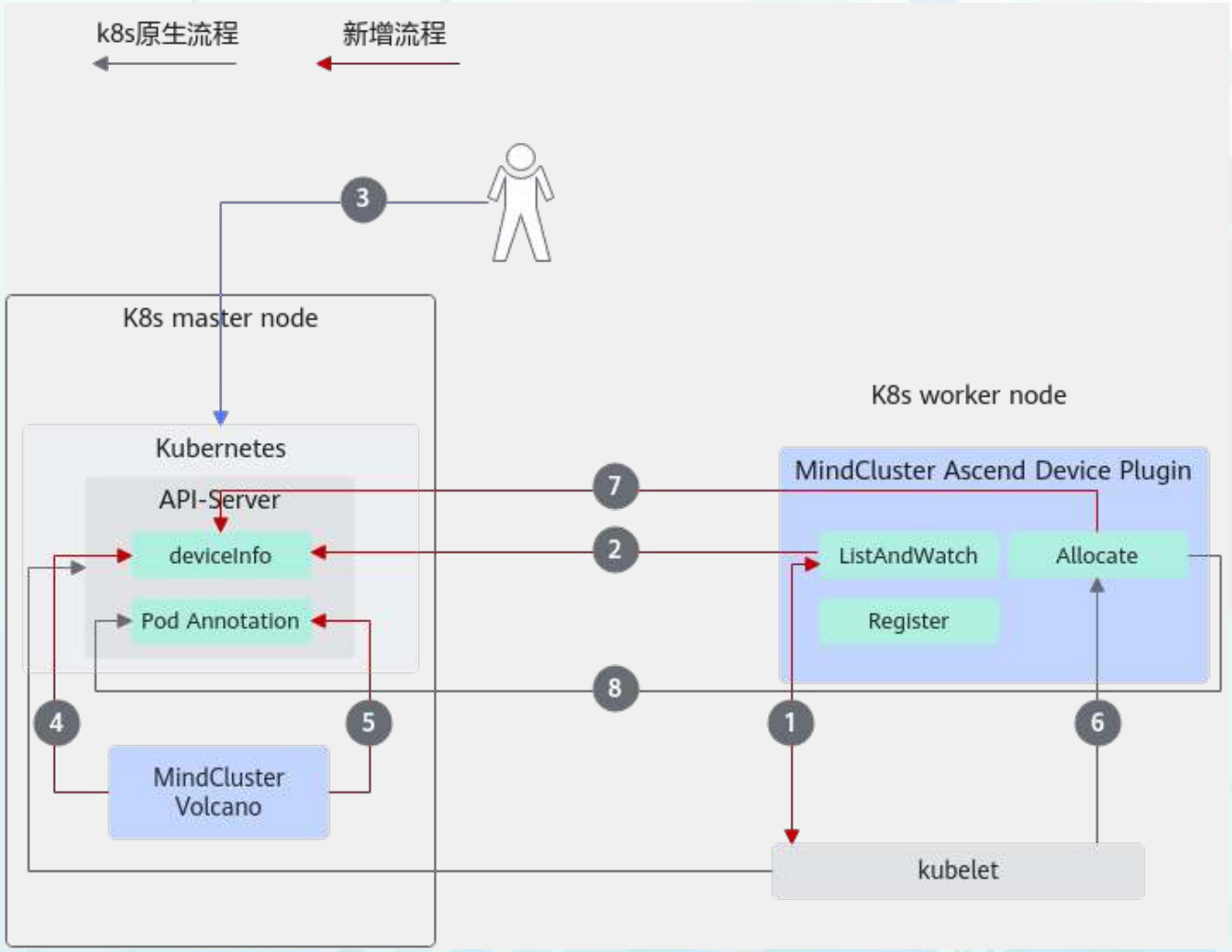
昇腾NPU调度流程

⑤Volcano根据亲和性调度原则，将NPU分配的结果写入Pod的Annotations字段中，随后把Pod Bind到节点

⑥⑦kubelet监测到有Pod调度到自己所在节点，挂载NPU设备。

⑧ NPU Device Plugin更新configmap中的NPU分配情况。

NPU调度流程详情见：
[昇腾AI处理器的调度流程]



