

从开源到商业化 云原生架构下大模型的大规模推理产品化

YeTing - DaoCloud

Content 目录

- 01** 业务背景和挑战
- 02** 云原生化的 SaaS 平台介绍
- 03** 开源技术的力量
- 04** 未来规划



Part 01

业务背景和挑战



产品销售形态



MaaS 服务

Pay for Tokens

模型部署

Pay for Instances

模型训练/微调

开发机

大模型推理的“三高”问题 - MaaS



大模型推理的“三高”问题 - MaaS

01

高并发需求

- 扩容算力资源
- 限流
- KV Cache

02

推理性能问题

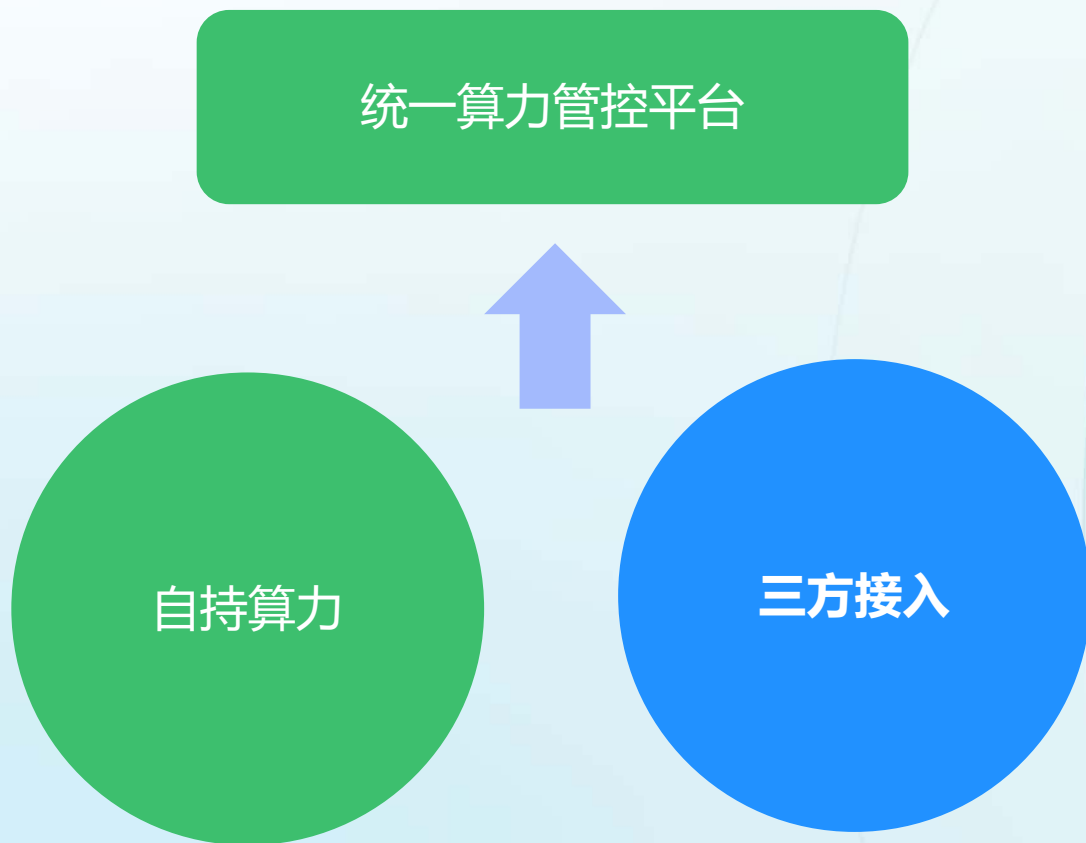
- runtime 的选择
- vllm
- sglang

03

成本问题

- 😂 短时间无法解决
- 可以按照GPU运行成本动态进行定价

算力资源接入方式

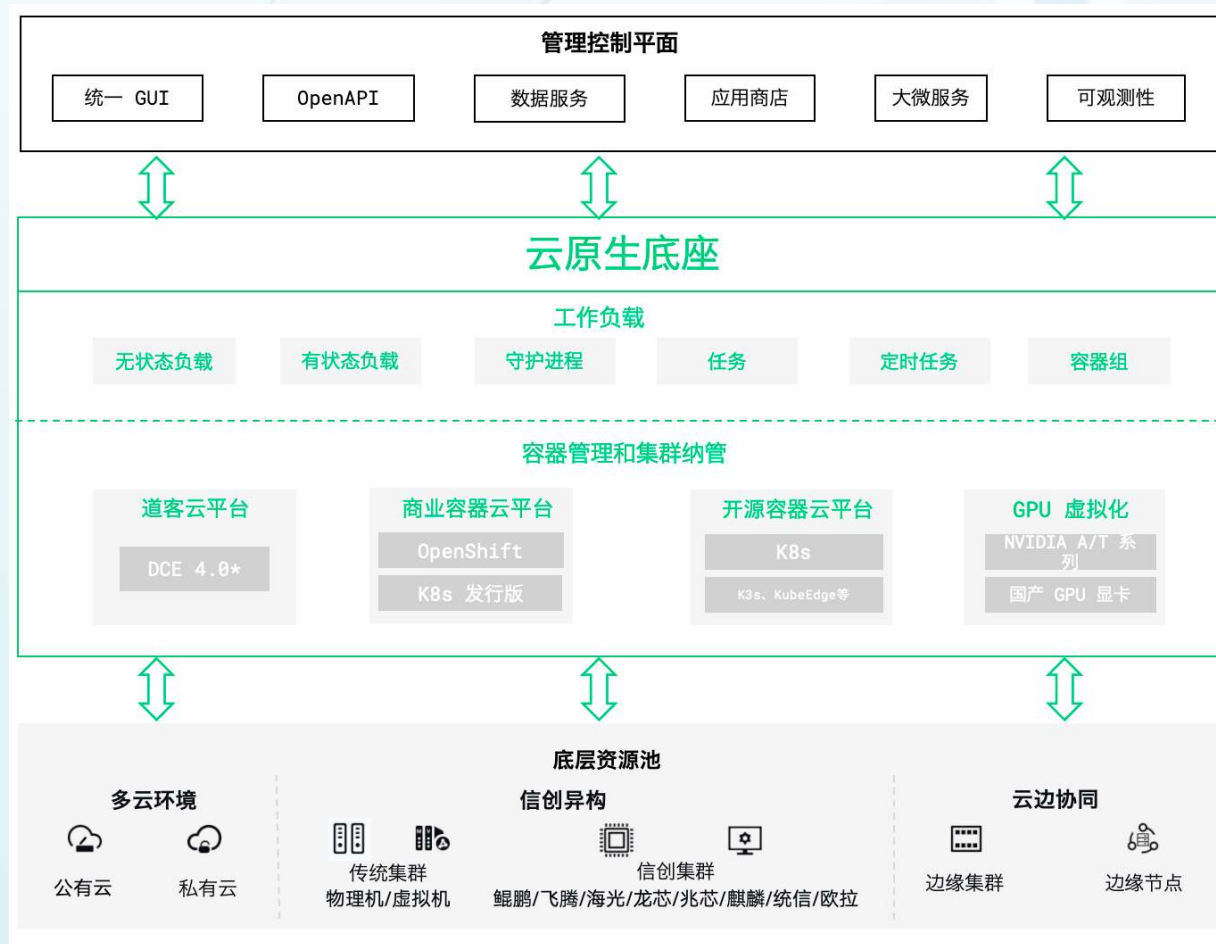


