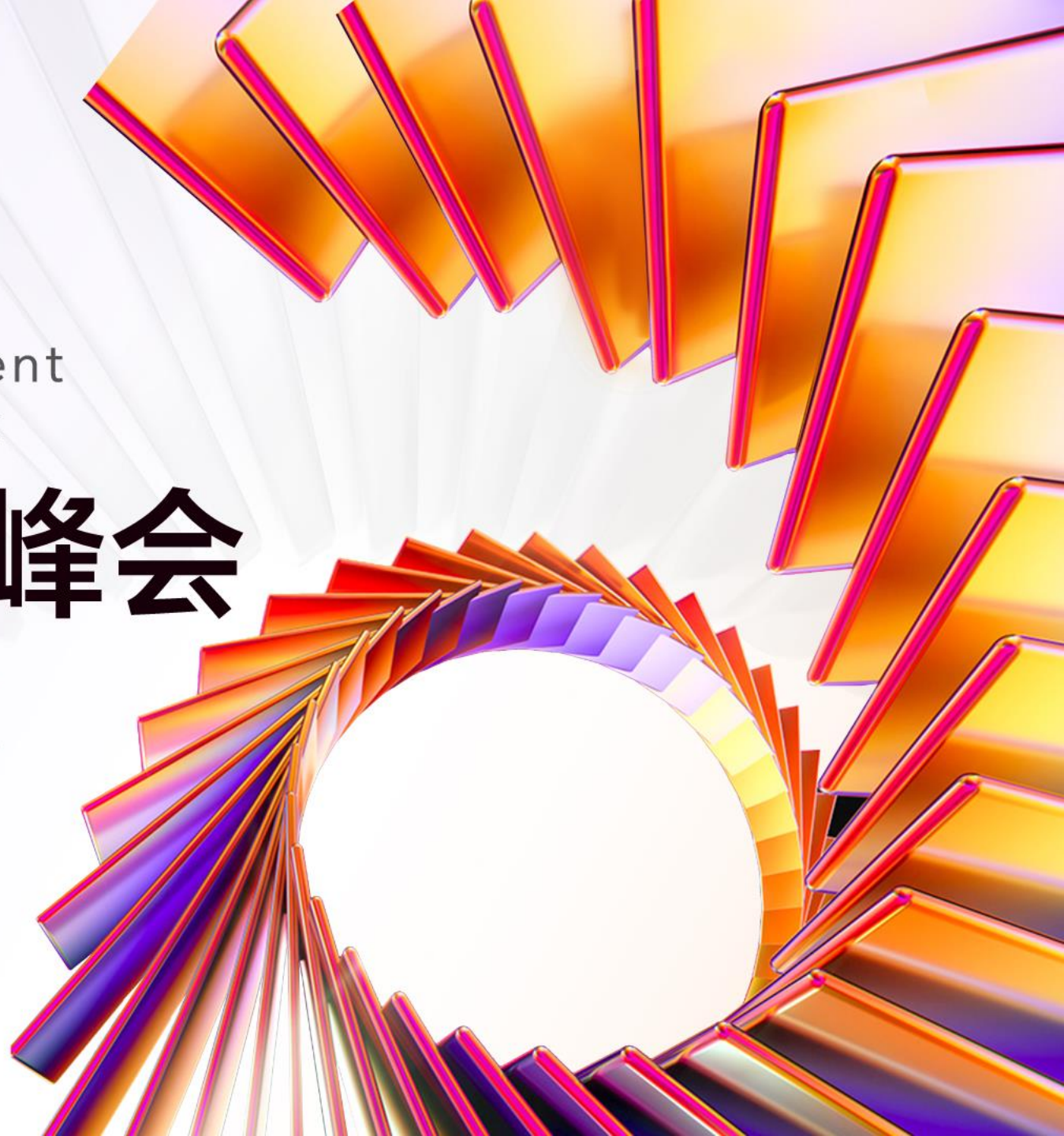
AiDD 8th 2025

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时代的用户洞家与需求发现

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+ 业界大咖 汇聚产品大咖的真知灼见

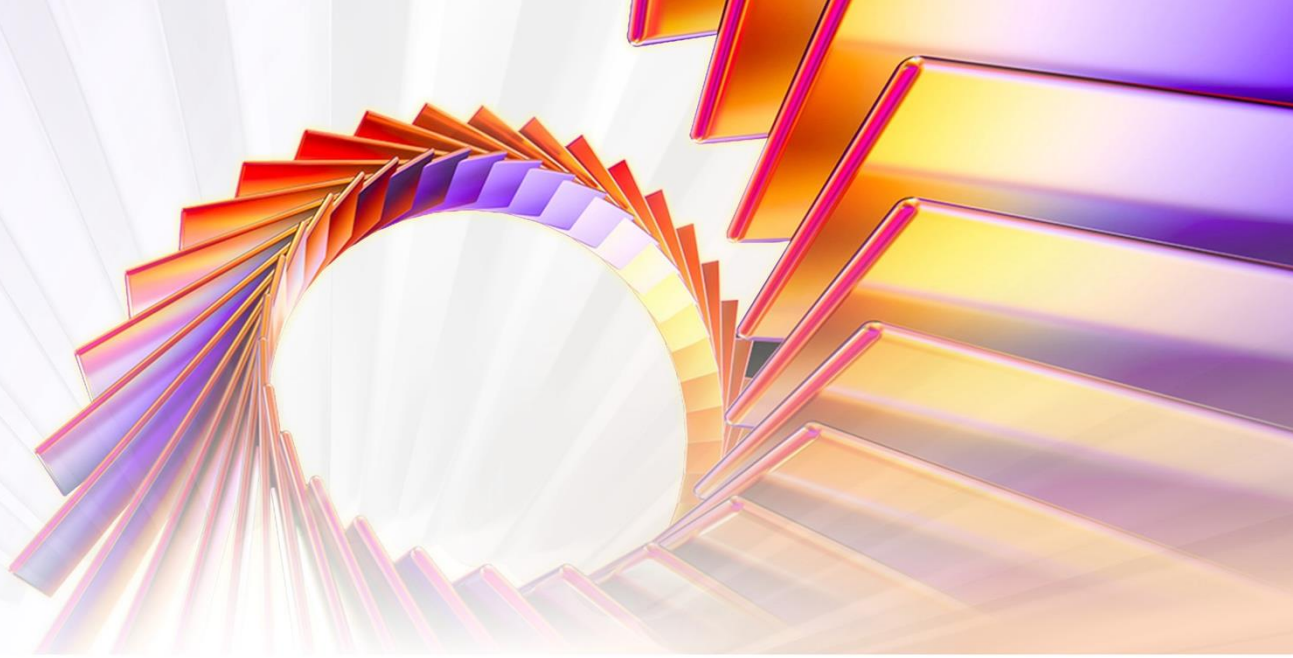
40+ 创新案例 开拓全新AI+产品视角

10+ 分论坛 涵盖产品到商业增长的全链路

800+ 听众规模 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



云原生大模型推理平台的企业级实践
基于Kubernetes的分布式推理架构设计与优化

徐中虎 | 华为云

徐中虎

CNCF TAG-Infra 技术负责人

Istio 治理委员会成员

华为云 主任工程师

Kthena社区负责人

CNCF TAG-Infra技术负责人，致力于帮助网络项目健康发展。Istio治理委员会成员，自2018年以来一直是Istio的核心维护者，也是Istio前三大贡献者。中虎是多个CNCF项目的维护者，包括Istio、Kmesh和Volcano等，也是Kubernetes前100名贡献者。拥有丰富的开源工作经验，主要研究方向有云原生、Kubernetes、容器、服务网格及分布式大模型推理。中虎还是《云原生服务网格Istio》、《Istio权威指南》的联合作者。

