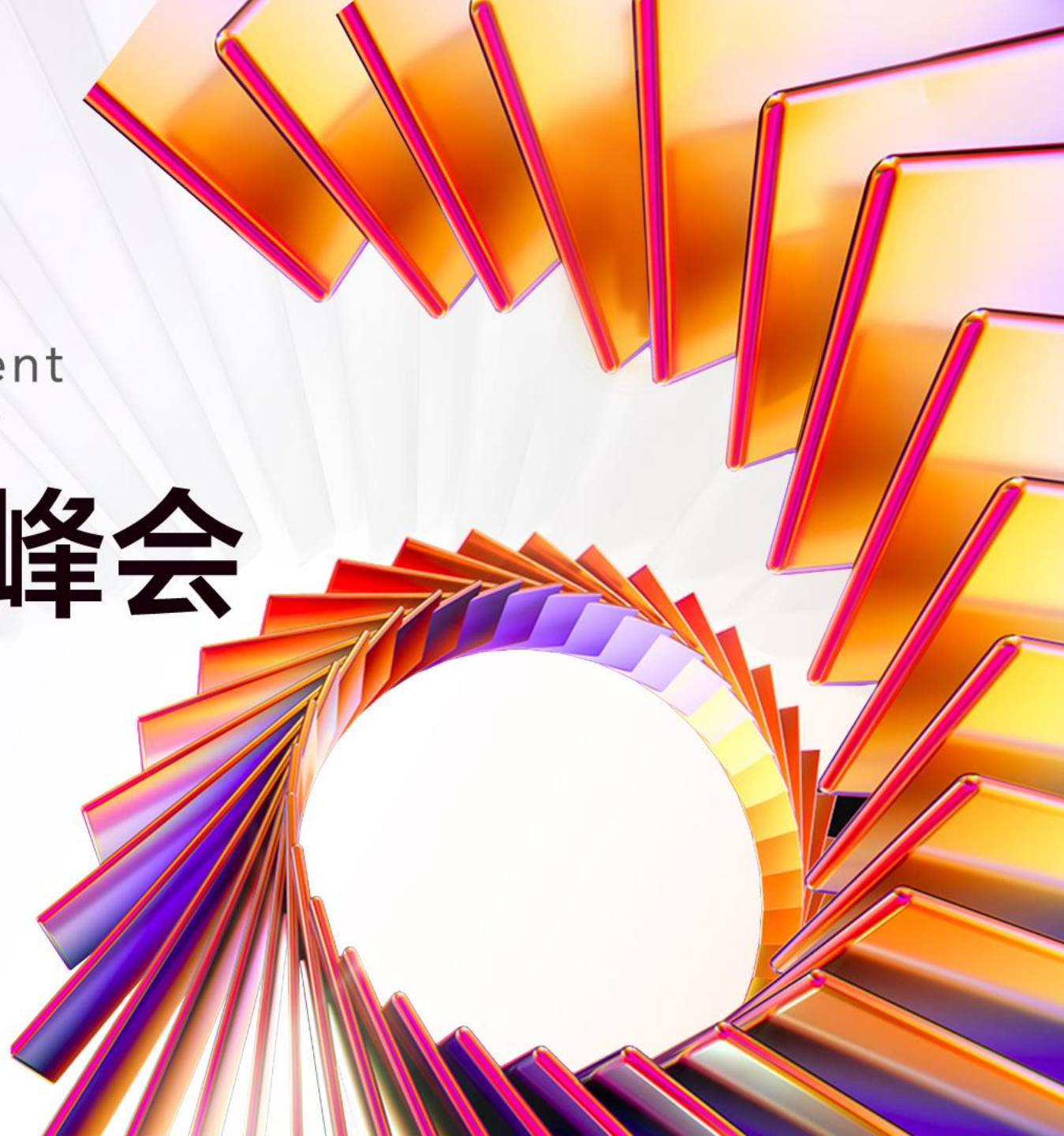
AiDD 8th 2025

第8届 AI+ Development Digital Summit AI+研发数字峰会

拥抱AI 重塑研发

11月14-15日 | 深圳



AiDD × K

2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会

Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战路与商业模式重构

论坛3: AgenticAI产品创新与交互设计

Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

2-hour Speech: 回归本质

用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，仍然考验“看见”的“同理心”

Panel 1: 出海前瞻

“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式

Panel 2: 失败复盘

为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

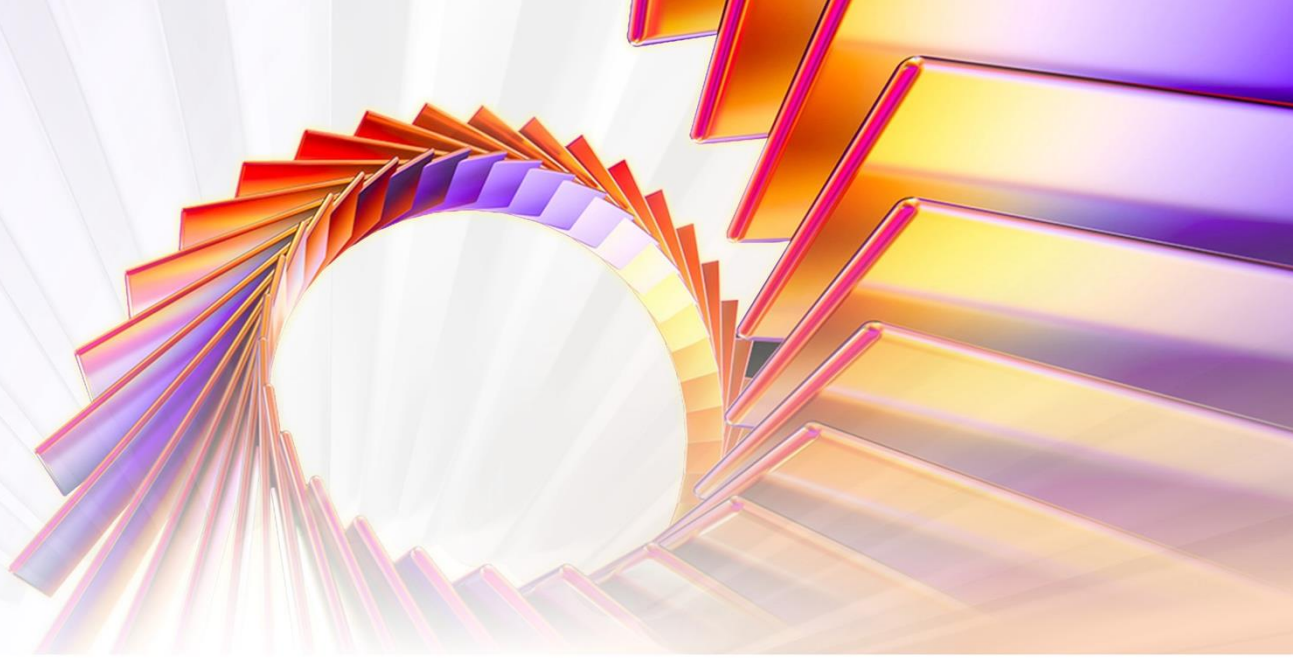
智能重构产品 数据驱动增长

Reinventing Products with Intelligence, Driven by Data

80+	40+	10+	800+
业界大咖	创新案例	分论坛	听众规模
汇聚产品大咖的真知灼见	开拓全新AI+产品视角	涵盖产品到商业增长的全链路	共同探索产品的智能重构

扫码查看会议详情

AiDD 8th 2025 | 11月14-15日 | 深圳
第8届 AI+ Development Digital Summit
AI+研发数字峰会
拥抱AI 重塑研发



构建 AI 原生调度生态: 通过 HAMi 释放异构 AI 算力芯片的推理性能

张潇 | 密瓜智能

张潇

密瓜智能创始人及CEO

拥有10余年云原生、容器、AI Infra领域研发及架构设计经验。曾带领20+研发团队主导DaoCloud容器平台架构设计及技术研发，产品多次入选Gartner容器管理魔力象限。拥有5项以上云计算相关发明专利，研究方向涵盖容器管理、多云、多集群、大规模集群高可用、AI Infra等。

目录
CONTENTS

PART 1

背景与挑战

GPU 算力供需两端现状

需求端

- 贸易禁令影响，品牌配置杂乱
- 规模各不相同，难以集中资源
- 资源使用粗犷，资源利用率低
- 新兴行业起步，运维人才储备不足
- 运营缺乏规范，用户成本居高不下
- 算力中心 多种异构 加速器 管理成本极高

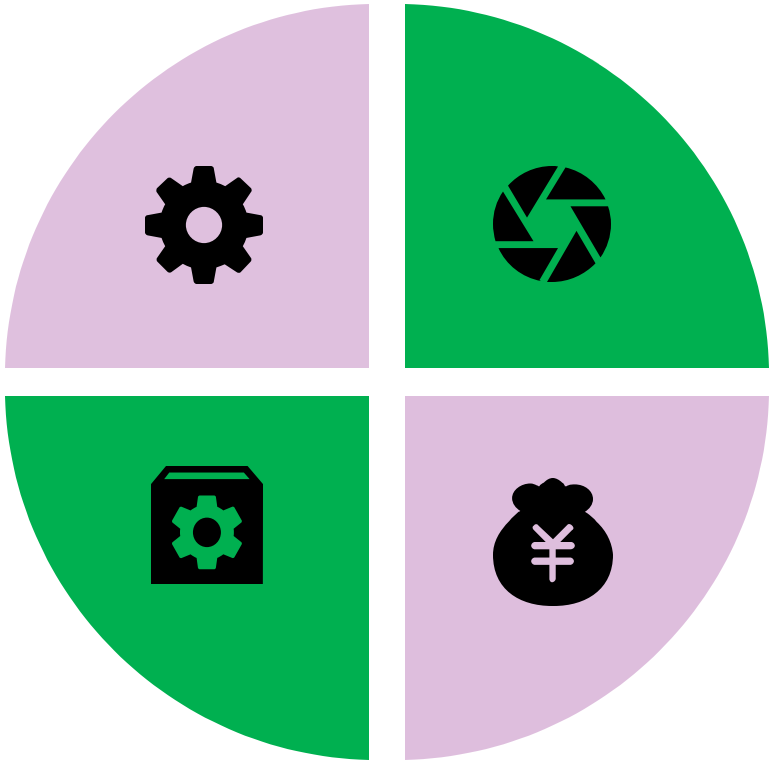
供给端

- GPU 资源属于稀缺资源，用户难以获取
- 推理等碎片化需求凸显，亟需优化资源调度
- 人工智能行业进入门槛高，缺乏技术指导团队
- 市场与战略双驱动，多元异构算力成为硬需求

以 GPU 为主的异构算力管理痛点

资源分散，难以统一管理

无资源池，无整体调度，监控，管理平面



资源利用率低

无虚拟化&资源池化技术，GPU使用为独占模式。

市场方案偏硬件、平台型，不通用，彻底。

资源分配不灵活

无法实现资源超分，不能指定任务所需的资源大小和种类。

采购大量冗余算力满足业务需求

由于异构算力的利用率低下，所以企业必须加大采购数量才能满足需求，而异构算力本身成本高昂，为企业带来了很大负担。

AI Infra Landscape

AiDD 8th 2025

模型

Llama · Qwen · DeepSeek · Gemma · Mistral · Molmo · Phi · 百川智能 BAICHUAN AI · ChatGLM 智普 · Intern

推理编排系统

llm-d · vllm-project/aibrix · Dynamo · llmaz

KV-Cache Store · AI Gateway · 弹性 HPA · PD分离

训练框架

Megatron-LM · 智源研究院 FlagScale · Colossal · dlrover

异构推理引擎

vLLM · SGL · TensorRT · NVIDIA TRITON INFERENCE SERVER

任务编排

VOLCANO · Kubeflow · kueue · kubeRay

异构算力调度

GPU池化 · 数据/镜像管理 · 队列系统 · AI调度 · 拓扑感知 · job任务编排

监控和观测 · 设备/驱动管理 · 容错和弹性 · 存储对接

异构基础设施 纳管

HYGON 中科海光 · 天数智芯 Iluvatar CoreX · Ascend · NVIDIA · Enflame 燧原科技 · METAX 沐曦 · 壁仞科技 BIREN TECHNOLOGY · Cambricon 寒武纪 · AMD EPYC INSTINCT · intel GAUDI · lustre · JuiceFS · ALLUXIO