- 借助云原生提供的技术底座，我们实现统一的算力管控平台，支持大量接入算力资源
- 主要接入的算力资源主要以两种为主
- 自持算力
- 合作企业提供算力租赁

算力接入方式 - 统一管控平台

自研产品 DCE 云原生操作系统的基座能力；天然具备了多集群纳管的能力；使得纳管算力集群的操作成本非常低。

- 通过 kubeconfig 即可快速接入算力集群
- 支持表单化的集群创建能力
- 可自定义安装 Addon，GPU驱动、管理模块的 Agent 全自动安装
- 提供完整的多集群中心化观测组件能力



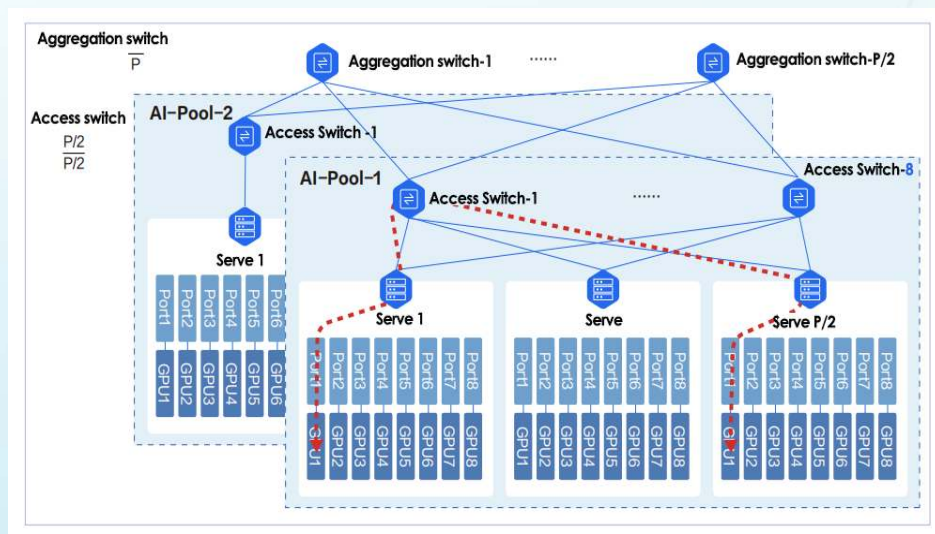
算力接入方式 - 三方接入的难题？

- 卡不同（异构） >>> 统一的算力资源池化（DCE 适配主流 GPU，可自适应管理）
- 地域不同（调度复杂性） >>> Kueue 的调度
- 稳定性（当三方算力集群出现波动时，如果保证已有用的资源稳定）
 - 可靠的算力供应合作选择（严选机制）
 - 产品在设计时就考虑不稳定的备份策略
- 监控运维 >> **KCover** 故障自恢复（掉卡续训）
- 接入成本高 >> 标准化的产品接入方式