NPU调度流程

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 02

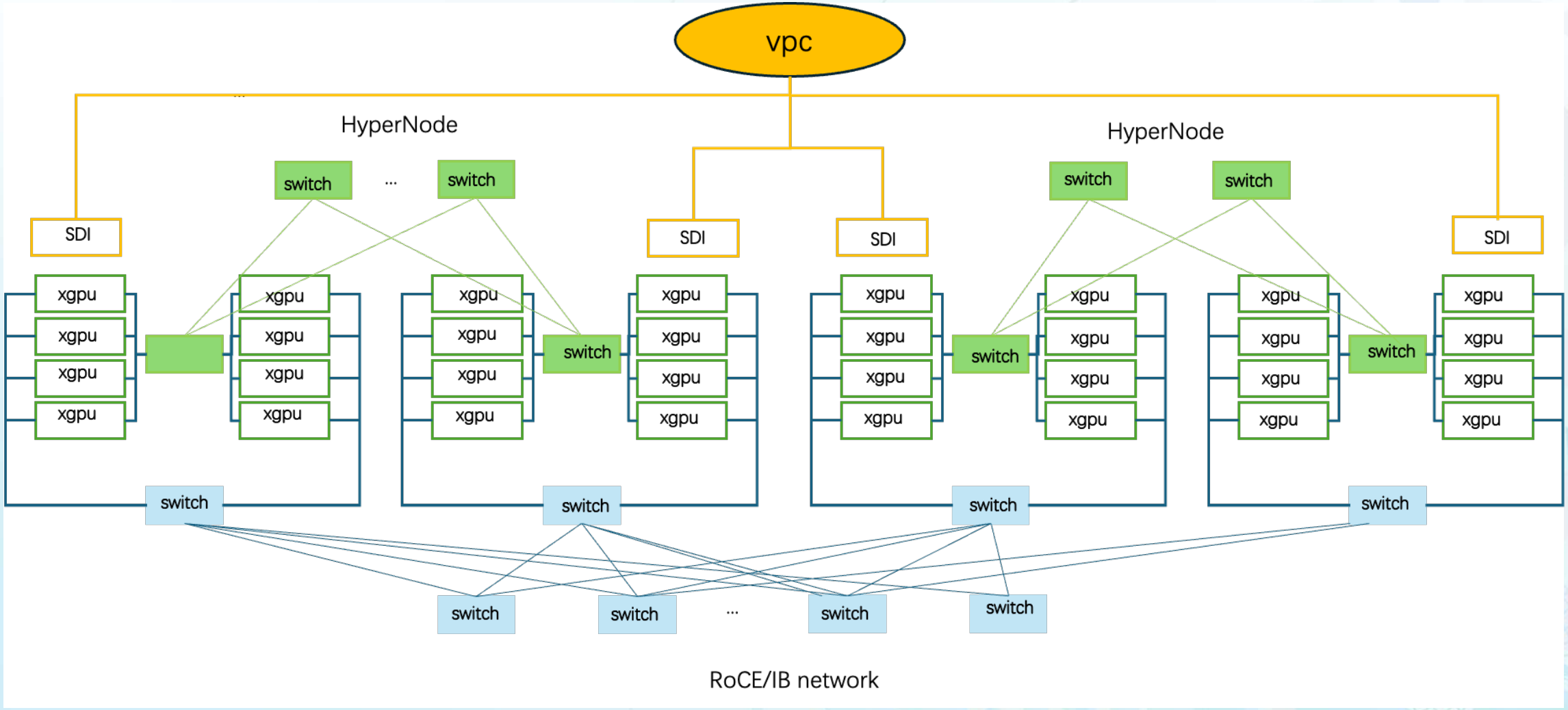
跨节点网络拓扑感知调度

技术背景

在AI大模型训练与推理场景中，**超节点架构**通过整合多个计算节点，为用户提供高效、可扩展的计算能力，已成为行业主流趋势。然而，随着各家厂商纷纷构建自有超节点方案，统一的资源管理和调度方案缺失问题日益凸显。特别是在**模型并行**技术下，模型被拆分到多个计算节点上，导致训练/推理过程中节点间需频繁交换海量数据（如梯度、参数等）。此时，**网络传输性能直接决定整体效率**，跨节点通信成为关键瓶颈。当前存在以下挑战和需求：

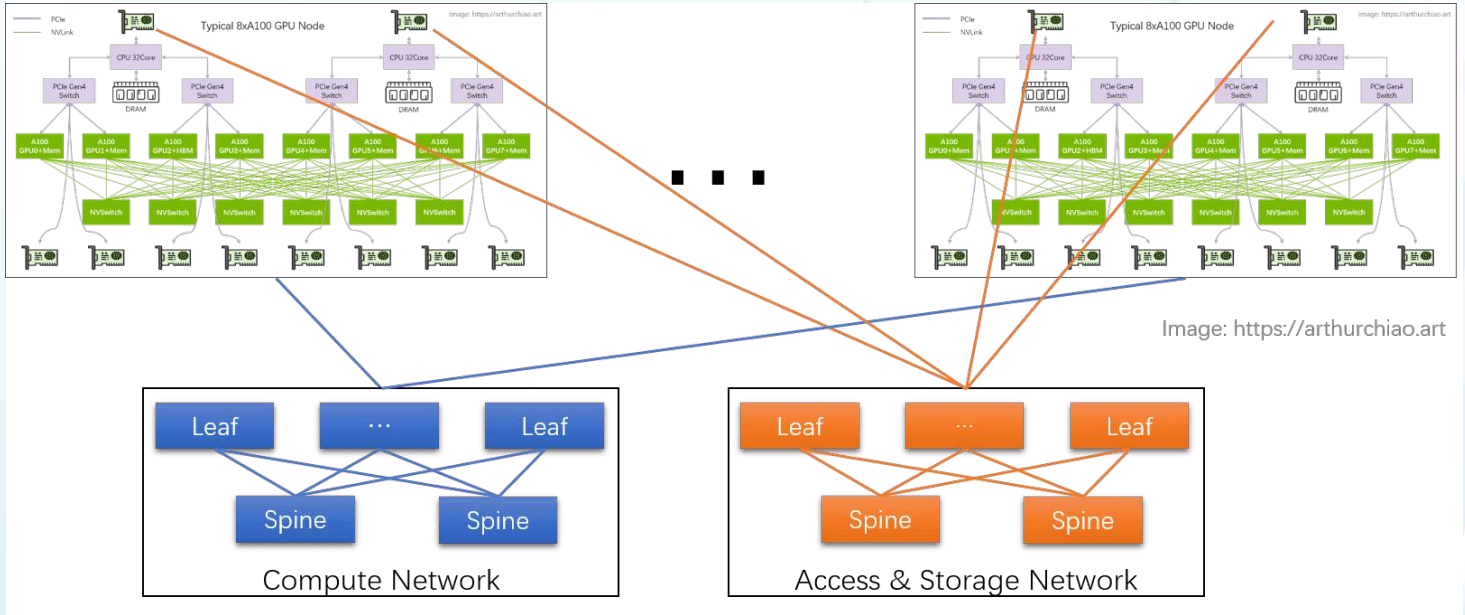
- 数据中心**网络类型多样**（如InfiniBand、RoCE、NVSwitch），**拓扑复杂**（多层交换机堆叠）。
- 通信路径中**跨越的交换机层级越多，延迟越高、吞吐量越低**。
- 通用调度器**缺乏对网络拓扑的感知能力**，无法根据交换机层级、带宽和延迟差异进行智能调度。
- **人工指定节点亲和性效率低**，难以动态适配复杂网络环境。