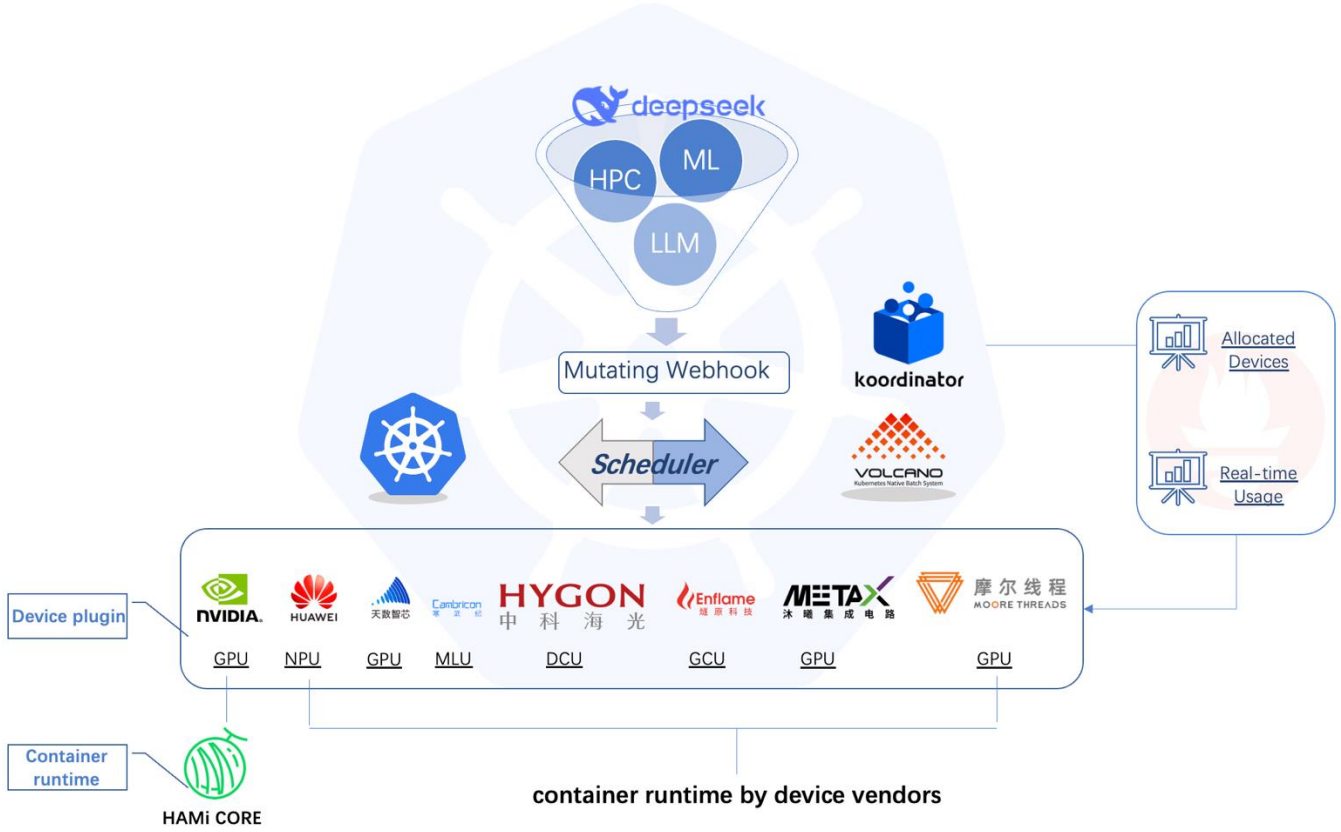
物理机 · 虚拟机 · 边缘 · 私有云 · 公有云 · 混合云

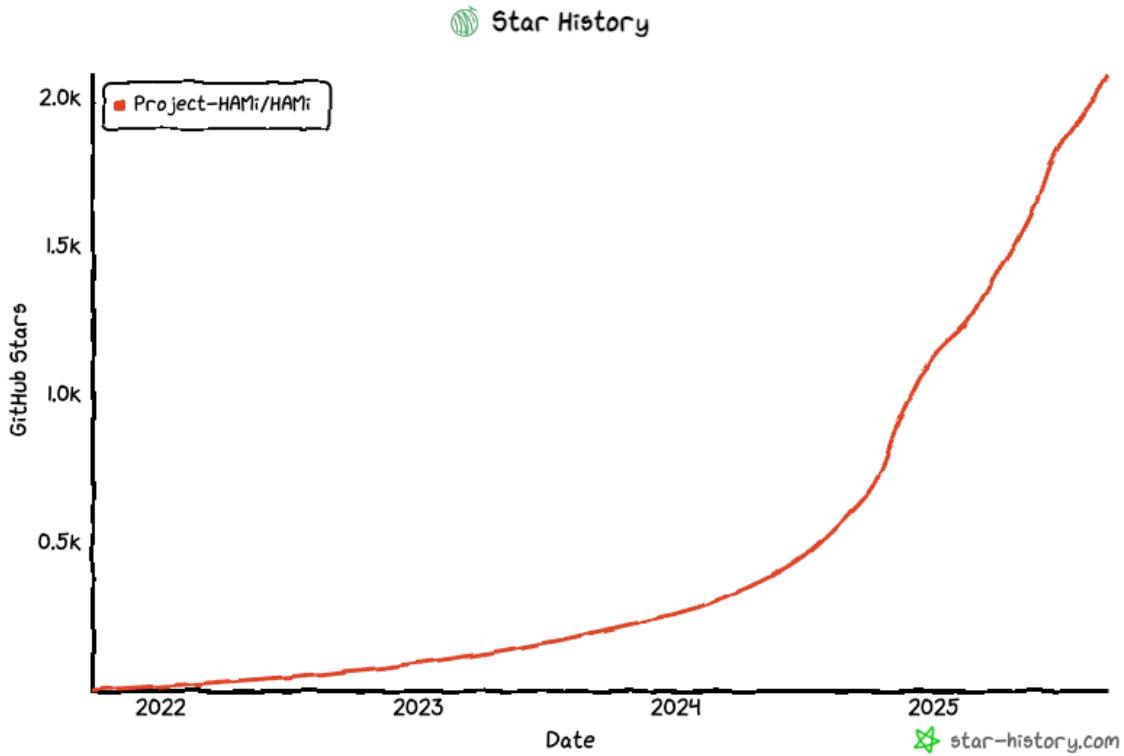
PART 2

HAMi

HAMi

HAMi 异构算力调度、管理中间件，超过 **15+** 个国家 **350+** 开发者贡献，**200+** 的企业落地，开源 **一年内** 即成为 **CNCF 基金会 Sandbox 项目** 与 **CNAI Landscape 项目**，**唯一关注 AI Infra 与 异构 AI 调度 的开源项目。目前已**支持超过 8 款 AI 芯片统一管理、高效调度、动态观测，最大化利用率。

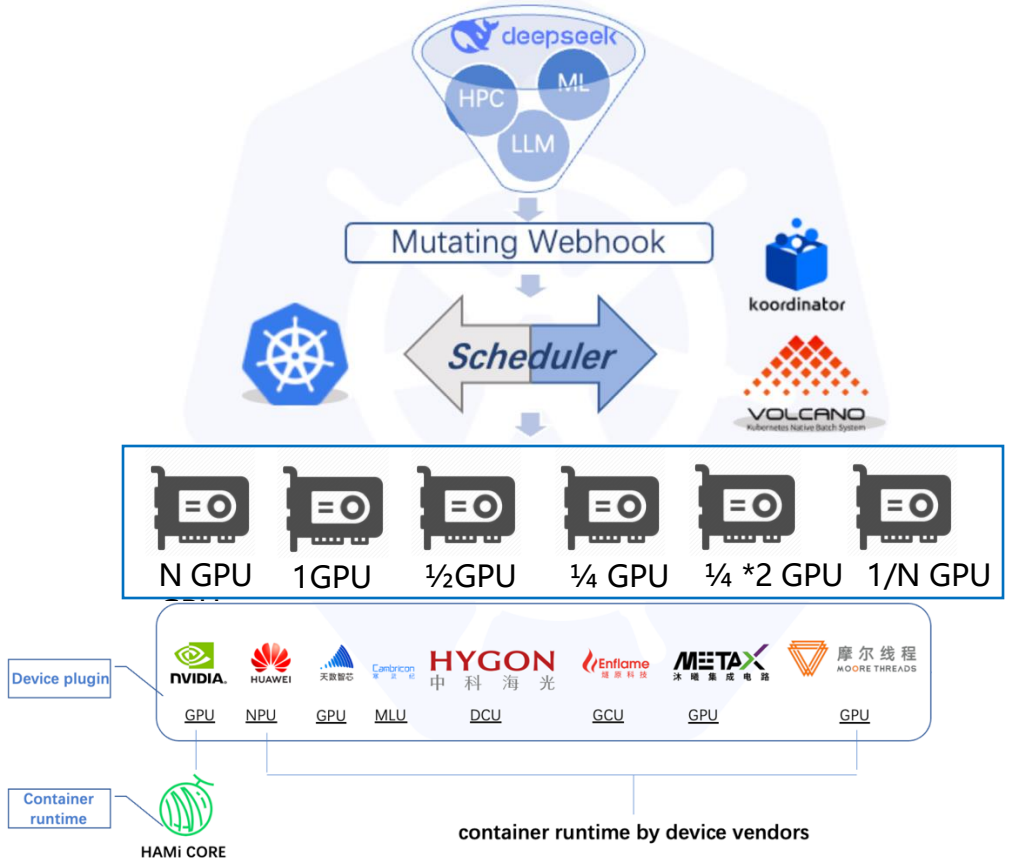




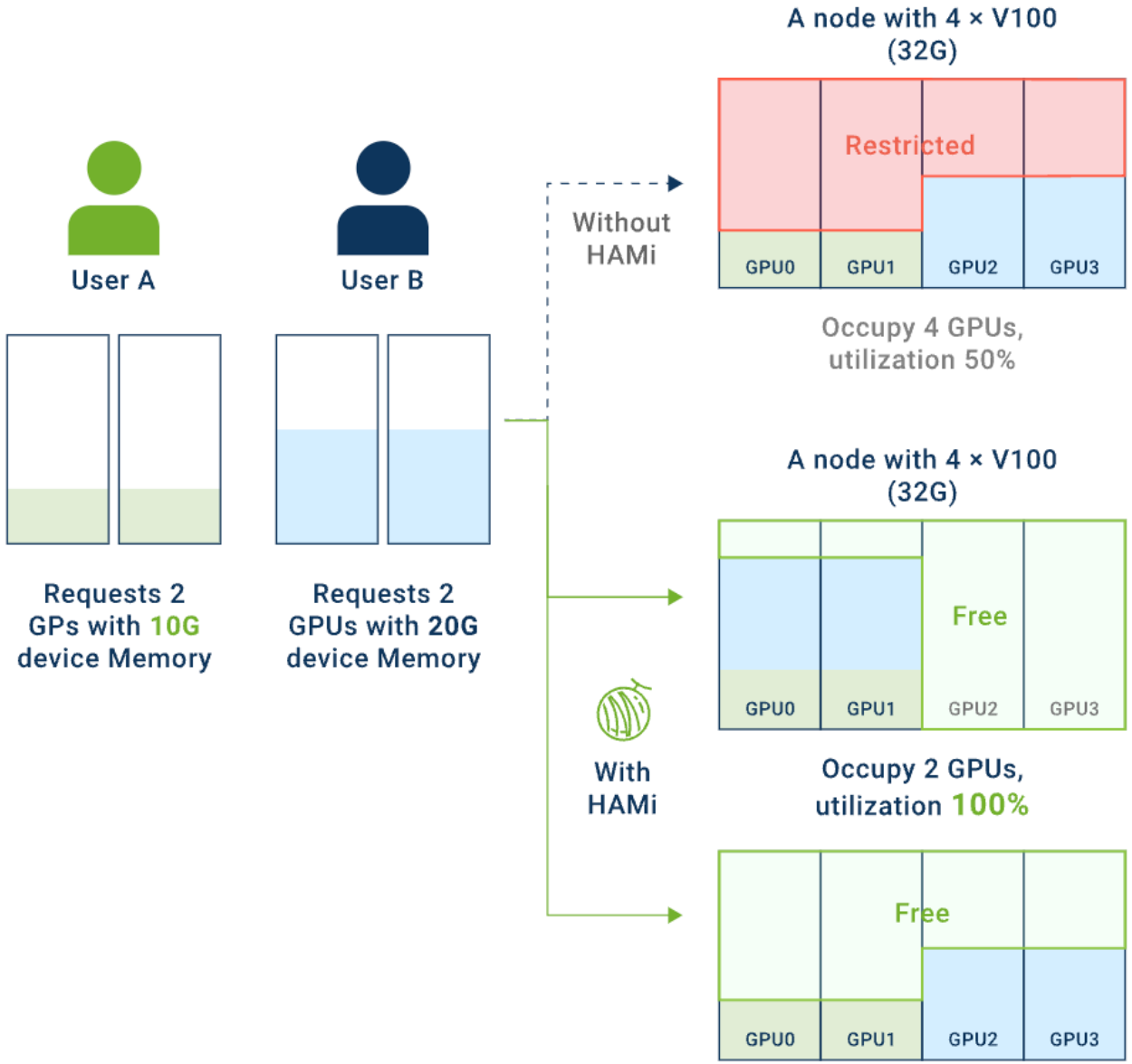
NVIDIA Cambricon 寒武纪 HYGON METAX 沐曦 Ascend 天数智芯 Iluvatar CoreX 摩尔线程 MOORE THREADS 昆仑芯 KUNLUNXIN Enflame 燧原科技

HAMi 主要功能

- 支持多种 GPU 设备，提供统一管理、高效调度能力 (NVIDIA、寒武纪、沐曦、天数、摩尔线程、海光、昇腾、燧原、昆仑芯、AWS 推理芯片)
- 弹性、按需、可靠 GPU共享，多个任务可以共享 GPU
- 显存，算力超配
- 任务优先级机制
- 显存自动弹性扩缩容
- 丰富、灵活、特定用户场景的调度策略
  - 支持指定设备型号，将任务调度到指定型号的 GPU 卡上
  - 支持指定 UUID，将任务调度到指定 UUID 的 GPU 卡上
  - 节点& AI 芯片级别的 堆叠和打散调度。
    - Binpack
    - Spread
  - GPU 亲和性 拓扑感知调度
  - Numa affinity 亲和性调度
- 企业级配额机制 & 完备的可观测性体系
- 上下游推理框架生态全支持(**vLLM, Xinference,Kueue,Koordinator,Volcano**)

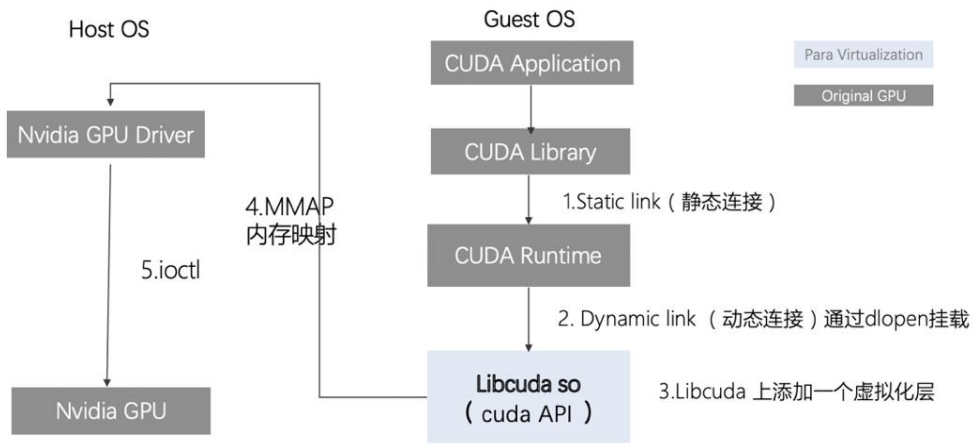


GPU 共享与虚拟化

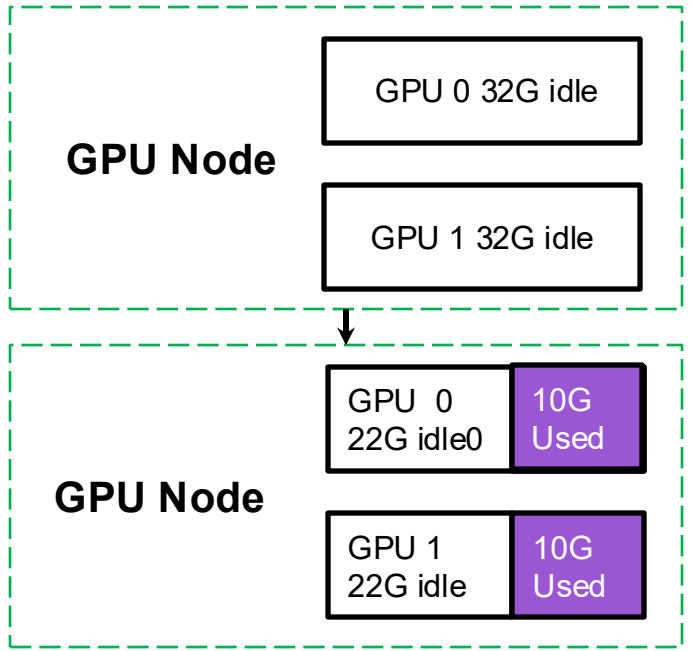


HAMi 提供灵活、可靠、弹性、按需虚拟化能力

- 1% 算力，1M 显存 极细粒度
- 强隔离，QOS 保证



开箱即用(应用无侵入)



参数描述:

nvidia.com/gpu: 表示容器中可见 GPU 的数量。

nvidia.com/gpumem: 指定每个 GPU 使用的内存大小。如果没有设置，默认是使用所有可用的 GPU 内存。

nvidia.com/gpucores : 指定每个 GPU 使用的百分比。

```
$ cat <<EOF | kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod12
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu:18.04
      command: ["bash", "-c", "sleep 86400"]
      resources:
        limits:
          nvidia.com/gpu: 2
          nvidia.com/gpumem: 10240
          nvidia.com/gpucores: 30
```

```
root@gpu-demo-6b6b88b75b-x9tjb:~# nvidia-smi
[HAMI-core Msg(35:140111864956736:libvgpu.c:836)]: Initializing.....
Thu Apr 18 06:22:54 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.12             Driver Version: 535.104.12   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA A800 80GB PCIe          On  | 00000000:13:00.0 Off |                    0 |
| N/A   36C    P0              81W / 300W |      0MiB / 10240MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+
|   1  NVIDIA A800 80GB PCIe          On  | 00000000:1C:00.0 Off |                    0 |
| N/A   39C    P0              82W / 300W |      0MiB / 10240MiB |      0%      Default |
|                                         |                      |             Disabled |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
+---------------------------------------------------------------------------------------+
[HAMI-core Msg(35:140111864956736:multiprocess_memory_limit.c:434)]: Calling exit handler 35
root@gpu-demo-6b6b88b75b-x9tjb:~#
```