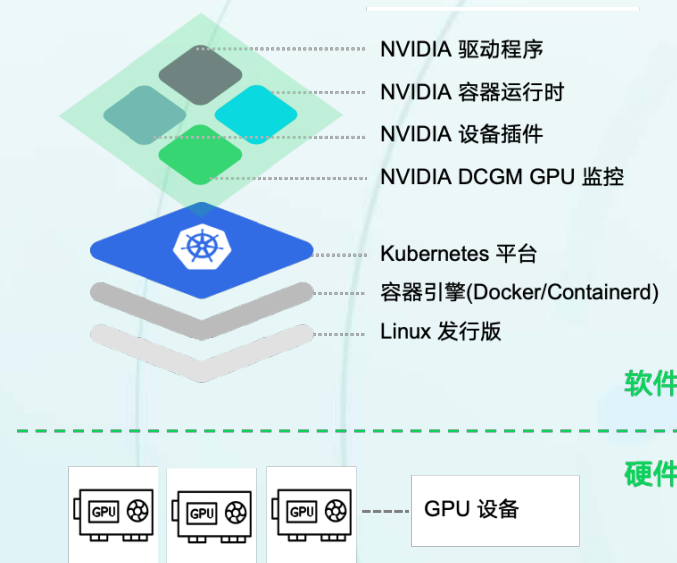
统一初始化为算力集群（**Kubernetes**），使用标准的集群方式纳管

GPU 算力集群的特殊性

相较于传统集群的架构，组建一个大规模的 **GPU 算力集群**，需要从底层设计考虑整个集群的架构设计。



算力网络
架构



驱动和插件
依赖

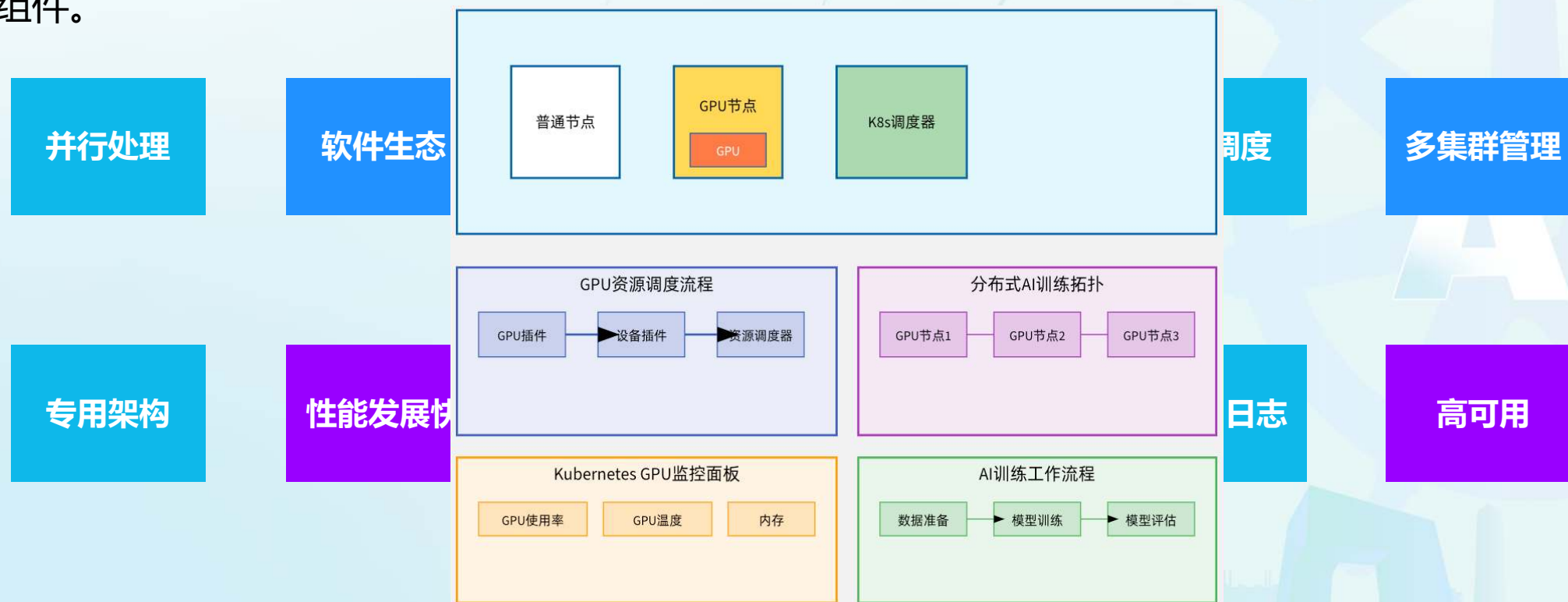


特殊调度策
略

Why Kubernetes 是最适合的算力集群基座 ？

GPU 凭借其并行计算能力和专用架构,在AI训练和推理中发挥关键作用,显著提升性能和效率,推动AI技术快速发展,成为现代AI基础设施的核心组件。

Kubernetes 作为领先的开源容器编排平台,在容器化部署和自动化调度优化能力,可以高效管理 GPU 等昂贵的计算资源。



高成本问题

- 算力资源纳管成本，底座环境经常变化，如果避免对用户的应用
- 运维成本
- 算力成本
- GPU 动态拆分 + 统一调度 (HAMi) / (Kueue)
- 一键接入算力集群 (todo 80%)

AI

高复杂性（模型/框架异构）

- 算力资源需要考虑异构问题
- 模型种类范围广
- 多模态支持
- 推理框架繁多
- 提供统一的模型范式定义
 - runtime
 - huggingface / modelscope

ModelHub

```
deployments:
  - runtime: vllm
    versionRequired: '>=0.7.1' # semver match for runtime.
    resourceRequirements:
      gpuType: nvidia-gpu
      gpuCount: 16
      perGPUMemoryGB: 80
      cpu: 8
      memory: 32
    customRuntimeArgs: [] # define runtime parameters that are optimized
    for this scenario.
  - runtime: sglang
    versionRequired: '>=0.4.3'
    resourceRequirements:
      gpuType: nvidia-gpu
      gpuCount: 16
      perGPUMemoryGB: 80
      cpu: 8
      memory: 32
    customRuntimeArgs: []
```