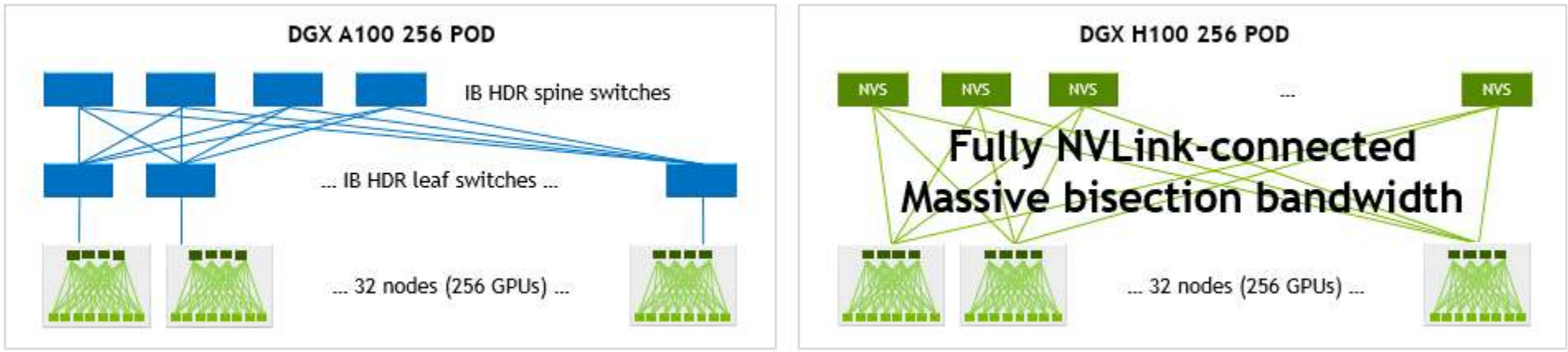
NPU超节点集群组网



NPU集群组网

GPU超节点集群组网





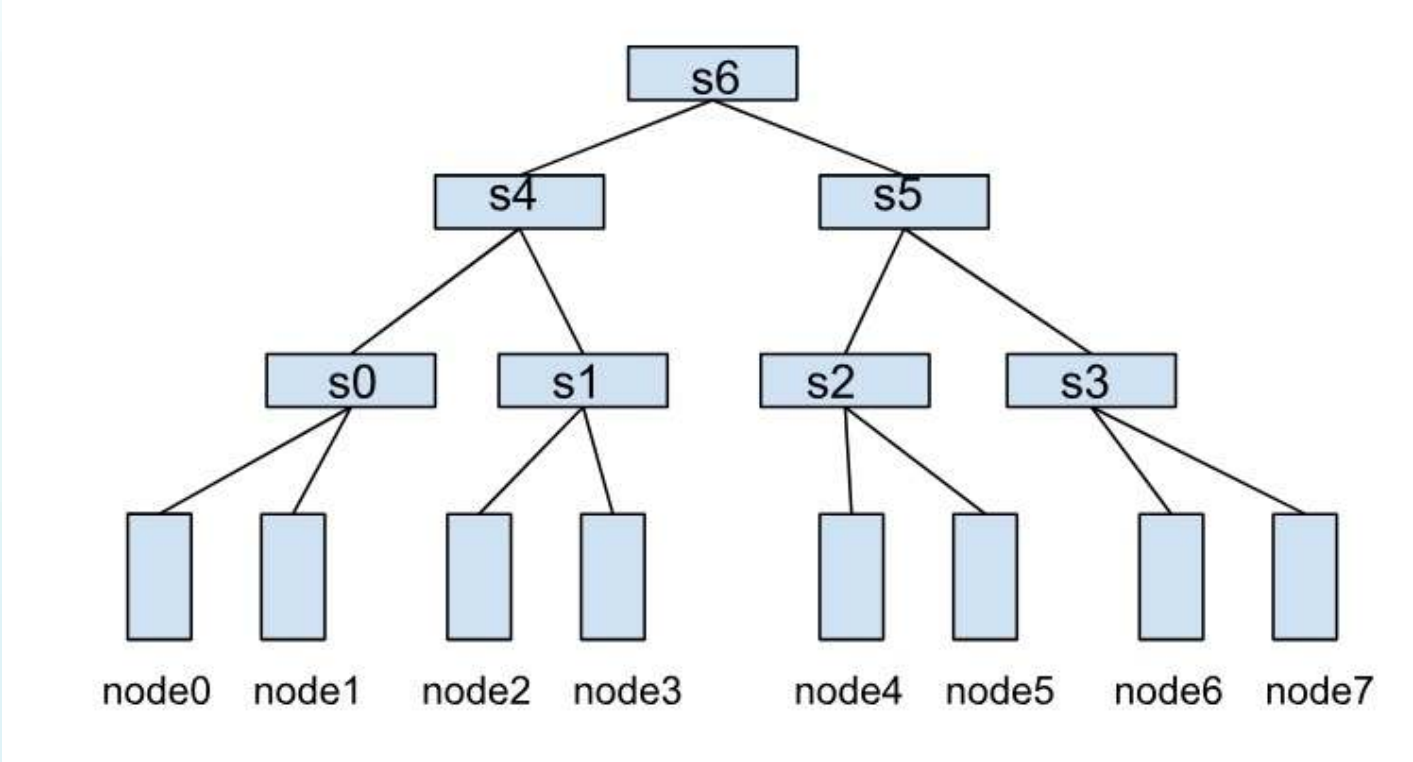
GPU集群组网

统一的网络拓扑API

Volcano定义了新的CRD **HyperNode**来表示网络拓扑，提供了标准化的API接口，**可屏蔽不同集群网络类型的差异，统一表示一个网络拓扑性能域。**

与传统的通过节点标签表示网络拓扑的方式相比，HyperNode具有以下优势：

- **语义统一**：HyperNode提供了标准化的网络拓扑描述方式，避免了标签方式的语义不一致问题。
- **层级结构**：HyperNode支持树状层级结构，能够更精确地表达实际的网络拓扑。
- **易于管理**：集群管理员可以手动创建HyperNode，或通过网络拓扑自动发现工具维护HyperNode。
- **统一的通信域健康状态管理**：可以统一承载通信域下的网络健康状态，节点健康状态。



层级网络拓扑示例

统一的网络拓扑API

```yaml
apiVersion: topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s0
spec:
  tier: 1  # HyperNode层级，层级越低通信效率越高
  members: # 子节点列表
  - type: Node  # 子节点类型为Node
    selector:
      exactMatch: # 精确匹配
        name: node-0
  - type: Node
    selector:
      regexMatch: # 正则匹配
        pattern: node-[01]
  - type: Node
    selector:
      labelMatch: # 标签匹配
        matchLabels:
          topology-rack: rack-1
```

叶子HyperNode示例yaml

```yaml
apiVersion:
topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3  # HyperNode层级
  members: # 子节点列表
  - type: HyperNode  # 子节点类型为
HyperNode
    selector:
      exactMatch: # 精确匹配
        name: s4
  - type: HyperNode
    selector:
      exactMatch:
        name: s5
```

非叶子HyperNode示例yaml

统一的网络拓扑API

Volcano Job和PodGroup可以通过networkTopology字段设置作业的拓扑约束，支持以下配置：

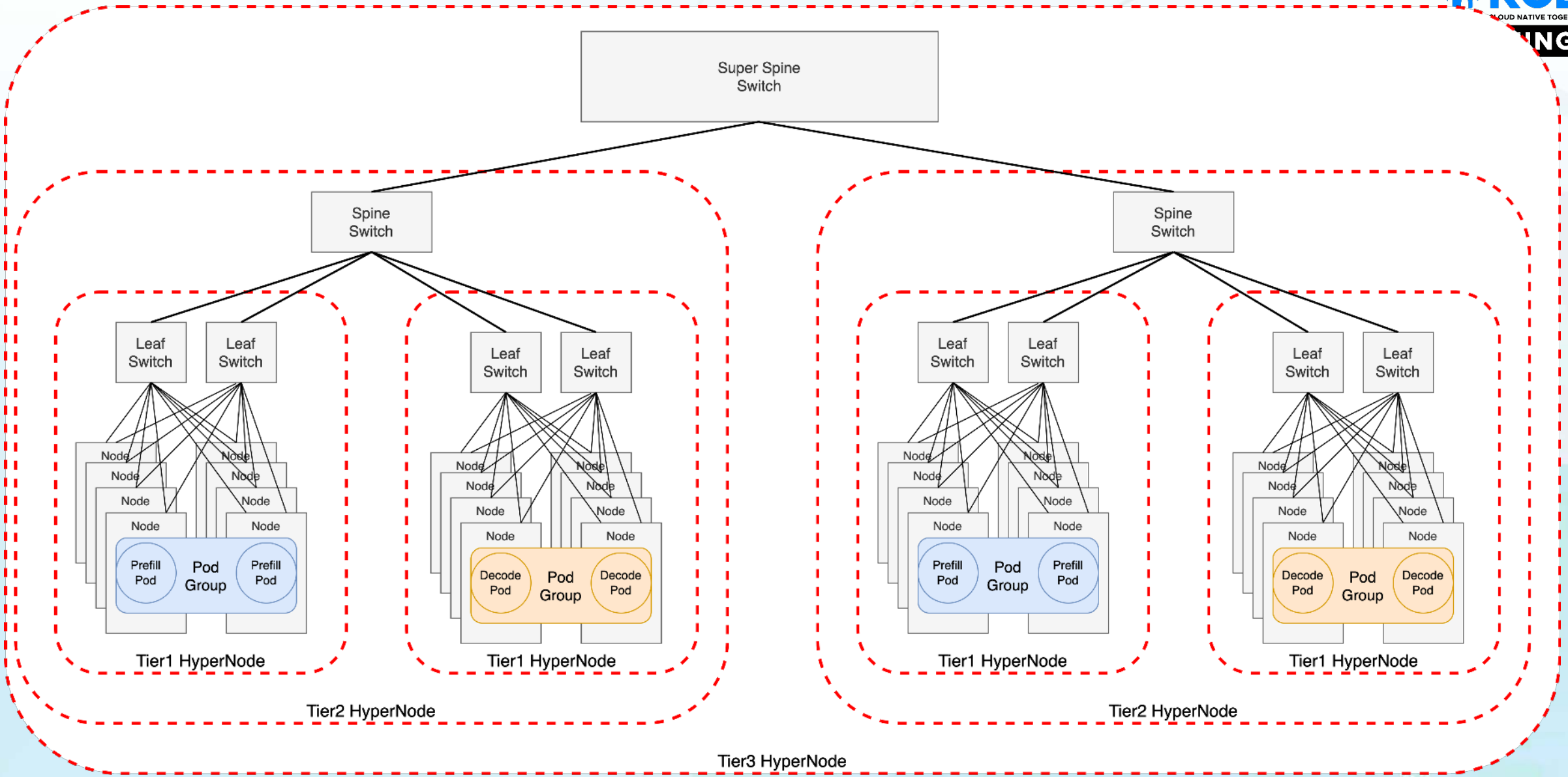
- mode：支持hard和soft两种模式。
- hard：硬约束，作业内的任务必须部署在同一个HyperNode内。
- soft：软约束，尽可能将作业部署在同一个HyperNode下。
- highestTierAllowed：与hard模式配合使用，表示作业允许跨到哪层HyperNode部署。