HAMi - 丰富、灵活、贴近真实场景的调度策略

01 基于优先级的调度

当高优 Pod 开始使用 GPU 算力，所有普通 Pod会立刻被暂停使用 GPU 算力，直到高优 Pod 计算任务结束，普通任务会重新继续使用 GPU 算力。

02 基于 GPU 卡型号的调度

在 vGPU 模式下，当存在多种型号的 GPU 卡时，支持将 Pod 调度到指定型号的 GPU 卡上。

03 基于 GPU 卡调度

在选定节点后进行卡维度的分配决策，为 Pod 中每个容器选择和分配节点上的 GPU 卡：

Binpack：优先选择资源利用率高的 GPU 卡，容器集中到同一块卡上

Spread：优先选择资源利用率低的 GPU 卡，容器分散到不同的卡上

04 基于节点调度

根据节点 GPU 算力和显存进行行加权平均打分提供 2 种策略：

Binpack：GPU 分配率越高，打分越高，Pod 集中调度到同一个节点

Spread：GPU 分配率越高，打分越低，Pod 分散调度到各个节点

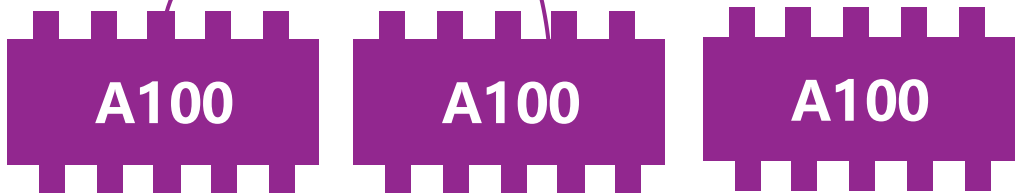
指定 GPU 型号

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod2
  annotations:
    nvidia.com/nouse-gputype: "V100"
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu:18.04
      command: ["bash", "-c", "sleep 86400"]
      resources:
        limits:
          nvidia.com/gpu: 2
```

node1

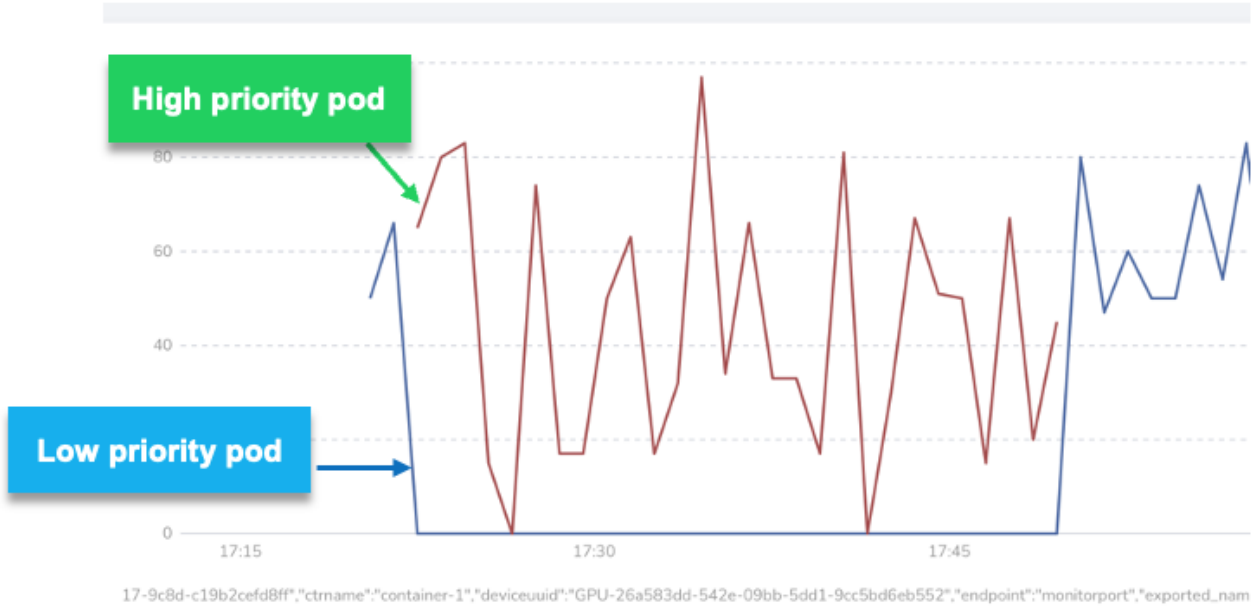

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod
  annotations:
    nvidia.com/use-gputype: "A100"
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu:18.04
      command: ["bash", "-c", "sleep 86400"]
      resources:
        limits:
          nvidia.com/gpu: 2
```

node2



任务优先级抢占

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: gpu-pod
spec:
  containers:
    - name: ubuntu-container
      image: ubuntu:18.04
      command: ["bash", "-c", "sleep 86400"]
      env:
      # vgpu task priority 0 for high and 1 for low
        - name: CUDA_TASK_PRIORITY'
          value: '0'
      resources:
        limits:
          nvidia.com/gpu: 1
          nvidia.com/gpumem: 3000
          nvidia.com/gpucores: 30
```



- 支持任务设置不同的优先级 (QOS)

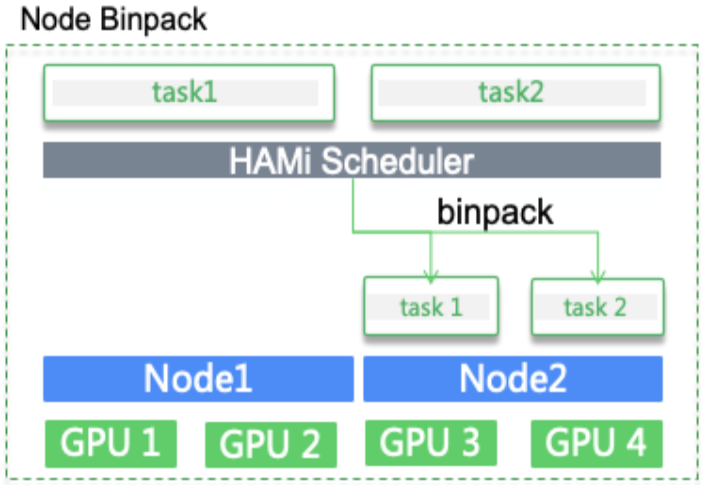
当 GPU 没有被高优先级任务使用的时候，低优先级任务就成了 『山中无老虎的猴子』

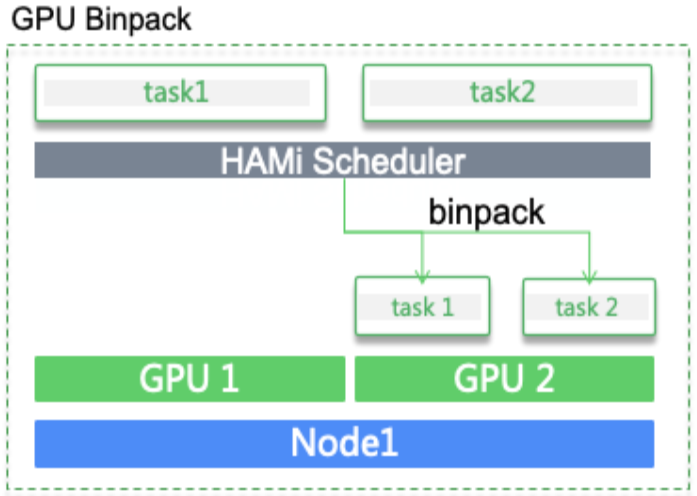
一旦高优先级任务运行，GPU 算力将得到优先保证，低优先级任务会被 “hang” 住，

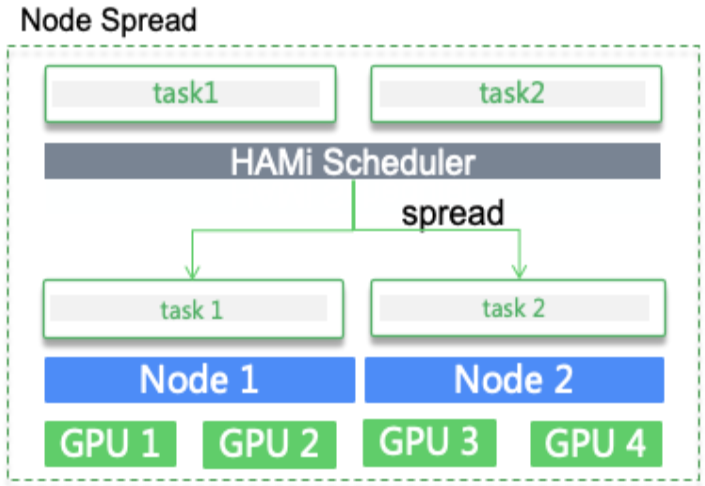
等高优先级任务释放，再次成为 『山中无老虎的猴子』

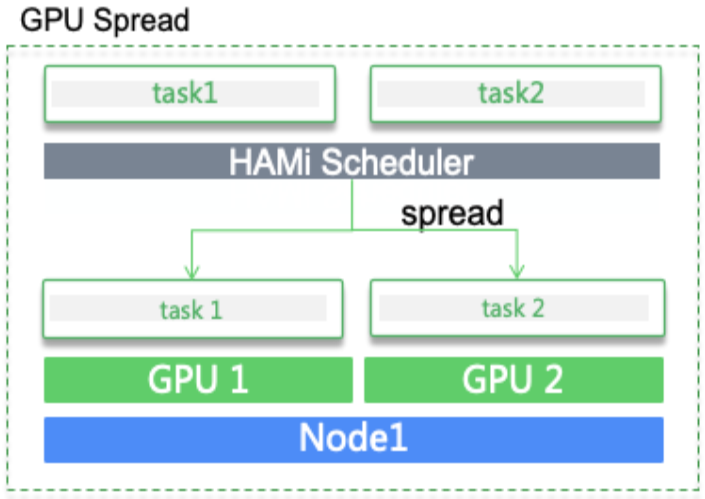
- 技术实现：代码会让对应进程进入无法执行 CUDA ” launchKernel” 的睡眠周期

调度策略(binpack & spread)



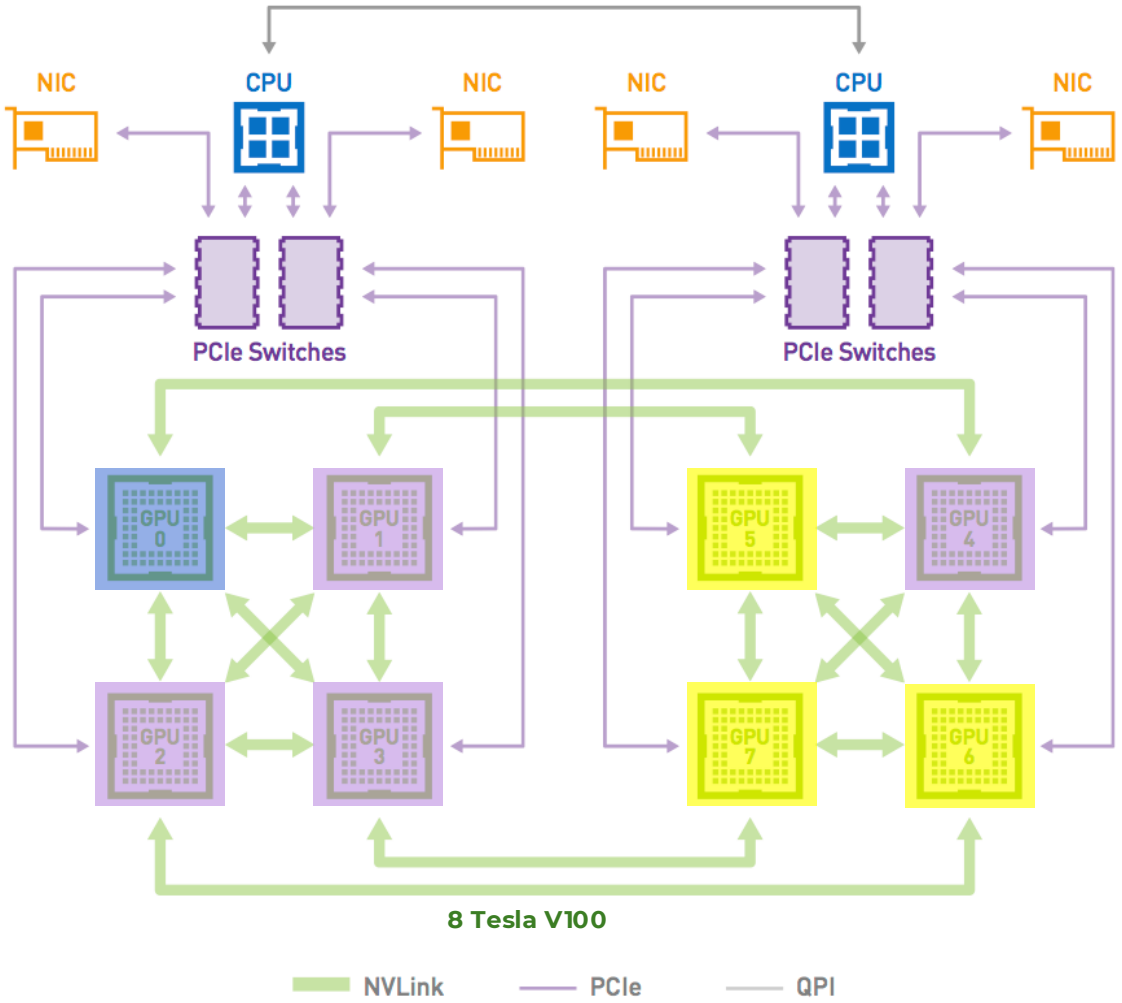


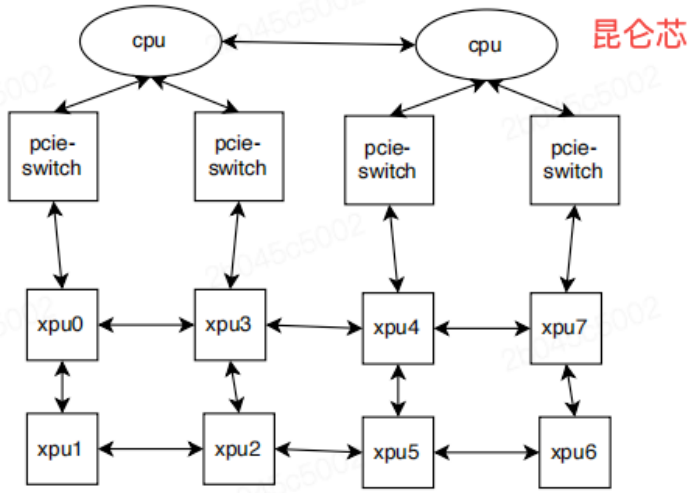




- 实际场景和需求
- 小任务饿死大任务
- 高优先级任务避免放在一个篮子中

GPU 拓扑感知调度

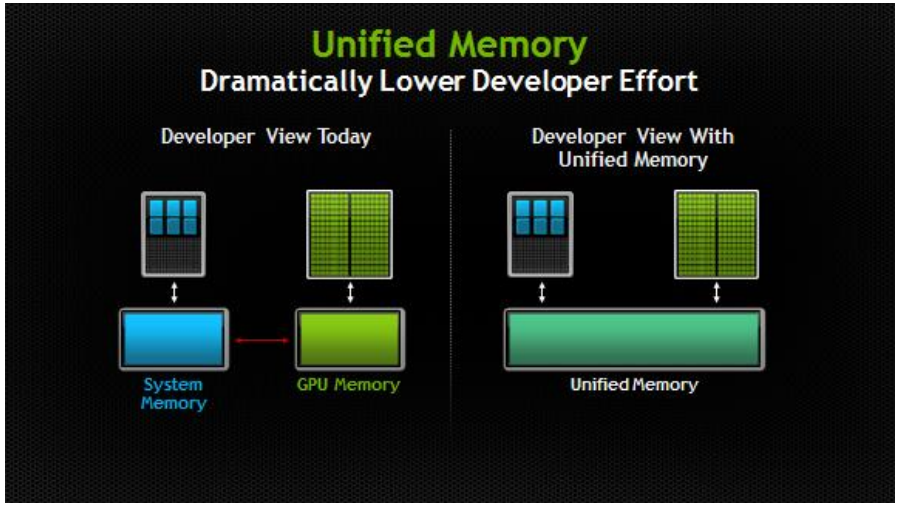




NVLink(25GB/s – 1800GB/s) > PCie(16GB/s)

通过 HAMi 内置 拓扑感知调度，能够让应用选择更高效的多卡进行使用

GPU 超配



23G Device Memory

Can host 1 13B inference

23G Device Memory | 46G virtual Device memory (in memory)

Can host 3 13B inferences

Before