Part 02

云原生化的 SaaS 平台介绍

DaoCloud - D.run SaaS 平台介绍



统一的模型广场



模型广场，国内外主流开源模型，一键体验，一键部署

支持国内主流开源模型

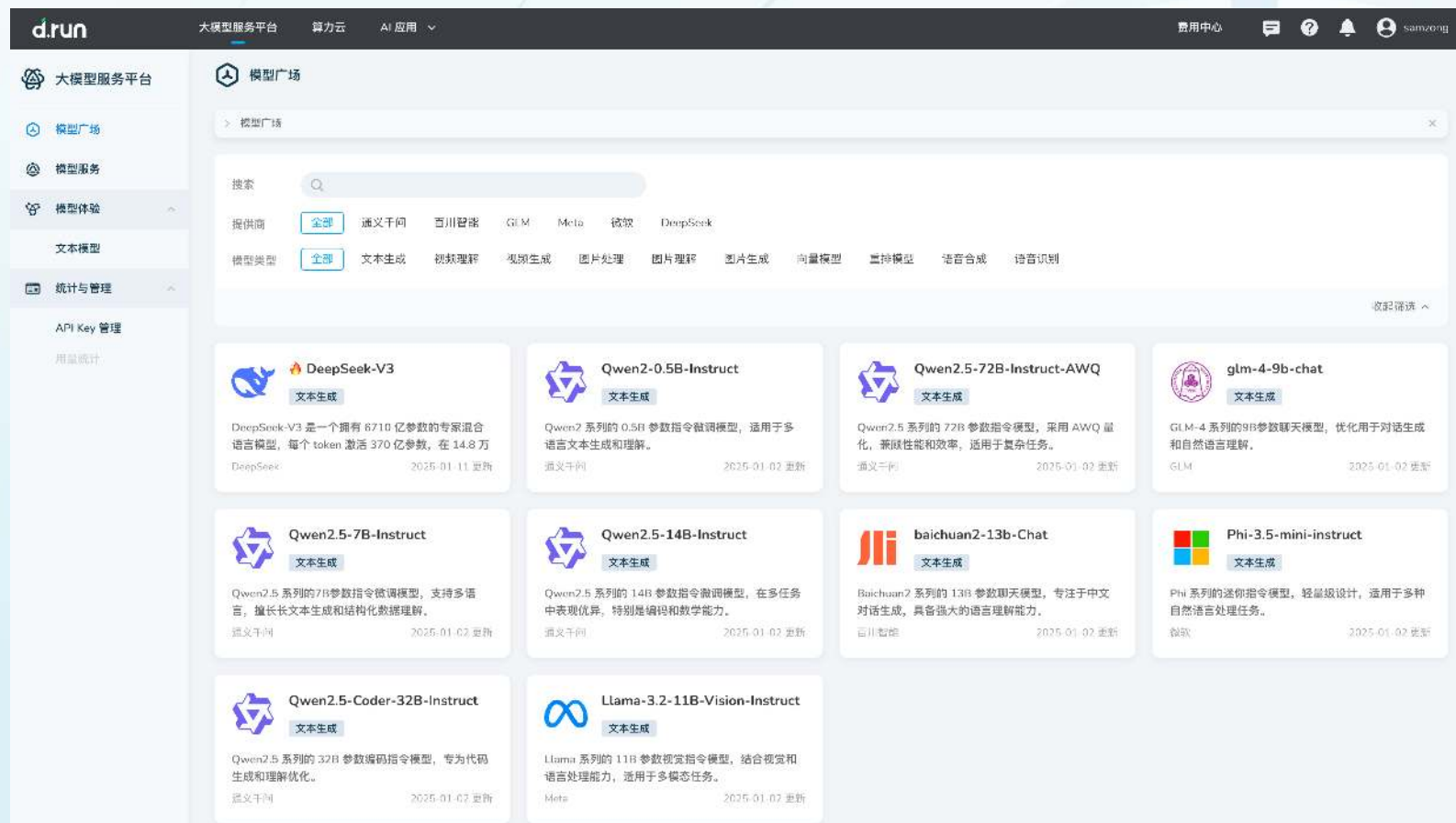
支持模型快速部署

无需自行估算算力资源

支持 API 调用，通用 API 支持

模型部署按实例计费

公共模型调用按 Token 计费



模型部署



d.run

大模型服务平台 算力云 AI 应用 费用中心

大模型服务平台

模型广场 模型服务 模型体验 文本模型 统计与管理

部署模型服务

基本信息

模型选择 Qwen2-0.5B-Instruct

模型服务名称 qwen2-05b

资源信息

地区 上海二区

实例数 1

购买信息

计费方式 按量付费 包年包月

配置费用: ¥3.52 元/小时 计算公式: 1 (实例数) * 3.52 (元/小时)

扩容 (模型服务 qwen2-05b)

实例数 当前 变更后 2

配置费用 ¥7.04 元/小时

确定

支持实时扩缩容

部署区域任意选择

极简模型创建过程

模型体验

快捷进行模型体验



DeepSeek-V3

文本生成

DeepSeek-V3 是一个拥有 6710 亿参数的混合语言模型，每个 token 激活 370 亿参数，在 14.8 万 DeepSeek

2025-01-27

体验


DeepSeek-V3 Qwen2.5-72B-Instruct-AWQ glm-4-9b-chat ...

输入 ¥ 0.000600 /千 token | 输出 ¥ 0.001200 /千 token

请输入您的问题

模型生成的所有内容由人工智能模型生成，其生成内容的准确性和实用性无法保证，平台方不对任何生成结果负责。

公共模型




DeepSeek-V3

文本生成

DeepSeek-V3 是一个拥有 6710 亿参数的专家混合语言模型，每个 token 激活 370 亿参数，在 14.8 万 DeepSeek

输入 ¥ 0.000600 /千 token | 输出 ¥ 0.001200 /千 token




Qwen2.5-72B-Instruct-AWQ

文本生成

Qwen2.5 系列的 72B 参数指令模型，采用 AWQ 量化，兼顾性能和效率，适用于复杂任务。

输入 ¥ 0.001600 /千 token | 输出 ¥ 0.004800 /千 token




glm-4-9b-chat

文本生成

GLM-4 系列的 9B 参数聊天模型，优化用于对话生成和自然语言理解。