例如，以下配置表示作业只能部署在2层及以下的HyperNode内，否则作业将处于Pending状态：

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

Volcano Job/PodGroup配置网络拓扑约束示例

PD分离部署实践


Super Spine Switch
Spine Switch
Spine Switch
Leaf Switch
Leaf Switch
Leaf Switch
Leaf Switch
Leaf Switch
Leaf Switch
Leaf Switch
Leaf Switch
Node
Prefill Pod
Pod Group
Prefill Pod
Decode Pod
Pod Group
Decode Pod
Tier1 HyperNode
Tier1 HyperNode
Tier1 HyperNode
Tier1 HyperNode
Tier2 HyperNode
Tier2 HyperNode
Tier3 HyperNode

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 03

昇腾NPU生态支持

为了实现异构硬件的高效利用，应用框架层面的良好支持必不可少。

推理方面，vLLM 是大模型推理的首选框架之一。

训练方面，verl 是新兴的 RL 训练库，社区极为活跃。

针对这两个优秀框架的昇腾NPU支持，正在不断产生新的进展。

昇腾NPU上的大模型推理

vLLM：专为大语言模型提供高吞吐量和内存高效推理服务的开源引擎

vLLM原生支持：支持vLLM开源生态

- 0 day & 原生支持vLLM社区新模型、新特性
- pip install vllm vllm-ascend

vLLM Ascend Plugin 是vLLM社区的官方项目，隶属于LF Data & AI基金会，由vLLM社区开发者共同维护，是一个让vLLM在Ascend NPU无缝运行的后端插件。

此插件是 vLLM 社区中支持昇腾后端的推荐方式，它遵循 [RFC]: Hardware pluggable所述原则：通过解耦的方式提供了vLLM对Ascend NPU的支持。

2025年2月19日，vLLM社区的vLLM Ascend项目正式发布了第一个RC版本：0.7.1rc1。

昇腾NPU上的大模型推理

vllm-ascend

支持特性

Feature	Supported	Note
Chunked Prefill		Plan in 2025 Q1
Automatic Prefix Caching		Plan in 2025 Q1
LoRA	✗	Plan in 2025 Q1
Prompt adapter	✗	Plan in 2025 Q1
Speculative decoding	✅	
Pooling	✅	The accuracy is not correct, it'll be fixed in 2025 Q2
Enc-dec	✗	Plan in 2025 Q2
Multi Modality	✅ (LLaVA/Qwen2-vl/Qwen2-audio/internVL)	Add more model support in 2025 Q2
LogProbs	✅	
Prompt logProbs	✅	
Async output	✅	
Multi step scheduler	✅	
Best of	✅	
Beam search	✅	
Guided Decoding	✅	Find more details at the issue
Tensor Parallel	✅	
Pipeline Parallel	✅	

支持模型

Model	Supported	Note
DeepSeek v3	✅	
DeepSeek R1	✅	
DeepSeek Distill (Qwen/llama)	✅	
Qwen 2.5	✅	
Qwen2-VL	✅	
Qwen2-Audio	✅	
NiniCPM	✅	
LLama3.1/3.2	✅	
Mistral		Need test
DeepSeek v2.5		Need test
Gemma-2		Need test
Baichuan		Need test
Internlm	✅	
ChatGLM	✅	
InternVL 2.5	✅	
GLM-4v		Need test
Molomo	✅	
LLaVA 1.5	✅	