目录
CONTENTS

PART 01

LLM推理工作负载编排的痛点

LLM推理现状

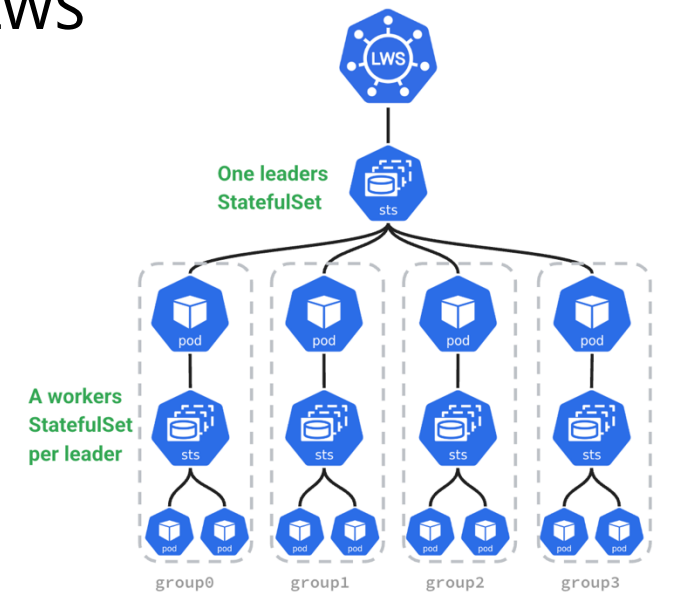
1. LLM推理不同于微服务，它是有状态应用。
2. 多种推理引擎vLLM， SGLang， Triton， TGI并存
3. 对于超大参数规模的模型，采用TP，PP，DP，EP等多种并行计算组合的方式多机部署
4. PD分离架构对于大模型推理能够降低Prefill和Decode的相互干扰，显著提升吞吐和时延SLO
5. PD分离架构，Prefill和Decode角色需要KV Cache的传输

PD分离架构：基于LWS封装

Wrapper Operator

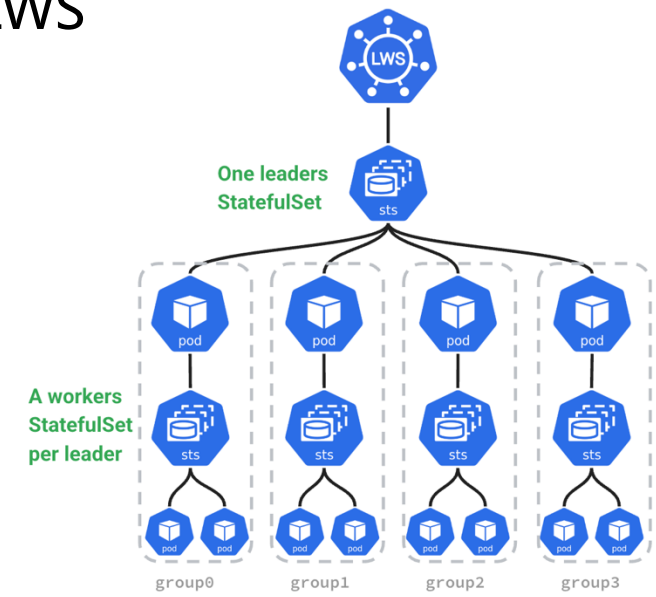
LWS Operator

LWS



Prefill

LWS



Decode

- 部署形态，顶层Operator + LWS，多组件维护成本高
- 调度策略配置复杂： 顶层Operator需要感知LWS创建的子资源
- 尤其在PD分离场景下：假设创建两个LWS表示2P4D的部署架构。
  - N个PD组，2*N个LWS表示
  - Gang调度，两LWS协同，缺乏调度支持
  - 拓扑感知调度：同PD组的worker尽量部署在同一网络性能域（例如超节点），同一Prefill或者Decode角色组内的Worker，网络要求更严格。
- 多层CR对象 Wrapper -》LWS-》Statefulset -》Pod