输入 ¥ 0.000800 /千 token | 输出 ¥ 0.002400 /千 token



Llama 系列

语言处理

输入 ¥ 0.001

模型服务

暂无数据

public/deepseek-v3

公共模型

介绍下大语言模型的发展过程

大语言模型 (Large Language Models, LLMs) 的发展过程可以追溯到自然语言处理 (Natural Language Processing, NLP) 领域的早期研究。随着计算能力的提升、数据的增加以及算法技术的进步，大语言模型逐渐成为人工智能领域的核心技术之一。以下是其发展的主要阶段：

- 1. 早期阶段 (20世纪50年代-90年代)**
 - 规则驱动模型：**早期的语言处理主要依赖于手工编写的规则和语法，如基于规则的机器翻译和语法分析。这些方法虽然在某些特定任务上有效，但缺乏灵活性和泛化能力。
 - 统计模型：**20世纪90年代，统计方法开始主导NLP领域，如隐马尔可夫模型 (HMM) 和贝叶斯方法。这些模型通过统计概率来处理语言，但依然受限于数据量和计算能力。
- 2. 深度学习兴起 (2000年代-2010年代)**
 - 神经网络模型：**随着深度学习技术的兴起，神经网络开始应用于语言处理任务。早期的模型如递归神经网络 (RNN) 和长短期记忆网络 (LSTM) 在文本生成、分类等任务上表现出色。
 - Word Embeddings：**词嵌入技术 (如 Word2Vec、GloVe) 将词汇映射到高维向量空间，极大地提升了语言模型的表现，为后续模型提供了基础。
- 3. 大语言模型的出现 (2010年代后期)**
 - Transformer架构：**2017年，Google提出了Transformer架构，通过自注意力机制 (Self-Attention) 解决了传统RNN在处理长序列时的效率问题。Transformer成为大语言模型的核心技术。
 - GPT系列：**OpenAI在2018年发布了GPT (Generative Pre-trained Transformer)，通过大规模预训练和微调，GPT展示了强大的文本生成能力。随后，GPT-2 (2019) 和GPT-3 (2020) 进一步提升了模型的规模和性能，GPT-3拥有1750亿参数，成为当时最强大的语言模型之一。
 - BERT：**Google在2018年发布了BERT (Bidirectional Encoder Representations from Transformers)，通过双向注意力机制在文本理解任务上取得了显著进展。BERT及其变体 (如BERT-large) 成为NLP领域的标杆。
- 4. 大规模应用与优化 (2020年代至今)**