https://github.com/vllm-project/vllm
https://github.com/vllm-project/vllm-ascend

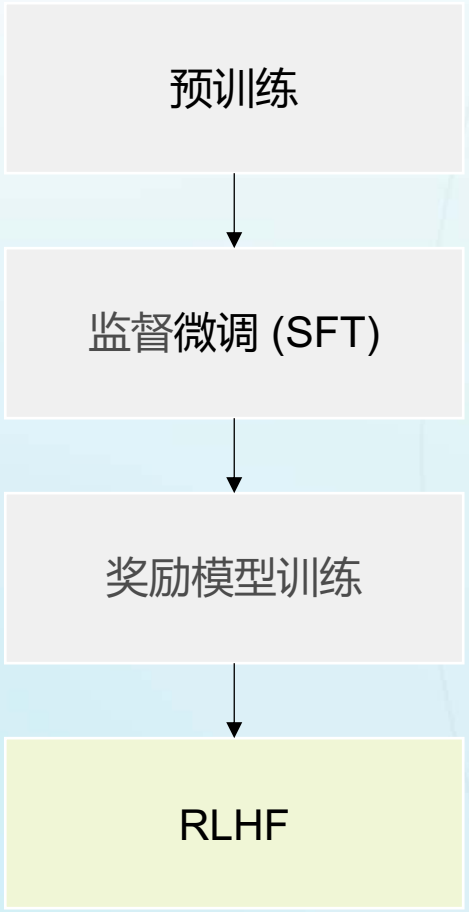
https://twitter.com/vllm_project

https://slack.vllm.ai sig-ascend channel

昇腾NPU上的RLHF

RLHF - 基于人类反馈的强化学习

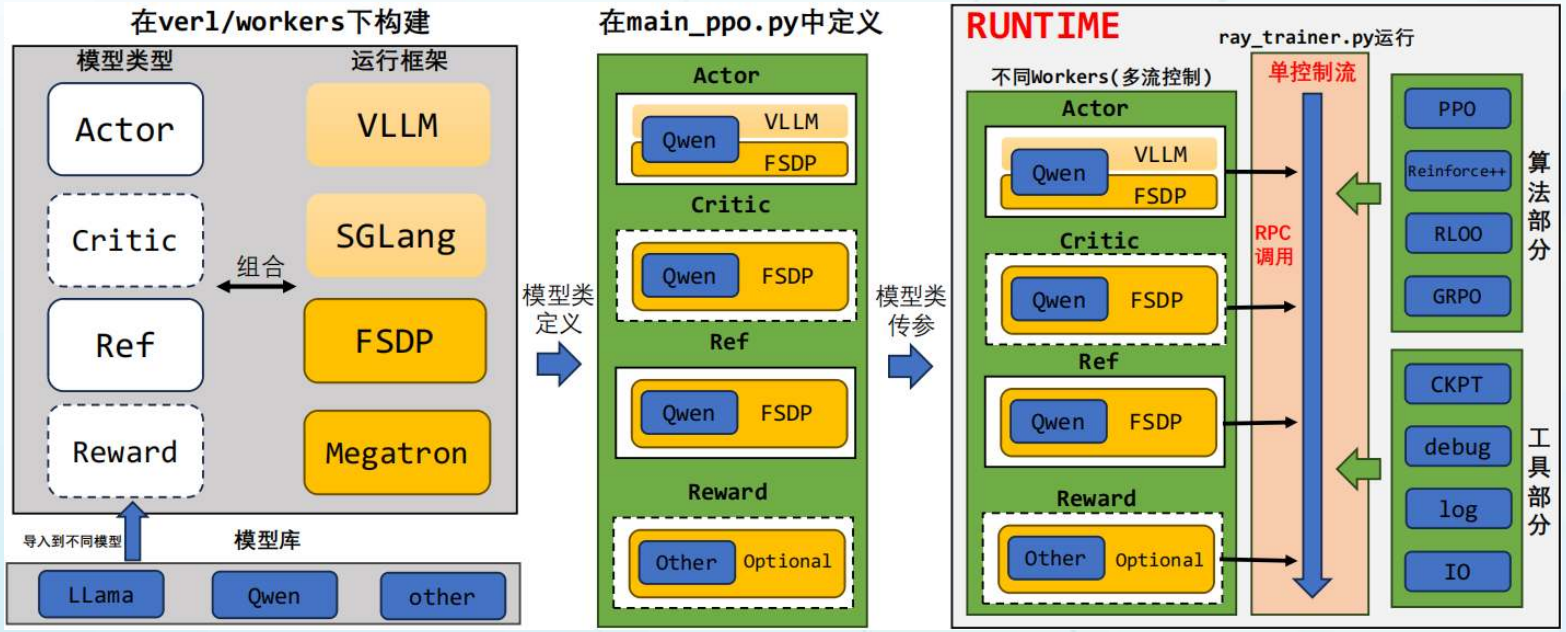
LLM训练流程



模型	使用的RLHF算法
DeepSeek-R1	GRPO
GPT-4	PPO
Mistral-7B	ReMax

昇腾NPU上的RLHF

verl 是一个灵活、高效且可用于生产用途的 RL 训练库，适用于大型语言模型（LLM）。
项目极为活跃，正在快速更新迭代。



#85 #198 通过引入MindSpeed解决Megatron后端的Ascend适配问题。

#332 解决Pytorch FSDP后端的PPO、GRPO和SFT的流程打通。

#465 打通Ascend CI流程

KCD
GROWING CLOUD NATIVE TOGETHER
BEIJING

Part 04

生产环境中管理算力负载

魔乐社区对于AI负载的管理

模型 8700+　数据集1000+　体验空间200+

魔乐社区体验空间：

提供机器学习和深度学习算法的应用案例，在浏览器即可演示模型的交互式应用程序。

gradio