PART 02

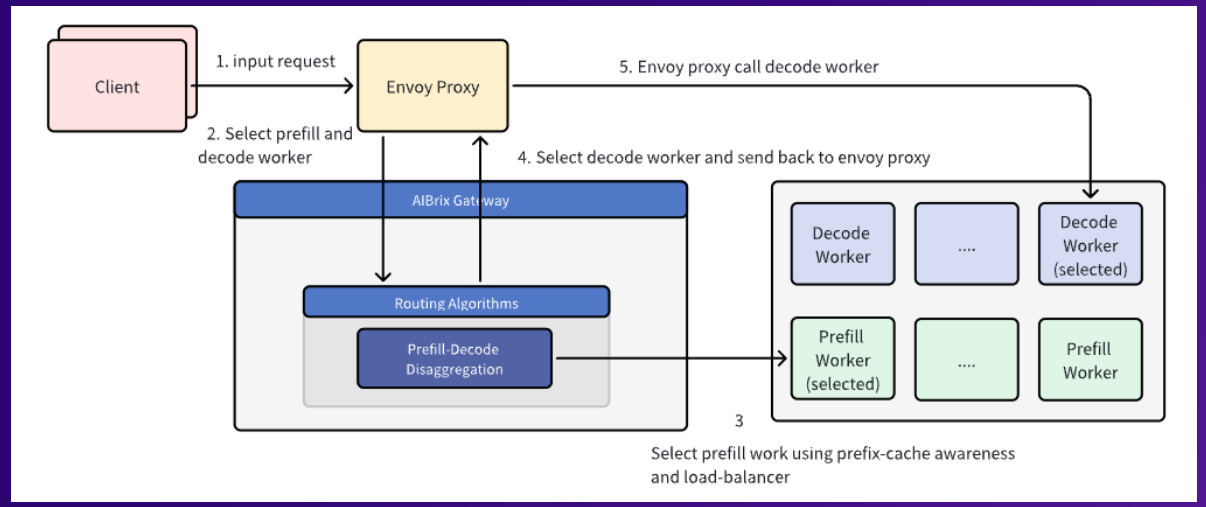
LLM网关的痛点

LLM推理与微服务的差异

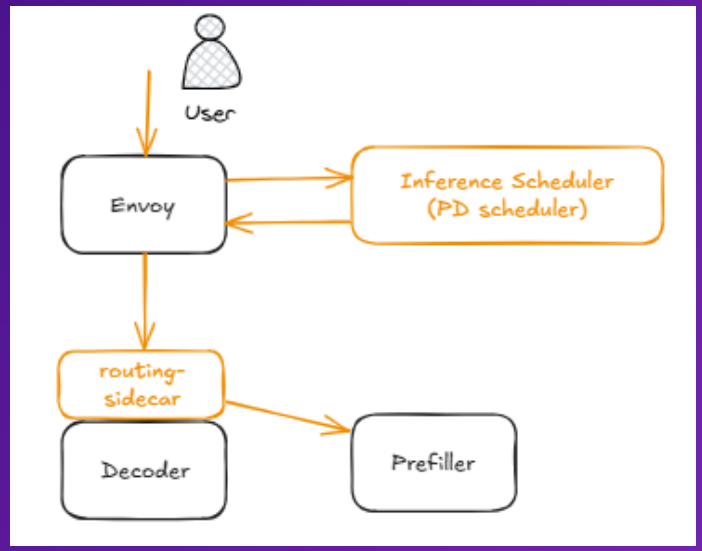
- 请求间时延、服务器资源消耗差异不大
- 微服务一般无状态
- GateWay or 负载均衡器不感知服务负载，常用的负载均衡算法：RR，Random、LeastRequest
- 限流：请求数粒度

- 推理请求差异巨大：时延、服务器资源消耗与Prompt长度及输出Token的长度息息相关
- 推理有状态应用，KV Cache命中率对性能影响很大
- LLM Gateway需要感知推理引擎负载、KV Cache
- 限流：Token级限流
- 高级特性：语义缓存，多模型语义路由

开源项目普遍采用Envoy扩展插件实现LLM网关



- 调度策略必须在请求Header：routing-strategy指定
- PD分离的调度方式比较Hack，
- 只支持单xPyD分组调度，因为只能识别Prefill Decode角色，不能识别它们是否属于同一个PD组
- 支持多模型路由



Llm-d infer scheduler

- 多组件依赖，维护成本高
- 不支持PD Group的感知调度
- 不支持多模型路由
- 支持Gateway API， Inference Extension API

Envoy扩展方式：运维复杂

- 复用Envoy的流量治理能力 = 历史债务，依赖过多，运维复杂
- LLM扩展插件，实际上是短路了Envoy原有的负载均衡过程
- 请求、响应均绕行Extension Processor，链路过长，故障定界复杂
- 高阶功能，Token限流，公平调度等支持有限