请输入您的问题

OpenAPI 调用（监控 OpenAI 接口风格）



模型广场，国内外主流开源模型，一键体验，一键部署

支持国内主流开源模型

支持模型快速部署

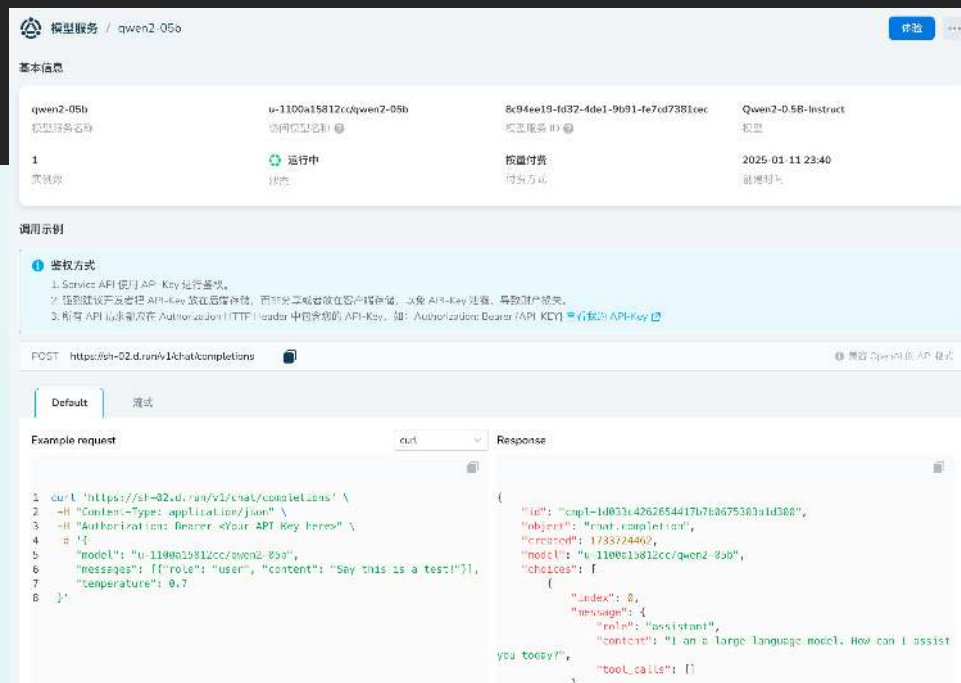
无需自行估算算力资源

支持 API 调用，通用 API 支持

模型部署按实例计费

公共模型调用按 Token 计费

```
curl 'https://sh-02.d.run/v1/chat/completions' \
-H "Content-Type: application/json" \
-H "Authorization: Bearer sk-x1VDTAFB7Ra1hIdATbncOa_dddVttDvRHQibTA-Oi7ucU" \
-d '{
  "model": "u-8105f7322477/test",
  "messages": [{"role": "user", "content": "Hello, model!"}],
  "temperature": 0.7
}'
```



Part 03

开源技术的力量（AI/LLM）

使用开源 > 参与开源 > 贡献开源



HAMi



BEIJING

HAMi(Heterogeneous AI Computing Virtualization Middleware) 异构 AI 芯片虚拟化组件，旨在解决 AI **芯片使用率瓶颈** 与 **异构 AI 统一管理** 两大挑战。