```
+-----------------------------------------+------------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|=========================================+========================+======================|
|   0                       On | 00000000:14:00.0 Off |                    0 |
| 0%   49C    P0   135W / 150W |  13127MiB / 23028MiB |     55%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1                       On | 00000000:15:00.0 Off |                    0 |
| 0%   48C    P0   131W / 150W |  13127MiB / 23028MiB |     54%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   2                       On | 00000000:18:00.0 Off |                    0 |
| 0%   48C    P0   132W / 150W |  13137MiB / 23028MiB |     55%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   3                       On | 00000000:1D:00.0 Off |                    0 |
| 0%   46C    P0   129W / 150W |  13127MiB / 23028MiB |     55%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   4                       On | 00000000:1E:00.0 Off |                    0 |
| 0%   49C    P0   135W / 150W |  13137MiB / 23028MiB |     56%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   5                       On | 00000000:21:00.0 Off |                    0 |
| 0%   46C    P0   132W / 150W |  13137MiB / 23028MiB |     55%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   6                       On | 00000000:25:00.0 Off |                    0 |
| 0%   47C    P0   130W / 150W |  13137MiB / 23028MiB |     55%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   7                       On | 00000000:2D:00.0 Off |                    0 |
| 0%   47C    P0   134W / 150W |  13137MiB / 23028MiB |     54%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     20907      C   python                         13135MiB |
|    1   N/A  N/A     20815      C   python                         13135MiB |
|    2   N/A  N/A     20704      C   python                         13135MiB |
|    3   N/A  N/A     20451      C   python                         13135MiB |
|    4   N/A  N/A     20512      C   python                         13135MiB |
|    5   N/A  N/A     20422      C   python                         13135MiB |
|    6   N/A  N/A     20597      C   python                         13135MiB |
|    7   N/A  N/A     20662      C   python                         13135MiB |
+-----------------------------------------------------------------------------+
```

8

After

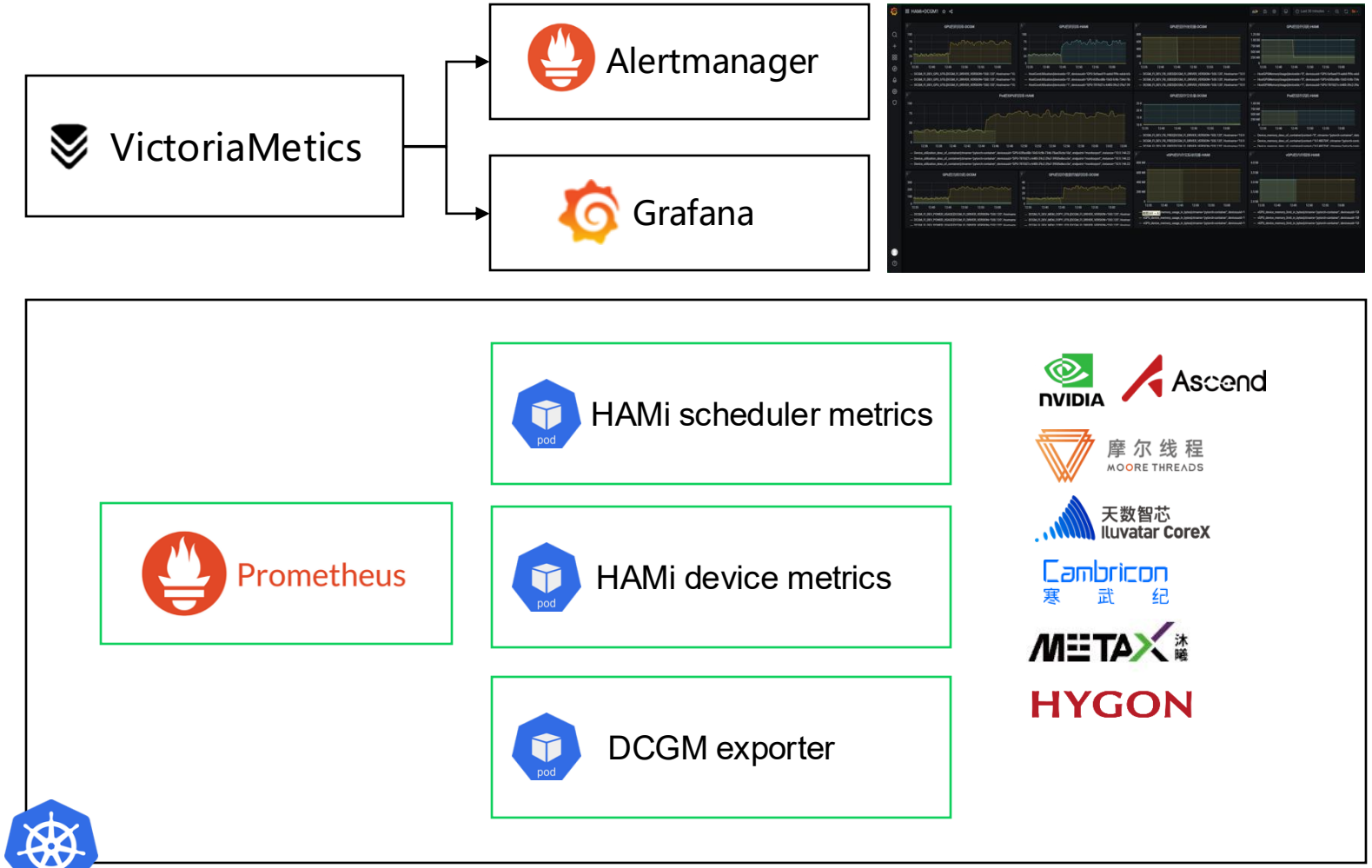
```
+-----------------------------------------+------------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|=========================================+========================+======================|
|   0                       On | 00000000:14:00.0 Off |                    0 |
| 0%   52C    P0    83W / 150W |  26430MiB / 32768MiB |     99%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1                       On | 00000000:15:00.0 Off |                    0 |