Streamlit

docker

KCD GROWING CLOUD NATIVE TOGETHER BEIJING

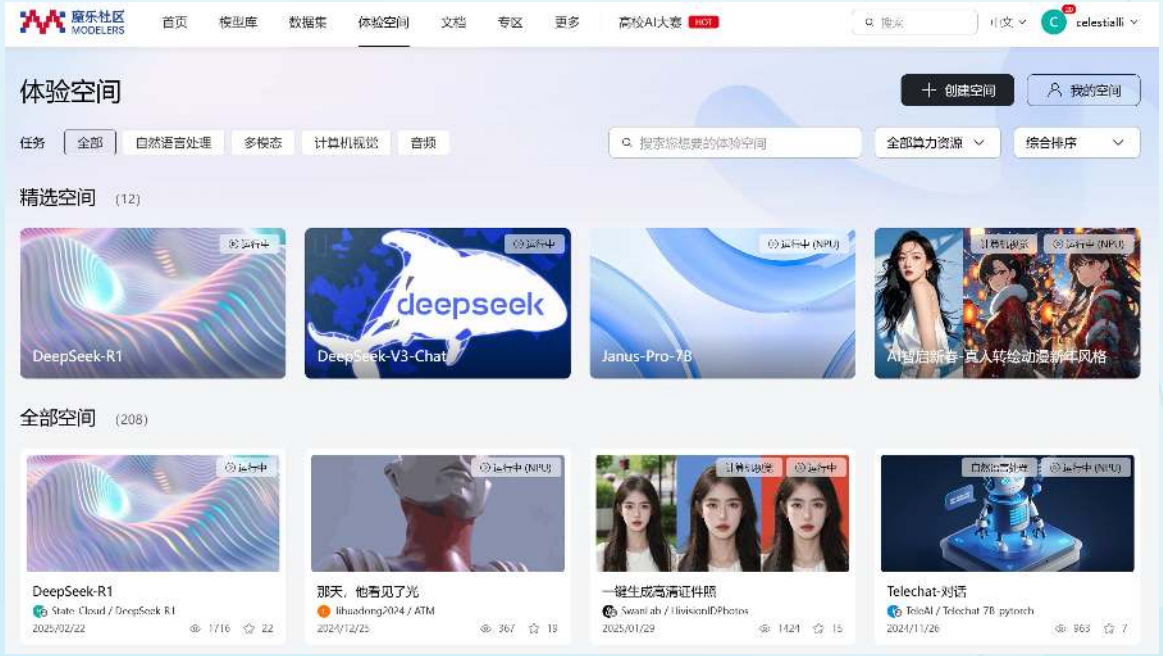
模型生产者

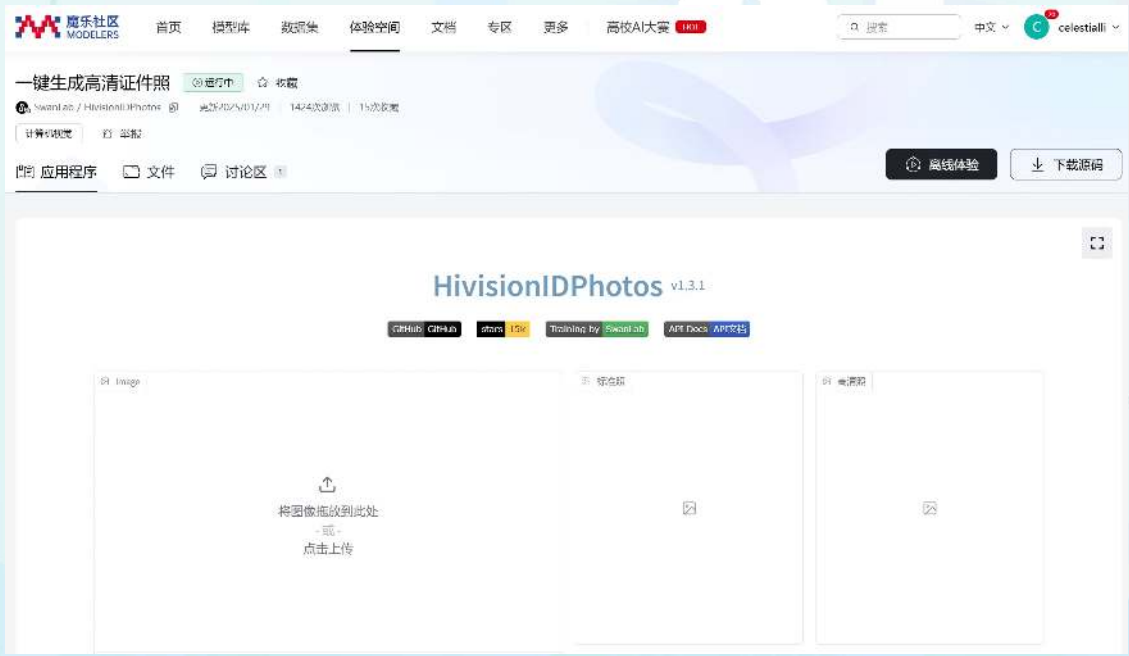
发布模型、体验空间

模型使用者

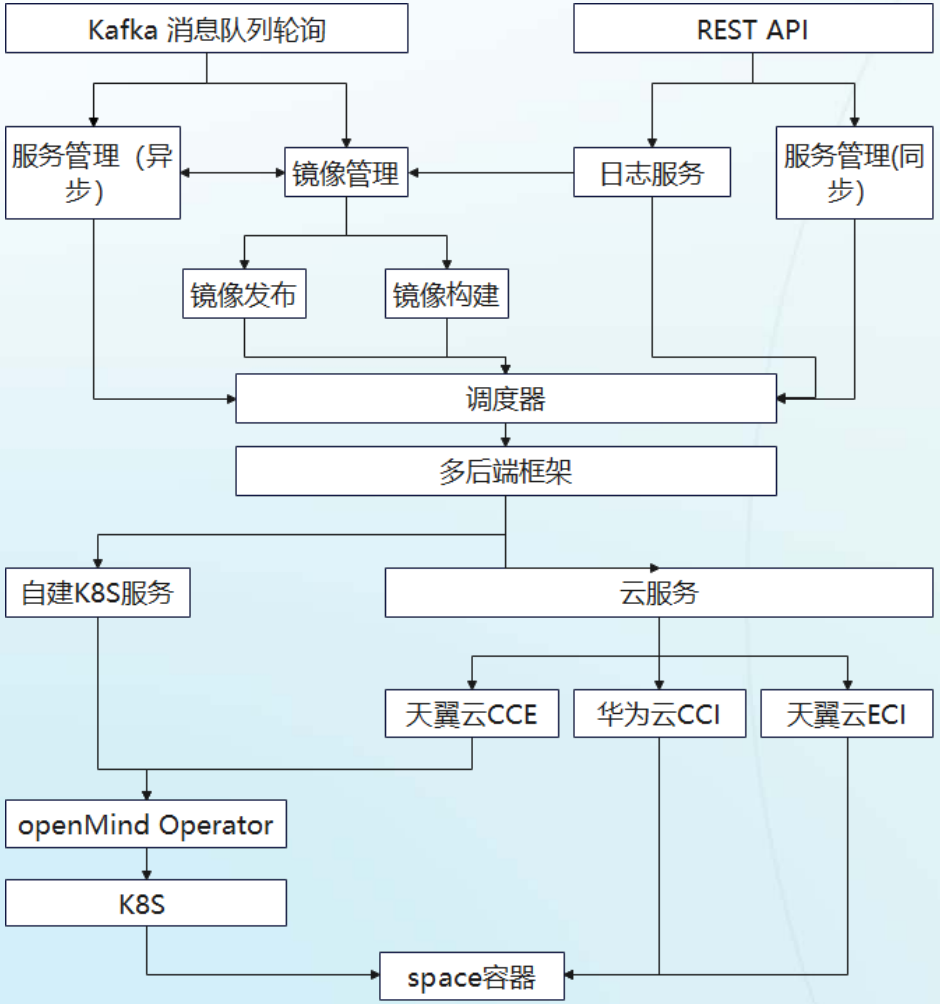
应用开发者

开发体验空间

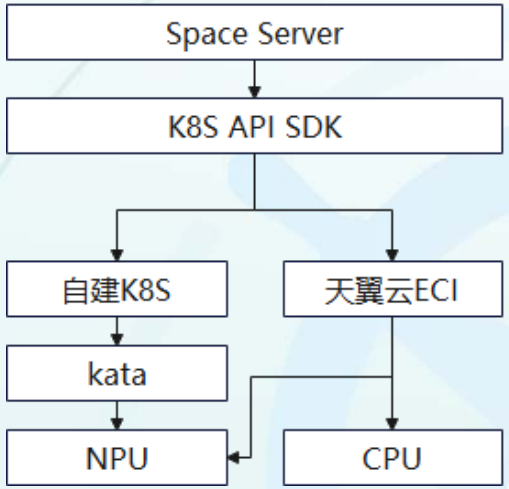




魔乐社区对于AI负载的管理



openMind Space Server架构概览



魔乐社区的体验空间算力后端同时支持自建资源和云服务资源。
自建资源使用Kubernetes Operator来管理NPU算力。

魔乐社区 MODELERS https://modelers.cn/

Part 05
小结

小结

- **第一章：节点内拓扑感知调度**

Volcano针对昇腾NPU的拓扑结构，优化节点内任务调度，最大化通信效率。

- **第二章：跨节点网络拓扑感知调度**

Volcano定义了统一的网络拓扑API **HyperNode**，屏蔽底层组网差异，将频繁通信的任务组调度至同一性能域，提升分布式训练/推理任务性能。

- **第三章：昇腾NPU生态支持**

昇腾NPU对于AI应用框架层的生态支持逐渐完备，支持了互联网大厂使用昇腾NPU承载较大规模AI训练或推理的需求。

- **第四章：生产环境中管理算力负载**