HAMi 支持以 **插拔式、轻量级、无侵入** 部署在任意云环境，使用成本低、对 AI 应用无侵入性。

插拔式

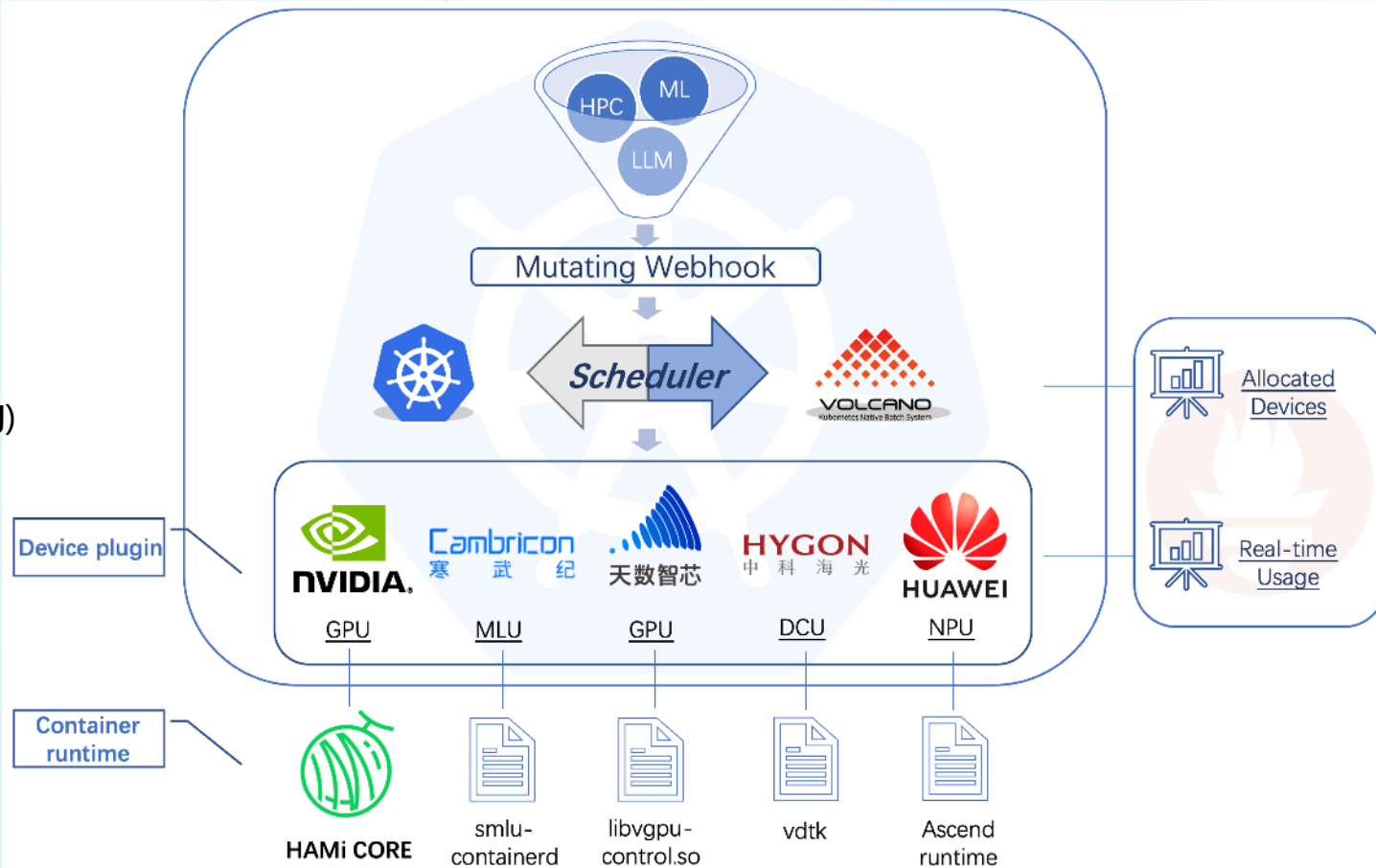
轻量级

应用无侵入

标准化

核心竞争力：

- GPU 细粒度、按需虚拟化(支持任意操作系统、任意架构)
- 算力资源抢占，优先保障高优先级任务
- 异构 AI 芯片 统一管理、调度、监控，提高管理效率，降低复杂性
- 算力、显存超配
- 丰富而灵活的调度策略应对更多的 AI 应用互联场景
- 企业租户 配额 管理，好钢用在刀刃上