| 0%   52C    P0    81W / 150W |  26432MiB / 32768MiB |    100%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   2                       On | 00000000:18:00.0 Off |                    0 |
| 0%   50C    P0    74W / 150W |  20708MiB / 32768MiB |     66%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   3                       On | 00000000:1D:00.0 Off |                    0 |
| 0%   49C    P0    74W / 150W |  20745MiB / 32768MiB |     75%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   4                       On | 00000000:1E:00.0 Off |                    0 |
| 0%   51C    P0    69W / 150W |  20751MiB / 32768MiB |     84%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   5                       On | 00000000:21:00.0 Off |                    0 |
| 0%   50C    P0    82W / 150W |  26416MiB / 32768MiB |    100%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   6                       On | 00000000:25:00.0 Off |                    0 |
| 0%   50C    P0    76W / 150W |  26400MiB / 32768MiB |    100%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   7                       On | 00000000:2D:00.0 Off |                    0 |
| 0%   51C    P0    78W / 150W |  24594MiB / 32768MiB |     70%      Default |
|                               |                      |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A     48750      C   python                                  |
|    0   N/A  N/A     55436      C   python                                  |
|    1   N/A  N/A     48445      C   python                                  |
|    1   N/A  N/A     55696      C   python                                  |
|    2   N/A  N/A     48211      C   python                                  |
|    2   N/A  N/A     55957      C   python                                  |
|    3   N/A  N/A     47935      C   python                                  |
|    3   N/A  N/A     55893      C   python                                  |
|    4   N/A  N/A     47988      C   python                                  |
|    4   N/A  N/A     55958      C   python                                  |
|    5   N/A  N/A     48210      C   python                                  |
|    5   N/A  N/A     55271      C   python                                  |
|    6   N/A  N/A     48268      C   python                                  |
|    6   N/A  N/A     56017      C   python                                  |
|    7   N/A  N/A     48740      C   python                                  |
|    7   N/A  N/A     55685      C   python                                  |
+-----------------------------------------------------------------------------+
```

16

开启 UMI 之后，Serving 任务占着的显存如果长时间不用会换回到内存中，新的任务即可使用显存

典型场景：潮汐类型混部，小模型推理, OCR 识别、小语种翻译等场景(大量企业生产落地， 5%-10% 左右性能损失)

HAMi 可观测性 - 调度，分配，使用 尽在掌握



- K8s 调度维度: vGPU 都被哪些任务使用了，任务、GPU 绑定关系
- (节点有三张卡，每张卡被哪些任务占用、占用了多少算力、显存)
- GPU 设备维度: 反映应用运行过程中真实算力、显存使用率
- (A100 有三个任务，每个任务真实使用了 20% 算力(时间片)、30% 显存)
- 任务维度：每个任务对于显存使用的具体情况，分为 Model，Context，Data 三个维度，了解各个维度分配情况