PART 03

云原生推理开源项目Kthena

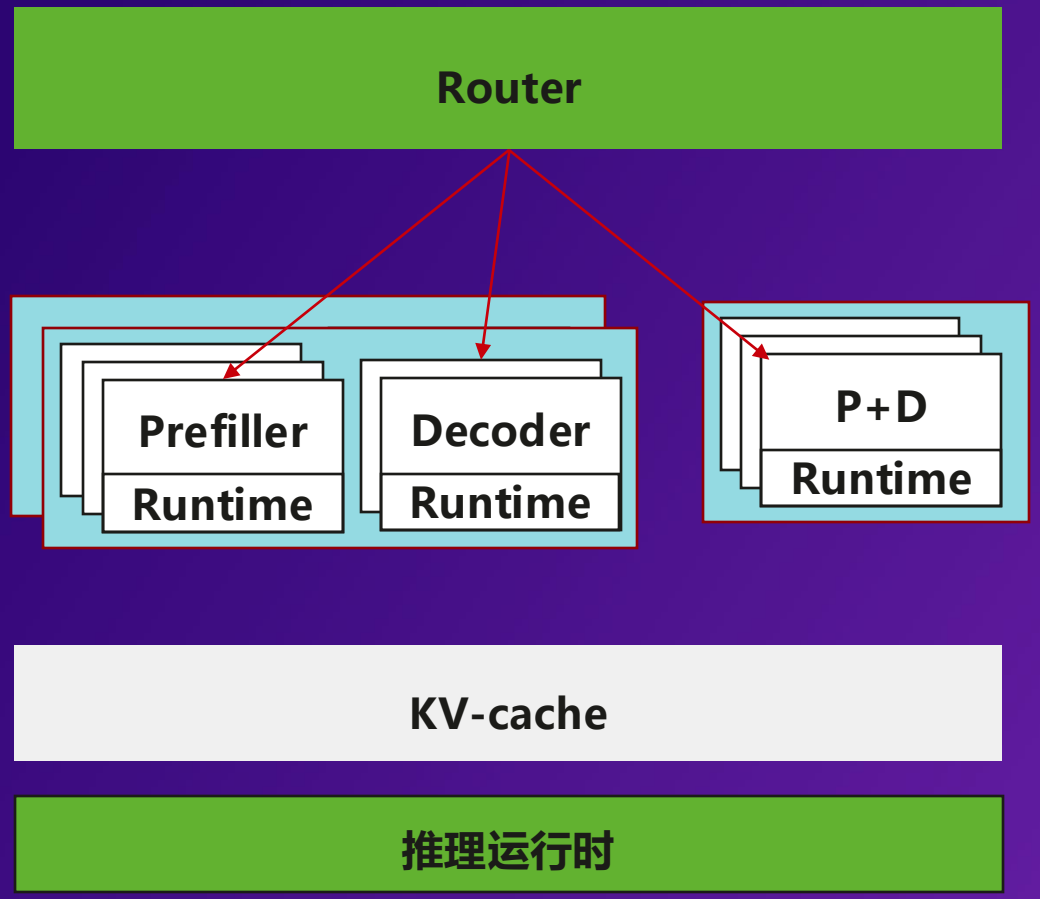
什么是Kthena

整体目标：为云原生重新定义大语言模型（LLM）智能推理

模型统一管理

分布式推理workload编排

Gang、超节点拓扑感知调度（volcano）



多模型路由

支持Token级限流
模型负载感知调度
KV Cache感知调度
PD分组路由

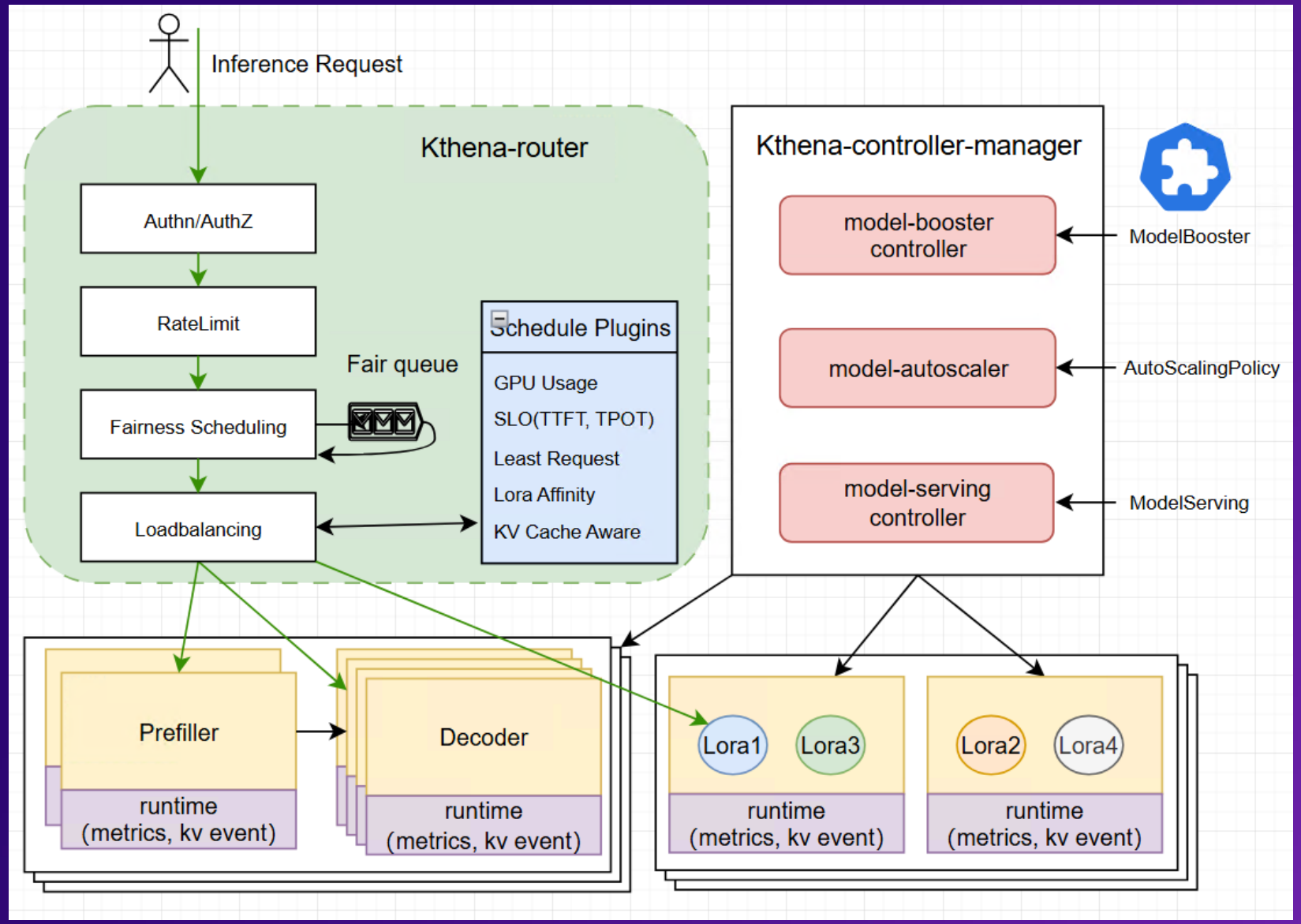
能力开箱即用

内置分布式推理、PD分离等最佳部署范式
内置超节点网络拓扑感知、Gang调度
全栈推理服务、KV Cache、模型路由管理

急速弹性

提供模型快照、镜像预热等能力
加速推理模型启动和扩缩容

Kthena架构图



1. Kthena Router：轻量高性能的网关
2. ModelServing ：分布式推理最佳部署范式，打造灵活、易用的PD分离、以及PP、TP等并行模式的部署范式，提供超节点拓扑感知的编排和快速故障恢复。
3. ModelBooster：模型元数据管理，模型参数预热，启动加速；动态的LoRA加载和卸载。
4. 弹性扩缩容：标准化推理指标采集，基于LLM实例负载指标提供极致弹性的扩缩容。