vllm & sglang

- 大模型推理目前支持，支持 vllm 与 SGLang
- 主动参与贡献



Part 04

未来规划

AI



项目开源计划



baizeai/KCover

baizeai/modelhub

baizeai/knoway

baizeai/KCover

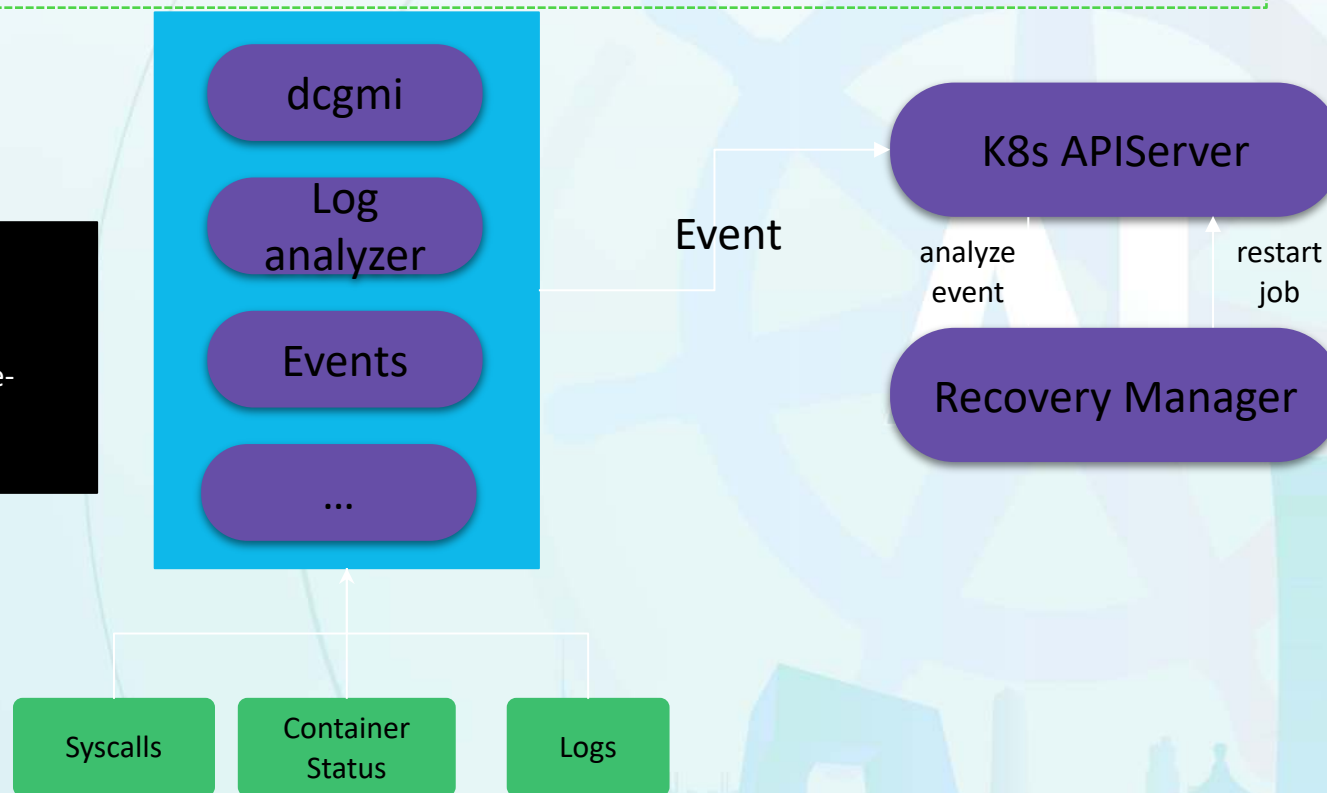


通过对采集到 Node、Pod 以及训练任务等的指标信息。

KCover 提供了一套全自动的断卡异常检测，帮助 GPU 服务自动恢复的能力。

```
λ helm repo add baizeai https://baizeai.github.io/charts
λ helm repo update baizeai

λ helm install kcover baizeai/kcover --namespace kcover-system --create-namespace
```



<https://github.com/baizeai/kcover>
<https://baizeai.github.io/talks/2024-08-21-kubecon-hk>

baizeai/ModelHub



- ModelHub 是一个基于 Kubernetes 的AI模型管理工具，专注于大型语言模型（LLM）和多模态模型的部署、管理和运行。旨在提供了一套标准化的方式来定义、部署和管理各种AI模型，支持从不同来源（如Hugging Face、ModelScope）获取模型权重，并通过不同的运行时（如vLLM、SGLang）进行高效部署。

```
deployments:
- runtime: vllm
  versionRequired: '>=0.7.1' # semver match for runtime.
  resourceRequirements:
    gpuType: nvidia-gpu
    gpuCount: 16
    perGPUMemoryGB: 80
    cpu: 8
    memory: 32
  customRuntimeArgs: [] # define runtime parameters that are optimized
for this scenario.
- runtime: sglang
  versionRequired: '>=0.4.3'
  resourceRequirements:
    gpuType: nvidia-gpu
    gpuCount: 16
    perGPUMemoryGB: 80
    cpu: 8
    memory: 32
  customRuntimeArgs: []
```

AI 网关



轻量级且易于使用的专用网关，具有各种针对LLM的特定优化和功能。你可以把它想象成Nginx，但专门为LLM和即将支持的模型（如Stable Diffusion等）设计。

- 🧑‍🔧 无服务器引导加载器: 能够按需引导服务的上游Pod，使LLM服务更具成本效益。
- ✅ 容错: LLM的容错能力，在与外部提供商打交道时具有重试，断路等能力。
- 🚦 速率限制: 基于令牌、提示等的速率限制，以保护服务于服务的LLM不被滥用。
- 📖 语义缓存: 基于提示和令牌的语义进行缓存，LLMs的CDN。
- 📖 语义路由: 根据提示的困难、语义等进行路由，以使LLMs服务更高效，模型正确。
- 🔍 OpenTelemetry: OpenTelemetry支持，能够跟踪对LLMs的调用以及网关本身。

AI
Knoway

Thanks.