- HAMi 也提供社区最佳实践 监控面板，一站式观测调度与真实使用情况

HAMi 与其他项目对比

Key features	HAMi	NVIDIA/k8s-device-plugin	NVIDIA/k8s-dra-driver
Support Other AI devices	✅(NVIDIA, Cambricon, Hygon, iluvatar, Huawei Ascend)	❌ only nvidia gpu	❌ only nvidia gpu
gpu sharing	✅(software definition)	✅(timeslice + mps + mig)	✅(timeslice + mps + mig)
Observability	HAMi metrics + DCGM exporter	DCGM exporter	DCGM exporter
cuda support matrix	**> 10.2**	**unknown**	**unknown**
os support matrix	**>= 3.10**	**unknown**	**unknown**
Kubernetes support matrix	**>= 1.16**	**>= 1.10**	**>= 1.26**
ARCH	K8s scheduler-plugin device-plugin	device-plugin, no scheduler	DRA
Task priority	✅	❌	❌
GPU Core Over subscription	✅	❌	❌
GPU Memory Over subscription	✅(By CUDA Unified Memory)	❌	❌
Node level spread/binpack	✅	❌	❌
GPU level spread/binpack	✅	❌	❌
Deployment method	Helm	Helm	Helm

GPU 虚拟化不同实现方式

Key features	HAMi vgpu	CUDA Streams	MPS	Time-slicing	MIG	Nvidia vGPU
Target Use Cases	The same cluster contains multiple heterogeneous AI devices+ Gpu sharing + flexible scheduler policies	Optimized for concurrencywithin a single application	When running multipleapplications in parallel butcan deal with limitedresiliency	When running multipleapplications that are notlatency-sensitive or cantolerate jitter	When running multipleapplications in parallel butneed resiliencyand QoS	When needing to supportmulti-tenancy on the GPUthrough virtualization
Partition Type	Logical	Single Process	Logical	Temporal	Physical	Temporal & Physical (VM)
Max Partitions	Unlimited	Unlimited	48	Unlimited	7	Variable
SM Performance Isolation	Yes(by % not per client)	No	Yes(by % not per client)	Yes	Yes	Yes
Memory Protection	Yes	No	Yes	Yes	Yes	Yes
Memory Bandwidth QoS	No	No	No	No	Yes	Yes
Error Isolation	Yes	No	No	Yes	Yes	Yes
Cross-PartitionInteroperability		Always	IPC	Limited IPC	Limited IPC	No
Reconfiguration	At process Launch	Dynamic	At process Launch	Time-Slice Duration Only	When Idle	No
Telemetry	Yes	No	Limited	No	Yes(including in containers)	Yes(including live migration)
Other noteworthy	Supports all GPUs, open source		cudaCapability >= 3.5	cudaCapability >= 7.0	cuda capability >= 8.0 Hopper,Ampere	license required

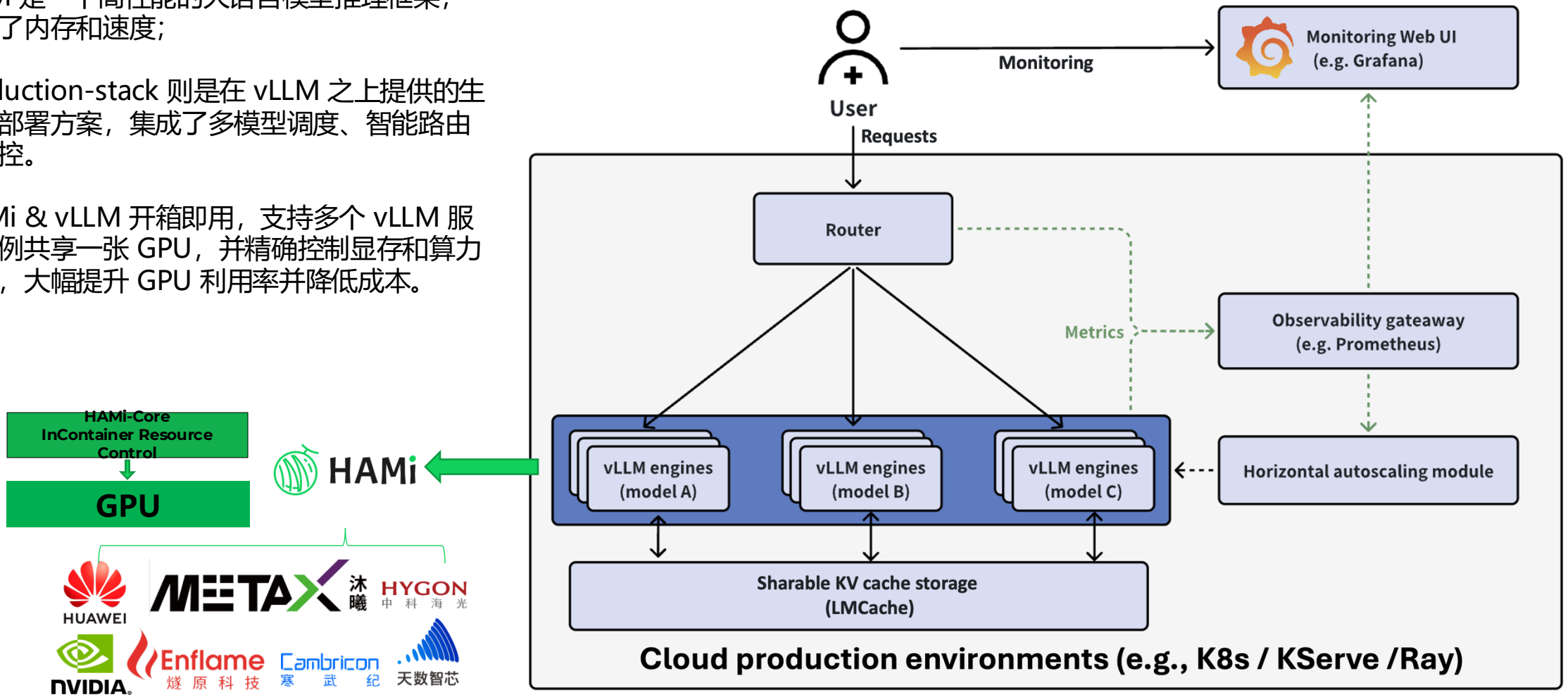
- Nvidia MIG 隔离程度最高(总线物理级别)，80G 最多虚拟 7 个 10g 实例，物理损耗
- MPS 是一个不错的选择，Nvidia 官方软件定义 vGPU，容错低
- 国内公有云 vGPU 都是 Kernel 层的拦截(自提供 OS, 有专门的 Kernel 团队，维护成本、代价更高)
- HAMi CUDA 层拦截, **5%** 以内性能损失，OS 、Kernel 无关，普适性更强

HAMi & vLLM 无缝结合

vLLM 是一个高性能的大语言模型推理框架，优化了内存和速度；

Production-stack 则是在 vLLM 之上提供的生产级部署方案，集成了多模型调度、智能路由和监控。

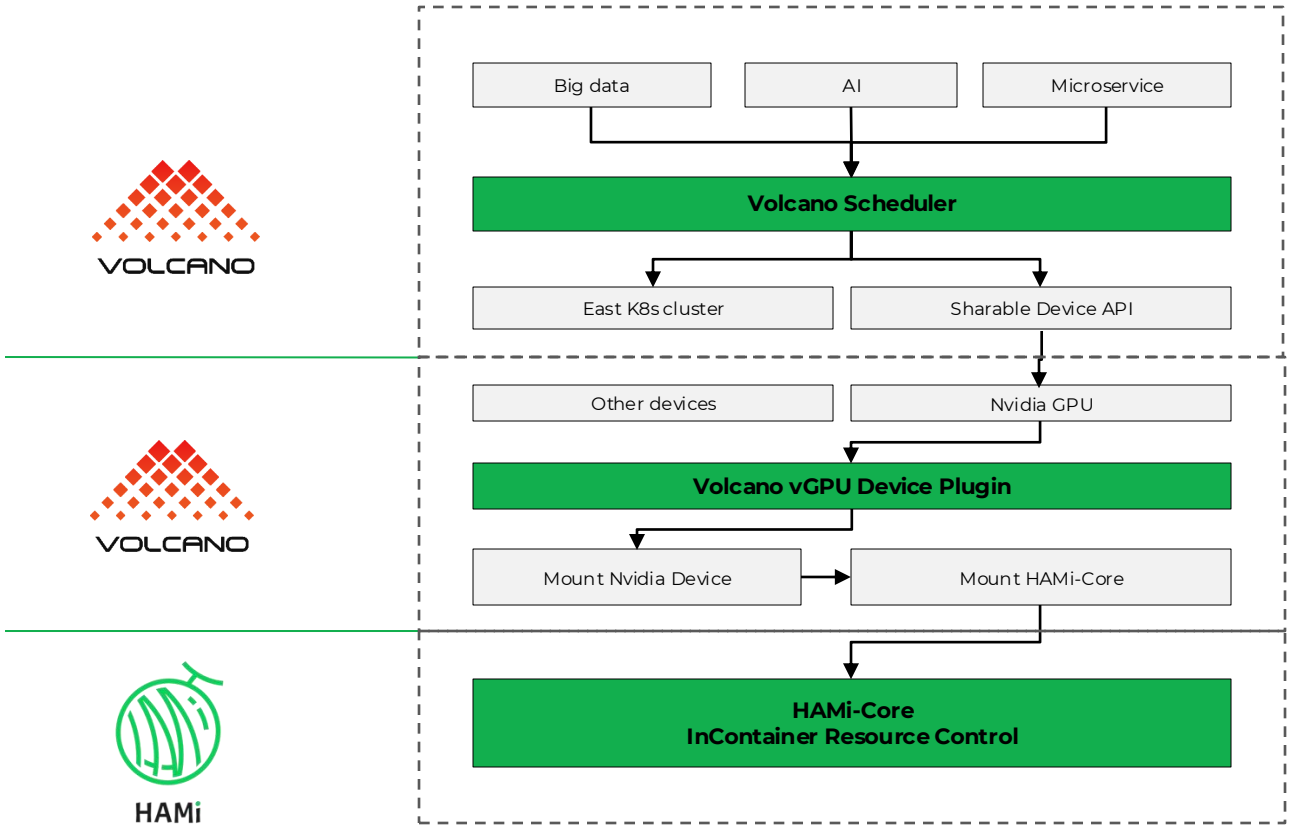
HAMi & vLLM 开箱即用，支持多个 vLLM 服务实例共享一张 GPU，并精确控制显存和算力分配，大幅提升 GPU 利用率并降低成本。



HAMi & Volcano 无缝结合

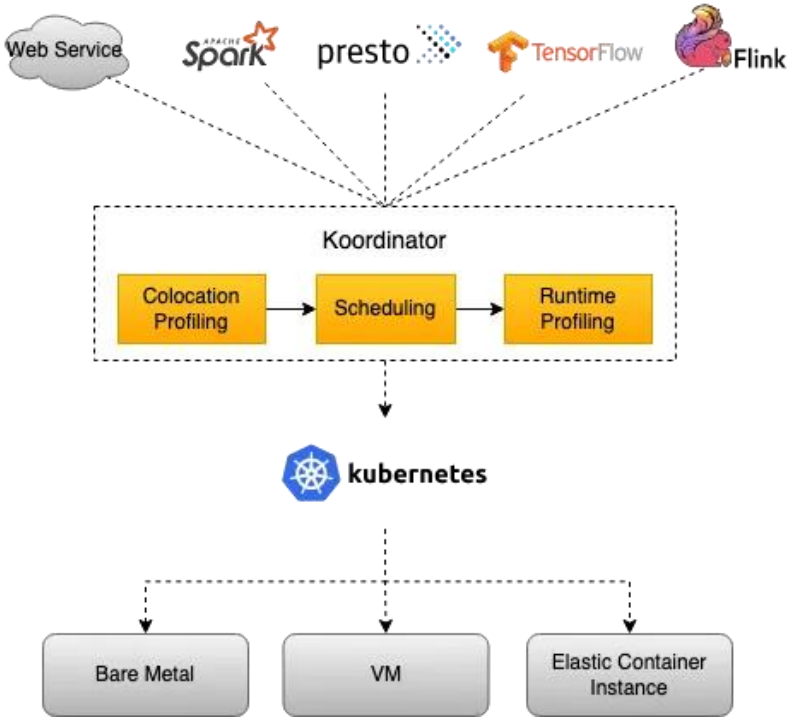
Volcano 是 CNCF 下首个也是唯一的基于 Kubernetes 的容器批量计算平台，主要用于高性能计算场景，提供了机器学习、深度学习、生物信息学、基因组学及其他大数据应用所需要而 Kubernetes 当前缺失的一系列特性。 开源后，吸引了来自 AWS、OpenAI、腾讯、百度、爱奇艺、小红书、滴滴、Vivo、趣头条等多家公司的参与和贡献。 目前，容器批量计算解决方案已经在社交资讯、基因测序、在线教育、视频、电商等行业广泛使用。

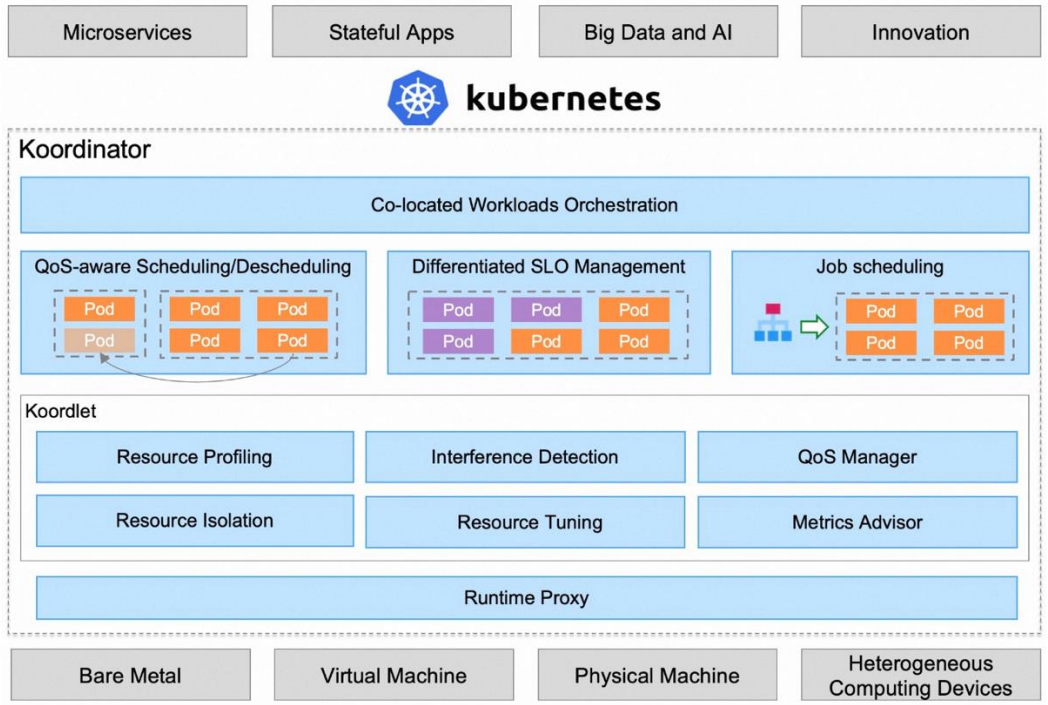
目前 HAMi 作为 Volcano 底层 GPU 虚拟化技术的提供者，用户可以无缝享受两者加成的能力