Kthena核心组件

1. **Kthena Router**：一个高性能的数据平面，负责接收所有推理请求，并根据 ModelRoute 规则，智能地将请求分发到后端的 ModelServer。
2. **Kthena Controller Manager**：

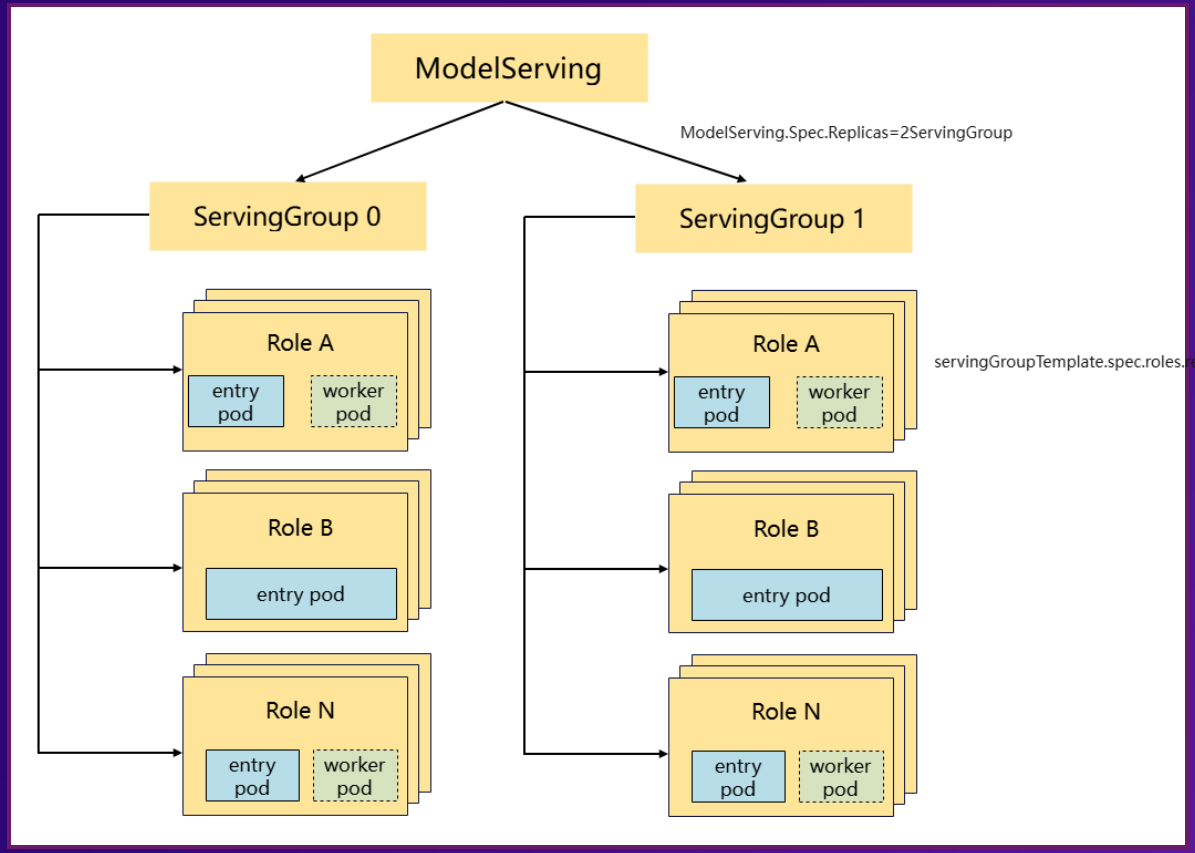
- Kubernetes 控制平面的控制器，它主要包含多种控制器，主要负责 LLM 工作负载的编排与生命周期管理。ModelServing 控制器负责编排推理工作负载分组；支持网络拓扑亲和调度和Gang调度、滚动升级与故障恢复；
- AutoScaler 支持成本驱动的弹性扩缩容。

这种架构使得 Kthena 成为连接用户请求与 LLM 模型的、高度可编程的桥梁。

PART 04

Kthena的设计及实现

推理工作负载编排设计1



三层架构设计：ModelServing -> ServingGroup -> Role，支持LLM原生部署、PD分离部署，乃至大EP部署等多种部署形态，简化管理多LWS的负担。