通过一个实际项目，分享了如何将算力负载的高效管理融入较大规模的生产环境。

未来展望

Volcano将继续增强对昇腾NPU调度的支持及持续优化网络拓扑感知调度功能，未来计划：

昇腾NPU调度：

- 持续适配最新的昇腾NPU架构，最大化昇腾NPU的性能

网络拓扑感知调度：

- 支持从节点标签**自动转换**为HyperNode CR，帮助用户迁移到Volcano；
- 集成底层网络拓扑自动发现工具，简化HyperNode的管理；
- **提供命令行工具**，方便用户查看和管理HyperNode层级结构；
- 支持**Task粒度**的网络拓扑感知调度；
- 支持HyperNode纳管节点的健康状态和网络健康状态上报。

未来展望

昇腾NPU生态支持：

- 原生支持的应用生态逐渐繁荣
- 原生支持的应用支持开箱即用 (例如，pip install xxx)

生产环境中管理算力负载：

- 越来越好地利用多卡训练、推理以提高效率
- 越来越多地使用跨节点能力
- 结合更高效的资源管理能力，更高效地利用硬件资源

Volcano社区生态

- CNCF首个云原生批量计算平台
- 4.5k Github Star，1k+ Fork
- 来自30+ 国家 800+ Contributors，60+ 企业生产落地

Volcano社区用户

加入社区

Volcano Website:

Volcano Github：

Volcano Slack：

vLLM X：

vllm-ascend Github：

vLLM Slack：

Thanks.