HAMi & Koordinator 无缝结合

Koordinator 是一个基于 Kubernetes 的 QoS 感知调度系统，能够有效提升集群资源利用率，并针对不同优先级的应用提供 QoS 保障能力，避免应用因资源竞争而影响性能表现。 在提升集群资源利用率的同时，保障Service应用的服务质量，适用于批处理、高性能计算、AI 任务和机器学习等业务场景。由阿里开源，开源以来 吸引了包括阿里巴巴、蚂蚁科技、Intel、小红书、小米、爱奇艺、360、有赞等众多企业的优秀工程师参与



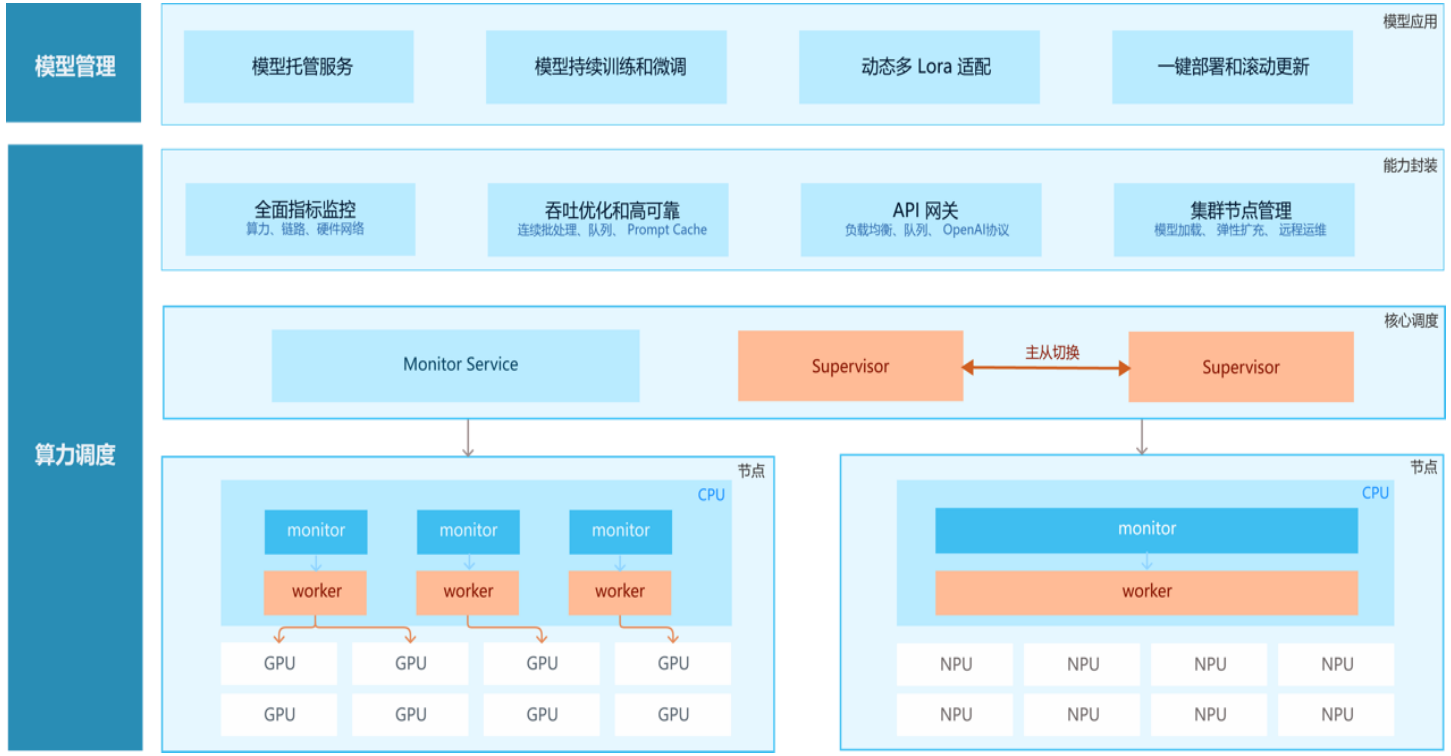


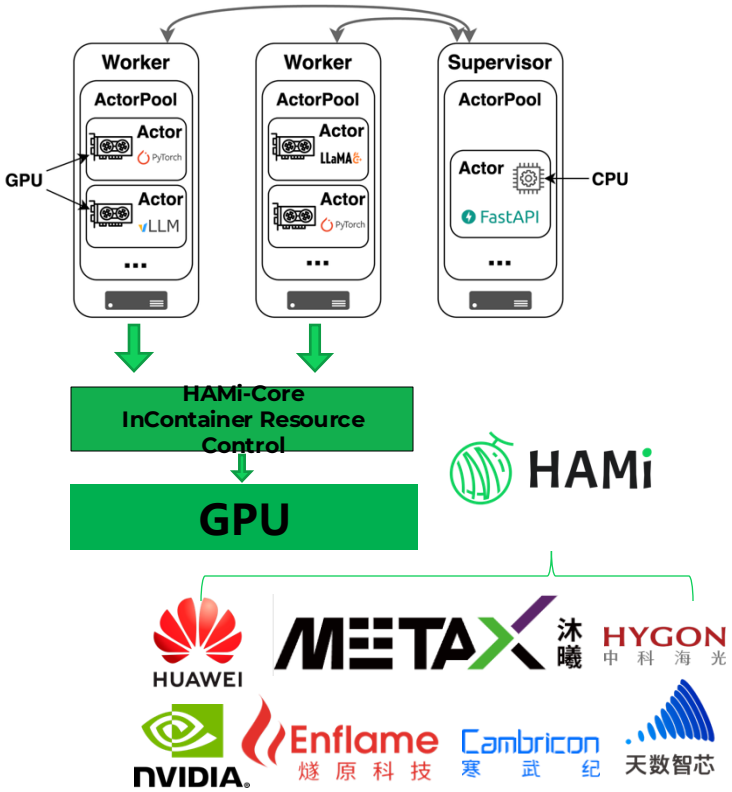
Koordinator GPU

虚拟化内置技术方案

HAMi & Xinference 无缝结合

Xinference（Xorbits Inference） 是一个开源 AI 推理平台，基于 Kubernetes 和本地环境，旨在通过统一的 API 接口支持 LLM、嵌入、多模态模型的高效部署与调用。它能够提高集群资源利用率，为关键 AI 服务提供稳定的 QoS 保障，避免因资源争夺造成性能波动。适用于对话生成、嵌入计算、图像/音频/视频生成等不同优先级与类型的 AI 任务。自开源以来，已有大批开发者和工程师参与社区建设，形成了丰富的部署和使用生态。



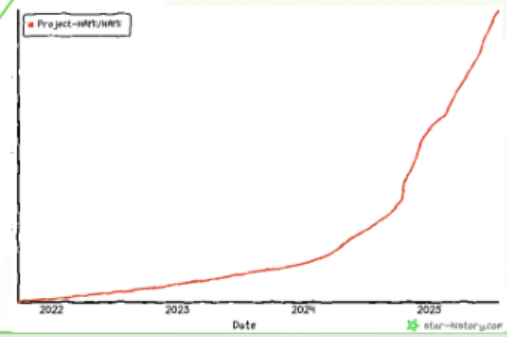


快速发展的生态

PPIO 派欧云
中国东信 China-ASEAN Information Harbor
ZStack 云轴科技
招商银行 CHINA MERCHANTS BANK
UCLOUD 优刻得
贝壳
马上消费 WWW.MSXF.COM
思爱普 SAP
4Paradigm 第四范式
基石智算
KYLINSOFT 麒麟软件
TECHNOLOGY 顺丰科技
Technische Universität München
UnionPay 银联
新浪微博 weibo.com
LOTUS ROBOTICS
HAMi
CNCF 唯一
科大讯飞 iFLYTEK
同程旅行
DaoCloud
DEEPROUTE.Ai 元戎启行

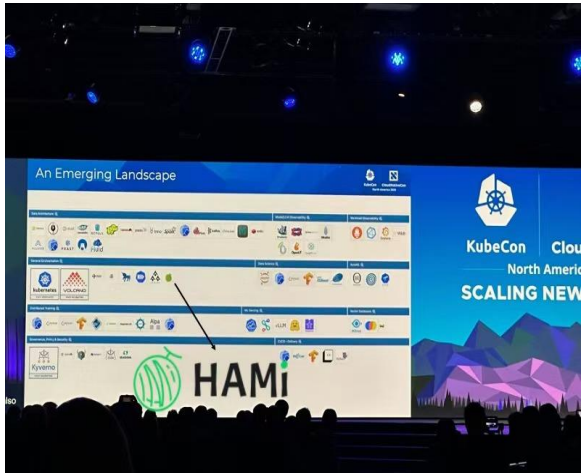
15 个国家 350+ 贡献者

250+ 用户(银行、金融，30+ 云厂商)



推广全球行

2024,2025多次受邀在中国、英国法国、香港、日本全球各地 AI 专业会议分享



PART 3

HAMi 适用场景

场景1：在离线推理服务共享有限资源

显著降低在线推理服务所占用的 GPU 卡的成本，提高 GPU 利用率

高低频在线推理方案详解

1. 高频任务队列使用高端卡和更大的vGPU资源池；
2. 加大高频任务队列GPU资源的弹性配额，在低频任务队列合理配置GPU显存超配；
3. 配置高频任务Pod支持HPA水平扩缩容；
4. 高低频业务带宽限制（IPVLAN+RoCE直通）；

高低频推理业务弹性配额

高低频业务分别部署在不同的 NS

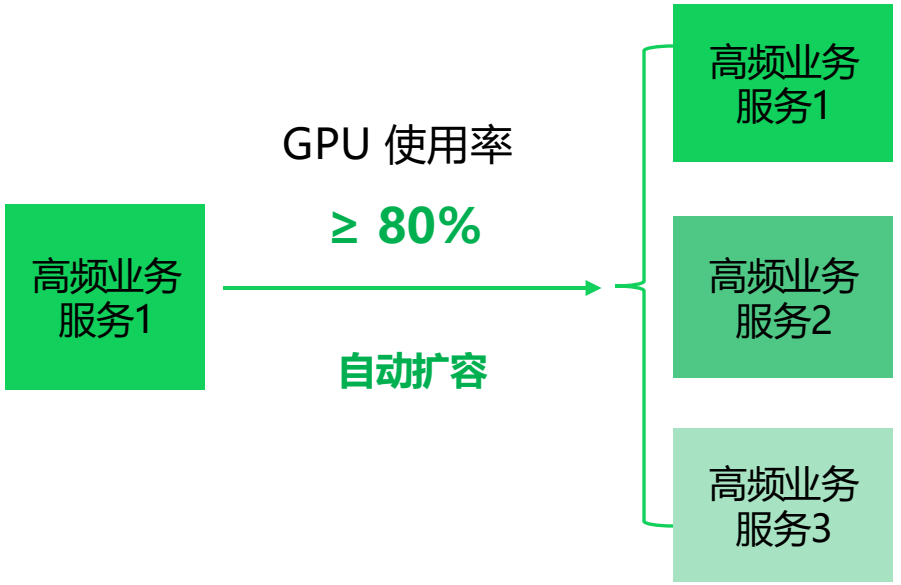
NS QuotaA (min:4, max:6)
NS QuotaB (min:6, max:8)

UserA 使用 4GPU (达到 min 值)， UserB 使用 3GPU，此时集群资源丰富，调度无任何异常

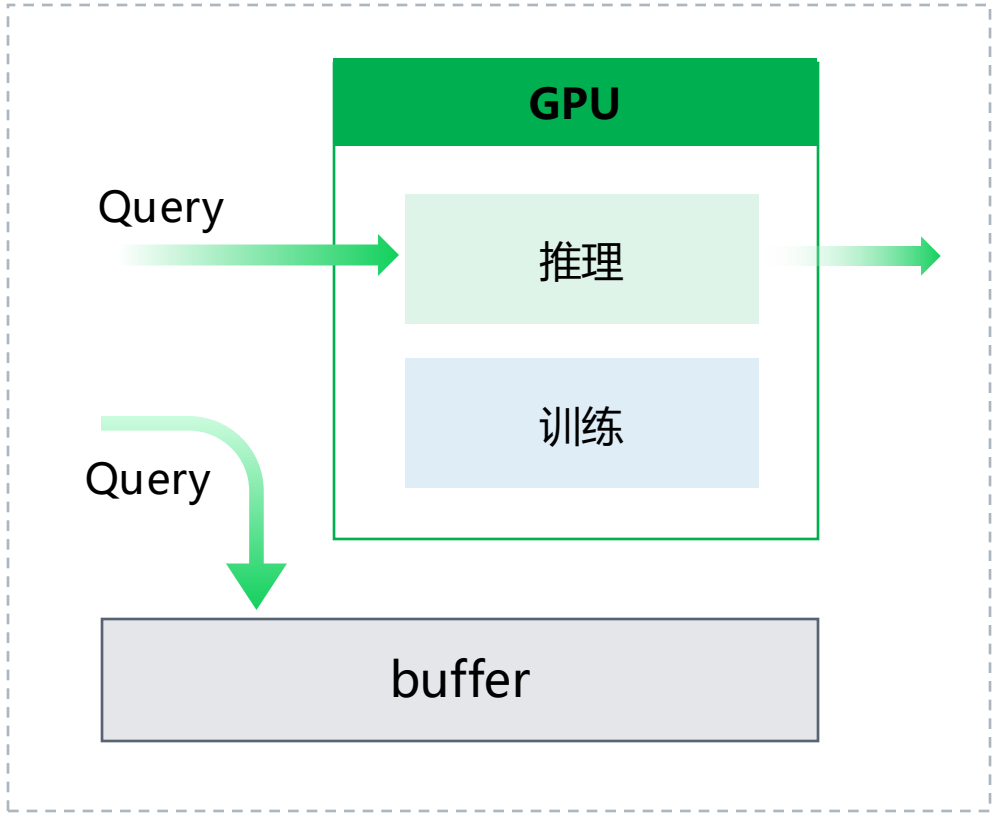
UserA 最大使用 6GPU (达到 max 值)， UserB 使用 3GPU，此时集群调度仍无任何异常

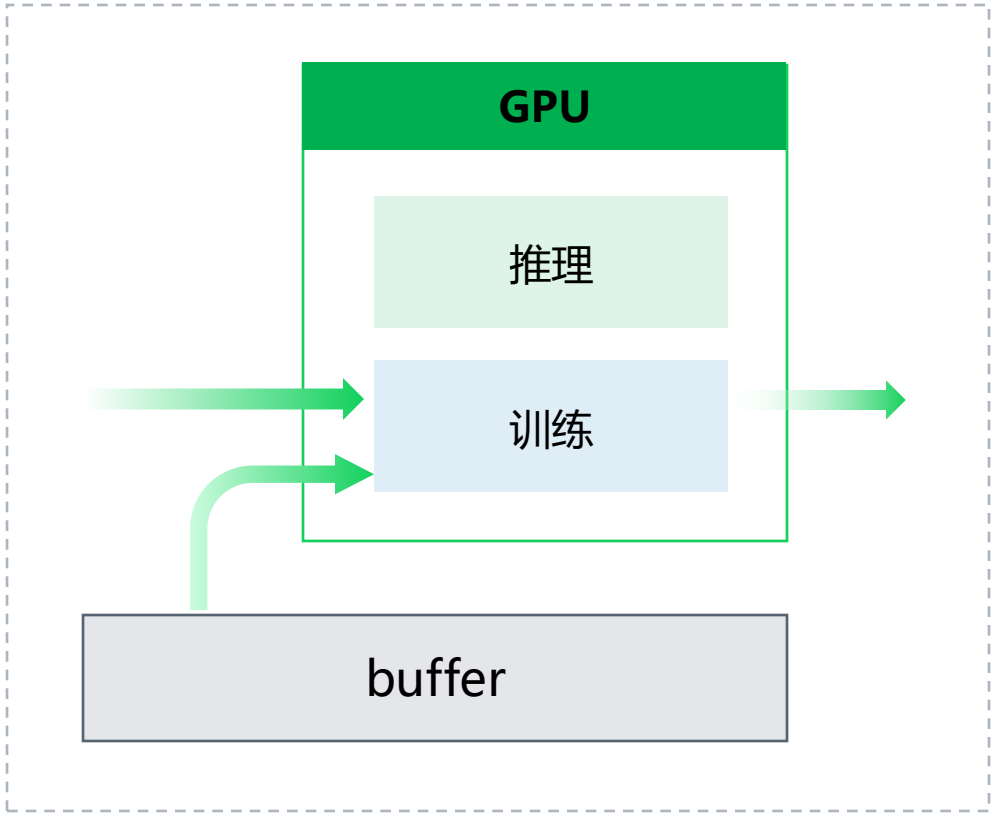
UserB 需要继续使用 3块 GPU, 将触发驱逐、抢占，UserA 需要资源归还，确保返回给 UserB 足够资源

高频业务弹性扩缩容



场景2：训推混部

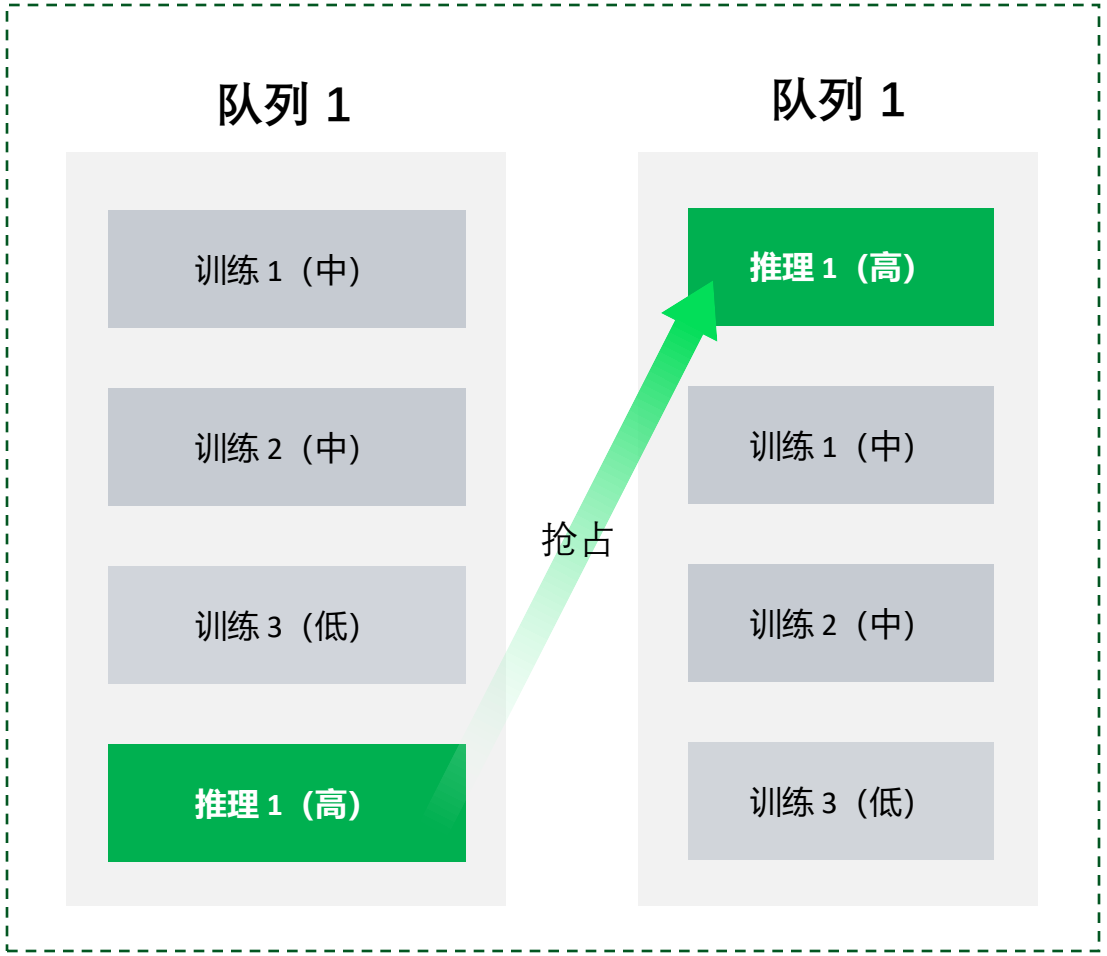




当推理有请求时，HAMi 将会暂时缓存训练任务的请求，将算力留给推理任务

推理任务没有请求时，HAMi 将缓存的训练请求交给 GPU 执行，不浪费算力

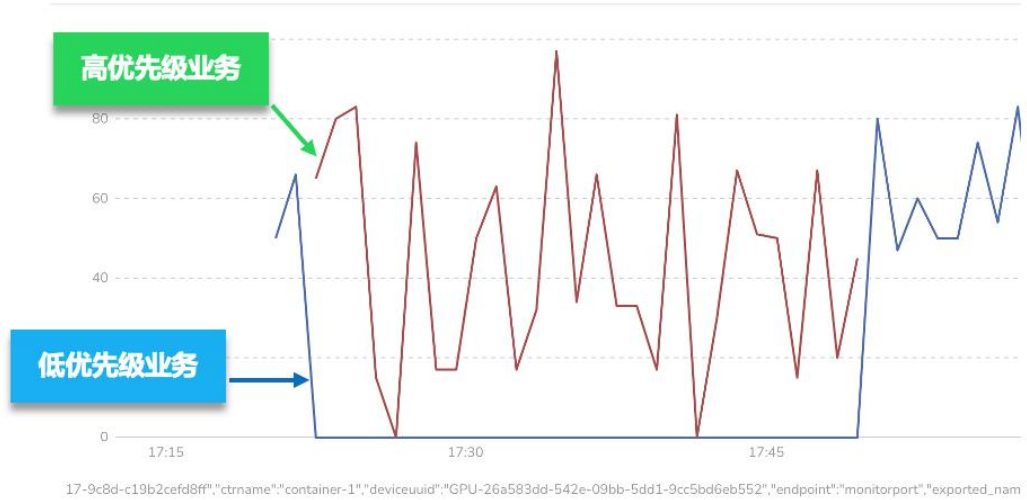
训推混部方案一



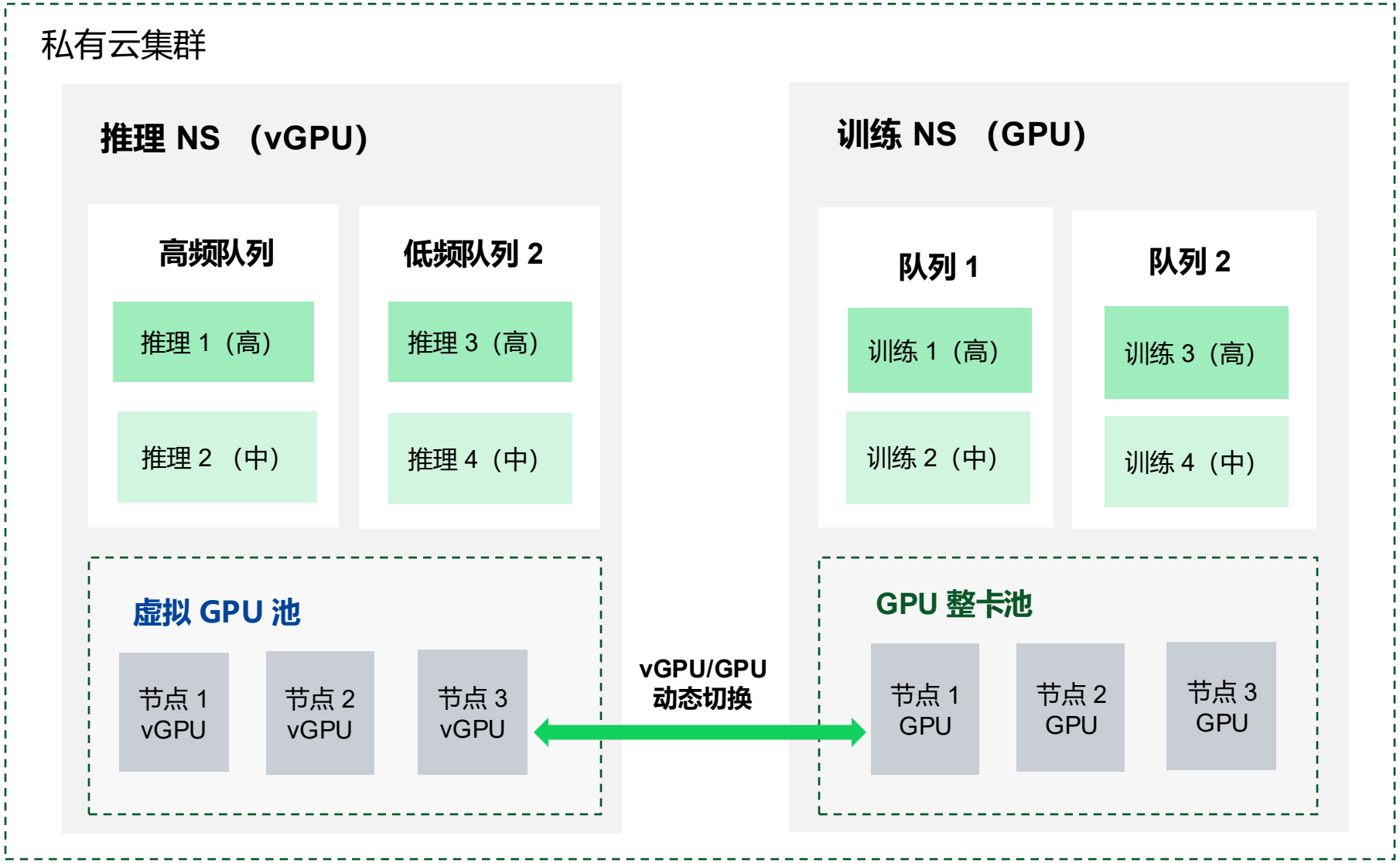
1. 推理基本是在线业务，实时性要求高；训练基本是离线业务，实时性要求低，优先级比推理低。

2. 训推混部时，可根据业务创建队列，配置GPU资源限额进行排队，推理优先级最高。

3. 当队列中出现 优先级高的任务，可进行抢占（或暂停低优先级任务的资源申请），优先完成推理任务。

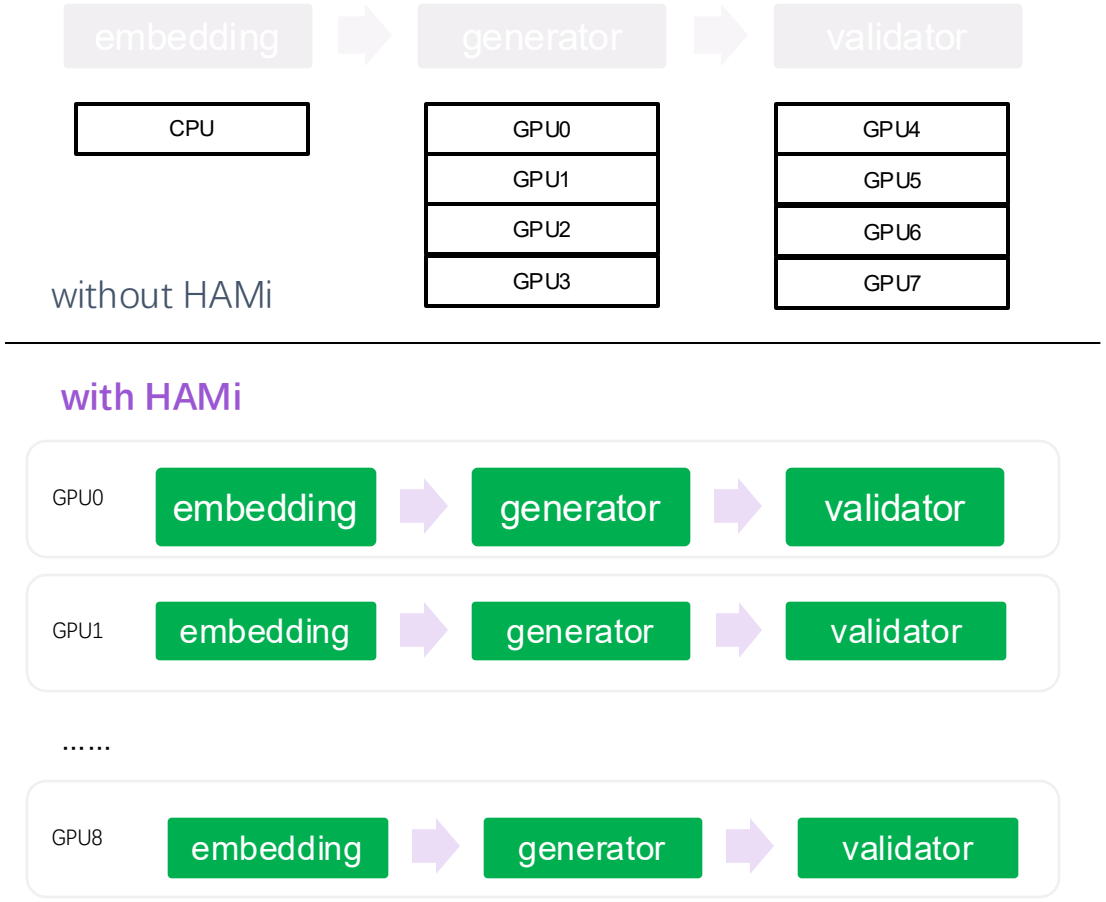


训推混部方案二



1. 为推理和训练创建不同的NS，设置不同的Quota
2. 白天虚拟 GPU 池动态扩大，用于支持推理业务
3. 晚上GPU 整卡池扩大，用于支持训练任务

场景3：提升 LLM 推理能力



Generally speaking, a LLM product does not only contain a generator, but consists several 'small' models. Taking product [Shishuo] from 4th paradigm for example.

Without HAMi:

- Embedding model in CPU
- Support 4 threads

With HAMi

- All parts in GPU
- Support 8 threads
- Compatible with mainstream large model inference frameworks such as **TGI**, **FastLLM**, and **vLLM**
- The inference performance is improved by about 1-8 times.

PART 4

用户案例

案例1：某头部银行 - HAMi 提高 GPU 利用率 40%

用户痛点：

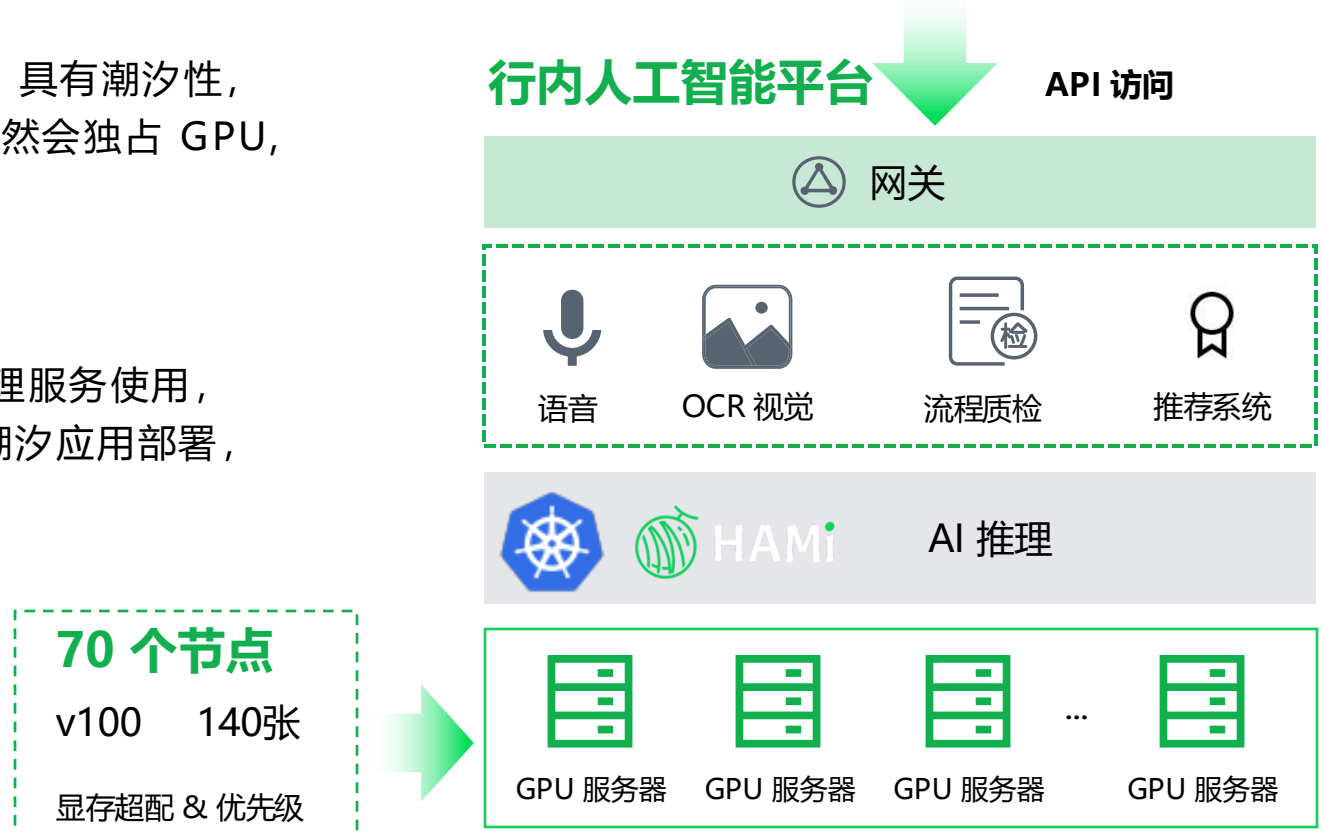
运行大量的中小型模型推理类应用 (OCR、搜索、推荐)，具有潮汐性，
默认一张 GPU 仅分配给一个推理任务，推理服务空闲任然会独占 GPU，
真实利用率不到 20%

解决方案：

采用 HAMi vGPU 技术，一张物理 GPU 同时给多个推理服务使用，
同时使用 GPU 显存能力与 任务优先级抢占机制，更多潮汐应用部署，
同时 QOS 得到保证。

客户收益：

- GPU动态分配，利用率获得提升；
- 支持GPU虚拟化，使用策略上更加灵活；
- 显存超配，让更多潮汐应用同时使用一张 GPU
- GPU 利用率从 20% → 70% (70 服务器，140卡)



案例2：某证券公司 - HAMi 解决人多卡少使用问题

用户痛点：

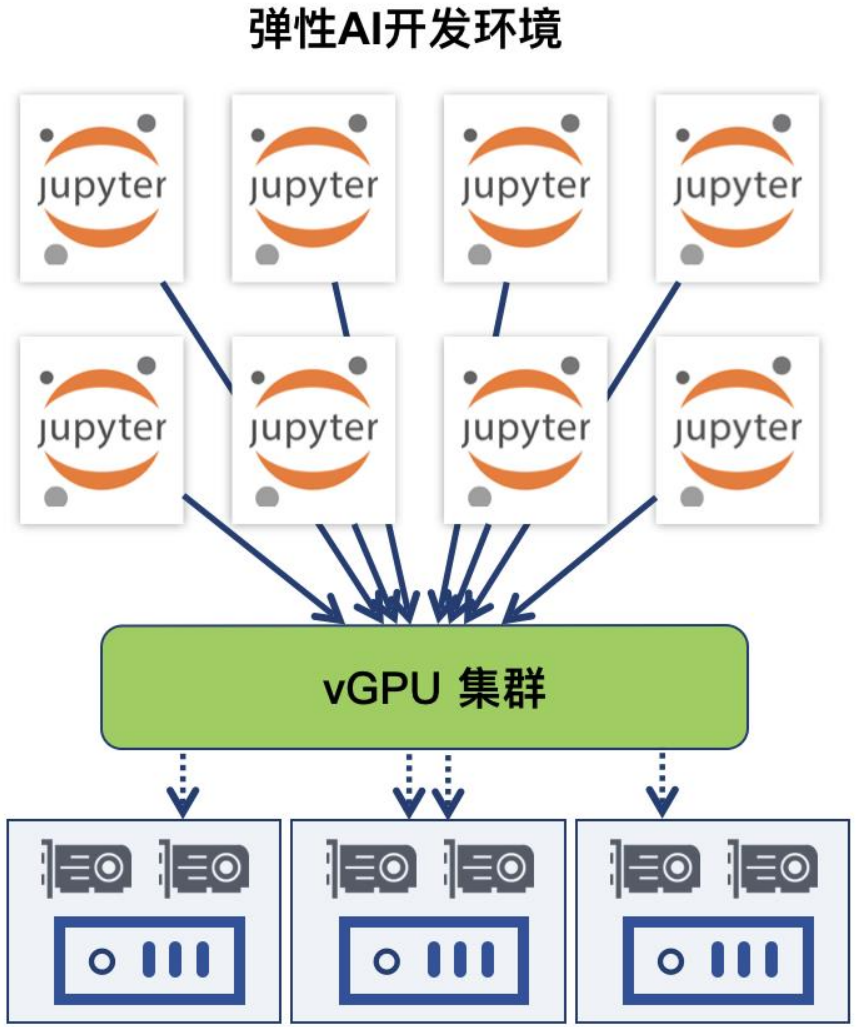
目前公司搭建了一套kubeflow平台，供算法工程师使用。但是默认情况下一张显卡只能分配给一个notebook，并且如果notebook不停止/销毁的话，会一直独占这张显卡,真实使用率不到 20%

解决方案：

采用 HAMi vGPU 技术，可以使一张物理显卡同时让多个算法工程师使用，提升工作效率。

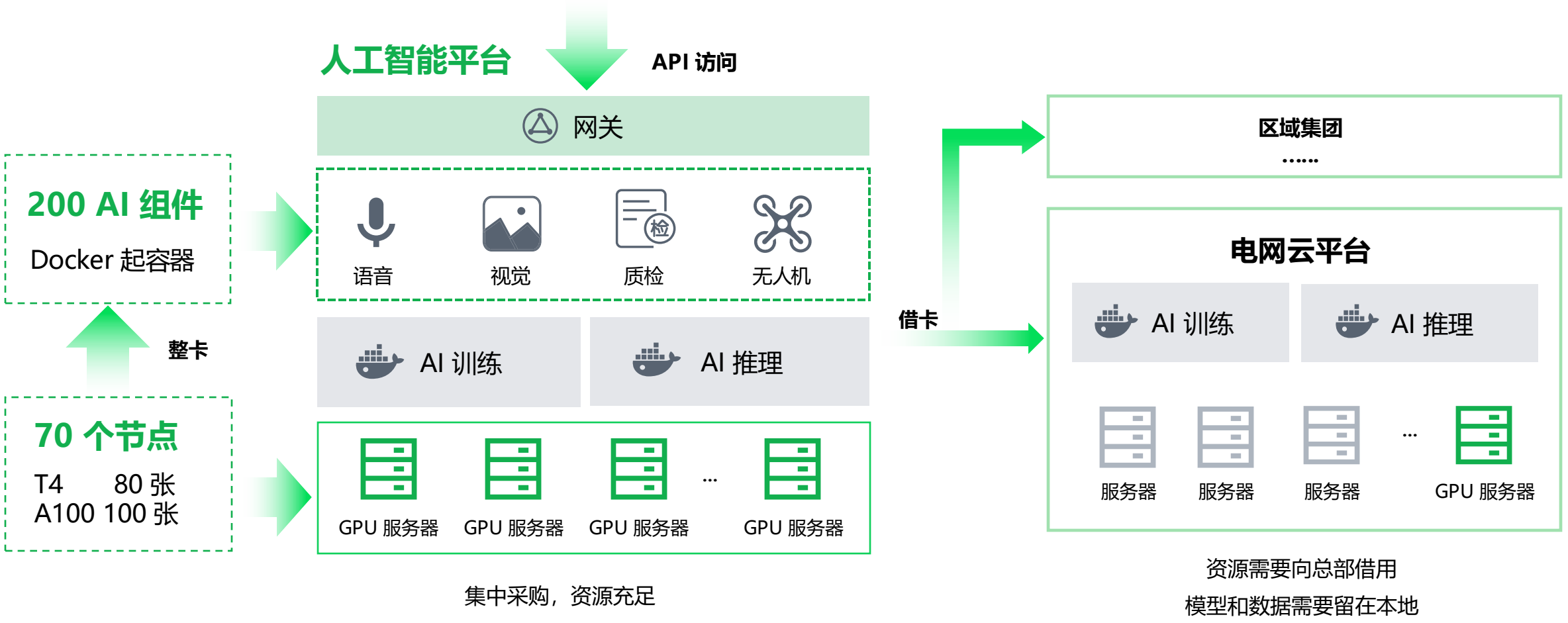
客户收益：

- GPU变为动态分配，使用效率获得提升；
- 支持GPU虚拟化，使用策略上更加灵活；
- 打通开发环境和训练任务过程，算法工程师不必困扰于资源分配的过程，可以更专注于业务的研发。



案例3：某电网集团 - GPU 池化与集中式高效运维

1. 人工智能平台由某电网总部承建，GPU 资源充足
2. AI 组件由下属地州区域供电局提供



案例4：某电信运营商平台 - GPU 虚拟化与异构设备管理

用户需求：

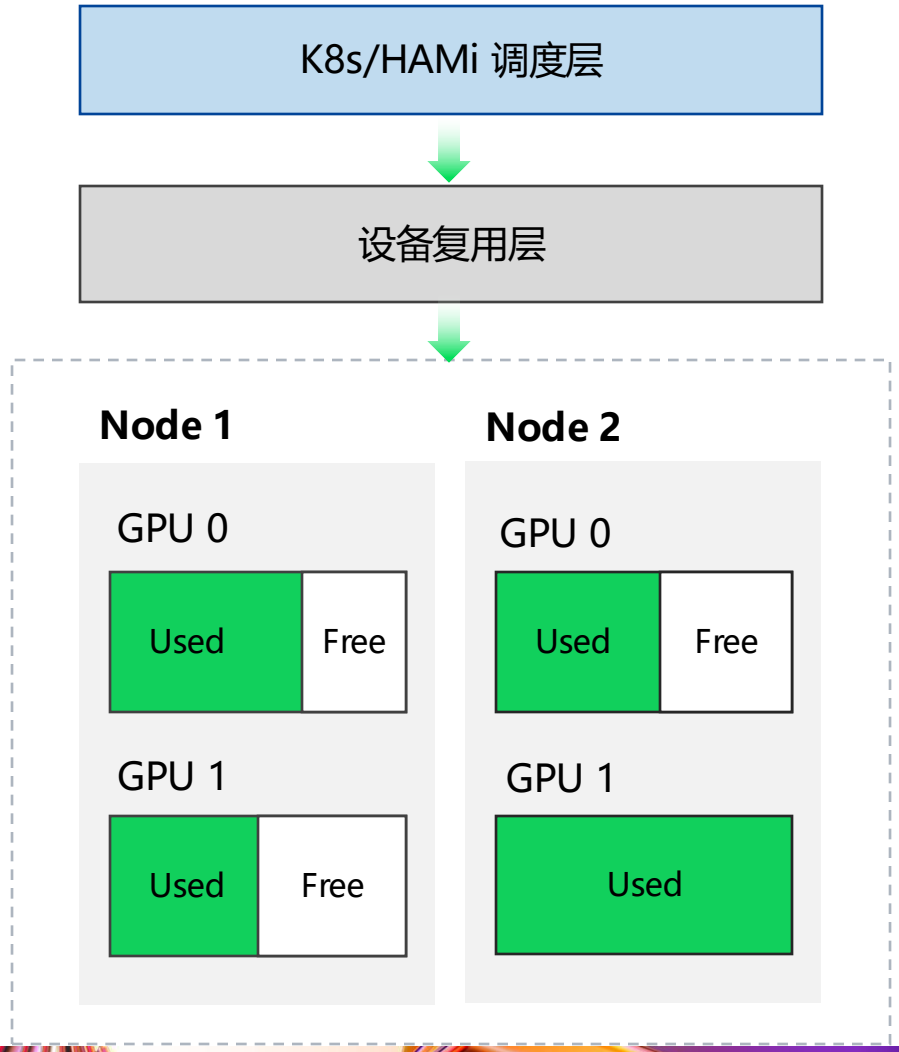
此案例为商业化案例，目前公司搭建了k8s平台包含了多种终端设备，不同架构的服务器，用于识别网络运营业务中的各类场景，设备识别、安全作业、违规检测等等，需要建设一套低使用门槛的平台，可以方便的进行模型选择，任务分发，结果收集等，并有设备复用的需求。

客户收益：

- 降低用户使用门槛
- 统一抽象不同体系架构下的和加速卡

解决方案：

通过 HAMi，可以将不同的设备在用户层做统一抽象，用户不需要关心不同设备使用上的差异，这些差异化的工作会由平台完成，并且提供设备复用的功能



案例5：算力云租赁 GPU 场景

解决方案：

使用 HAMi，进行 GPU 虚拟化 的售卖服务，用户付费意愿提升，并获得更高收入和利润空间

客户收益：

使用前：

H800 整卡 80G 显存 **16 元/时**，无法拆分售卖，显存资源使用率低

使用后：

通过 vGPU 模式可以进行细颗粒度的资源售卖，H800 卡 3G 显存仅 **1.89** 元/时，提高用户的付费意愿

从收益考量，1 张 H800 可同时支持 **26** 位用户以 3G 显存/用户进行租赁使用，同比收入增长超 **3.15 倍**

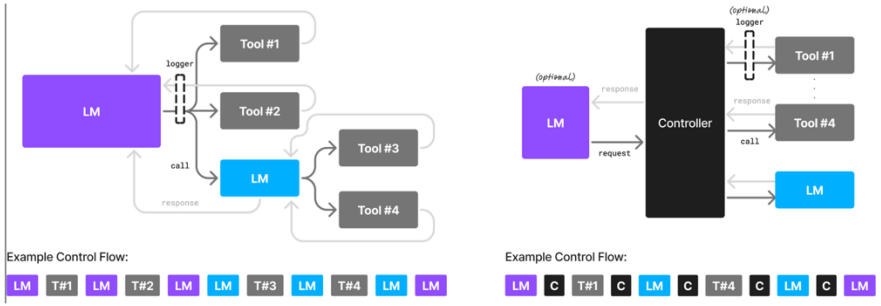
PART 5

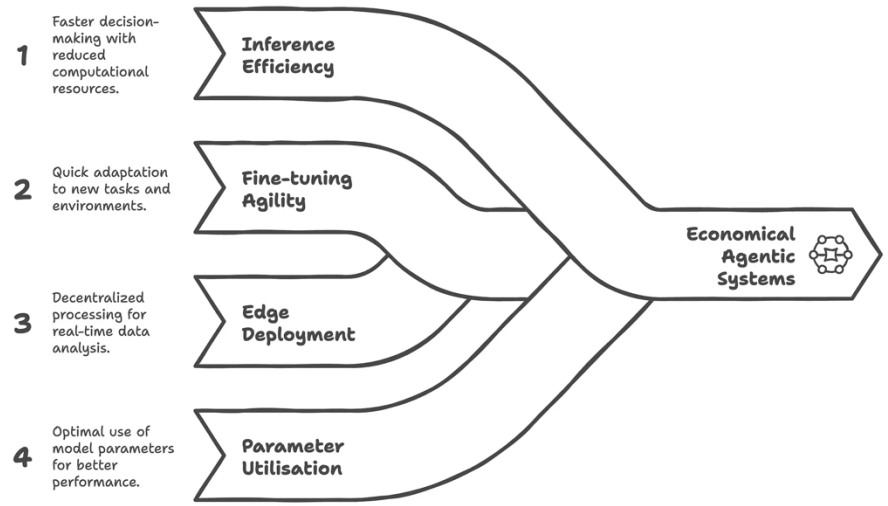
未来展望

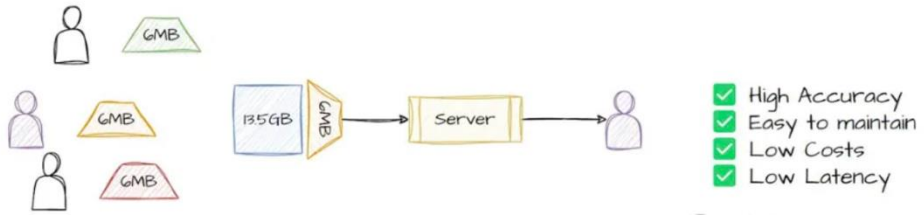
Small Language Models are the Future of Agentic AI

Small Language Models are the Future of Agentic AI

Peter Belcak[1] **Greg Heinrich**[1] **Shizhe Diao**[1] **Yonggan Fu**[1] **Xin Dong**[1]
Saurav Muralidharan[1] **Yingyan Celine Lin**[1,2] **Pavlo Molchanov**[1]
[1]NVIDIA Research [2]Georgia Institute of Technology
agents-research@nvidia.com







在Agent任务中，大语言模型经常处理重复、专业化的子任务，这让它们**消耗大量计算资源，且成本高、效率低、灵活性差。**

相比之下，小语言模型则能在性能够用的前提下，让 Agent任务的执行变得更加经济灵活。

https://arxiv.org/pdf/2506.02153

算力即国力，异构算力必然成为主流

NVIDIA

商业 + 战略 双驱动

intel DATA CENTER GPU

Ascend

Cambricon 寒武纪

HYGON

Enflame 燧原科技

摩尔线程 MOORE THREADS

METAX 沐曦

AMD

天数智芯 Iluvatar CoreX

昆仑芯 KUNLUNXIN

感谢聆听

github，欢迎关注 star，贡献

科技生态圈峰会 + 深度研习

——1000 + 技术团队的共同选择

KEYLINK ing

AiDD 峰会

AiDD峰会 上海站
AI+研发数字峰会
时间：2026.05.22-23

AiDD峰会 北京站
AI+研发数字峰会
时间：2026.08.21-22

AiDD峰会 深圳站
AI+研发数字峰会
时间：2026.11.20-21

AiDD峰会详情

AiDD × AI+产品峰会

AiDD × 上海站
AI+产品创新峰会
时间：2026.01.16-17

AiDD × 北京站
AI+产品创新峰会
时间：2026.08.23-24

产品峰会详情

AiDD × K

2026 AI+ PRODUCT INNOVATION SUMMIT

01.16-17 · ShangHai

AI+产品创新峰会

Track 1: AI 产品战略与创新设计

从0到1的AI原生产品构建

论坛1: AI时代的用户洞察与需求发现

论坛2: AI原生产品战路与商业模式重构

论坛3: AgenticAI产品创新与交互设计

Track 2: AI 产品开发与工程实践

从1到10的工程化落地实践

论坛1: 面向Agent智能体的产品开发

论坛2: 具身智能与AI硬件产品

论坛3: AI产品出海与本地化开发

Track 3: AI 产品运营与智能演化

从10到100的AI产品运营

论坛1: AI赋能产品运营与增长黑客

论坛2: AI产品的数据飞轮与智能演化

论坛3: 行业爆款AI产品案例拆解

2-hour Speech: 回归本质

用户洞察的第一性

——2小时思维与方法论工作坊

在数字爆炸、AI迅速发展的时代，仍然考验“看见”的“同理心”

Panel 1: 出海前瞻

“出海避坑地图”圆桌对话

——不止于翻译:

AI时代的出海新范式

Panel 2: 失败复盘

为什么很多AI产品“叫好不叫座”?

——从伪需求到真价值:

AI产品商业化落地的关键挑战

智能重构产品 数据驱动增长

Reinventing Products with Intelligence, Driven by Data

80+	40+	10+	800+
业界大咖	创新案例	分论坛	听众规模
汇聚产品大咖的真知灼见	开拓全新AI+产品视角	涵盖产品到商业增长的全链路	共同探索产品的智能重构

扫码查看会议详情

第8届AI+研发数字峰会
AI+ Development Digital Summit

感谢聆听！

扫码领取会议PPT资料