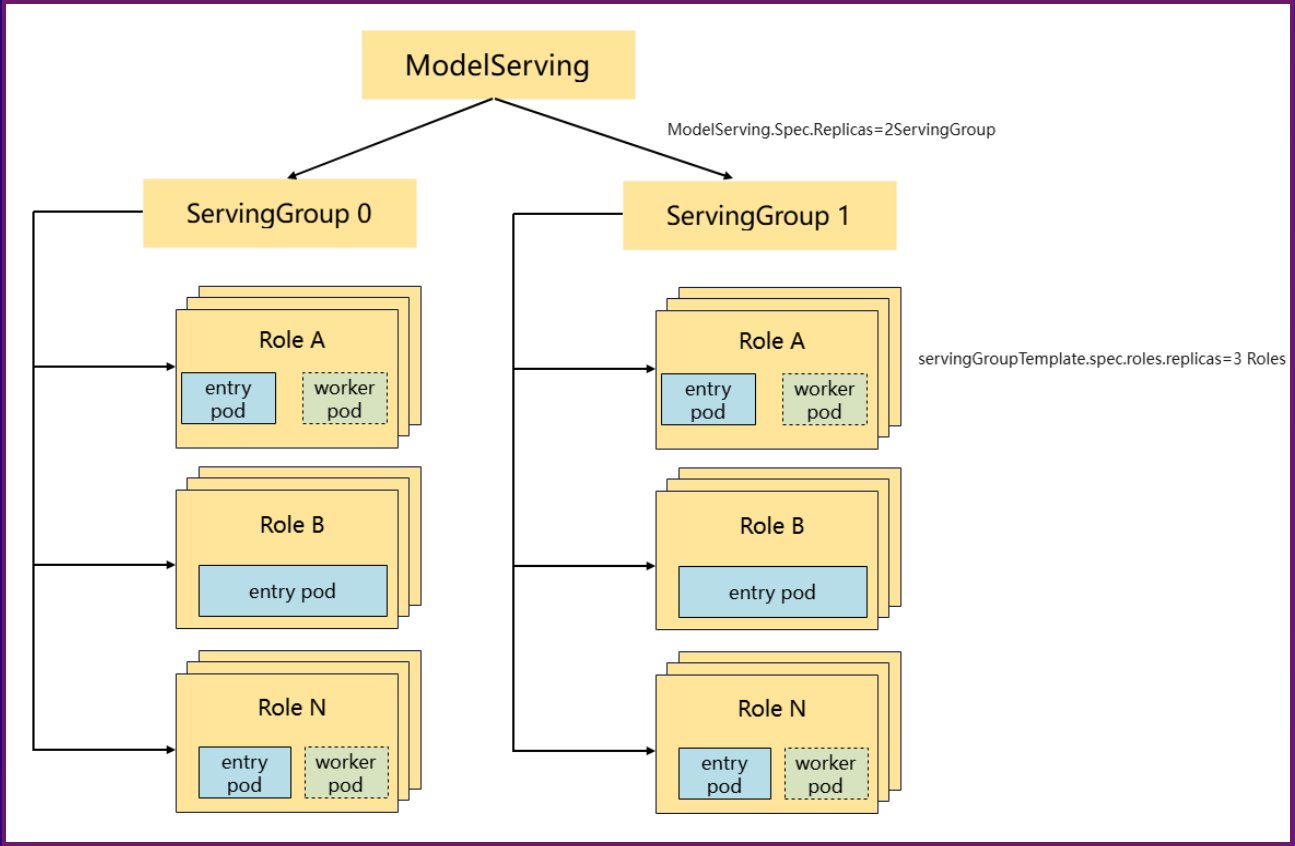
ServingGroup：推理实例组，代表能够独立完成一次推理服务的最小单元。

- 基于角色表示Prefill Decode等角色的推理实例，可管理xPyD复杂的场景
- Gang调度，尤其在xPyD场景下，使用volcano进行两级gang调度。

Role：Prefill或者Decode，在非PD分离部署架构下，一个ServingGroup可以只有一种Role。

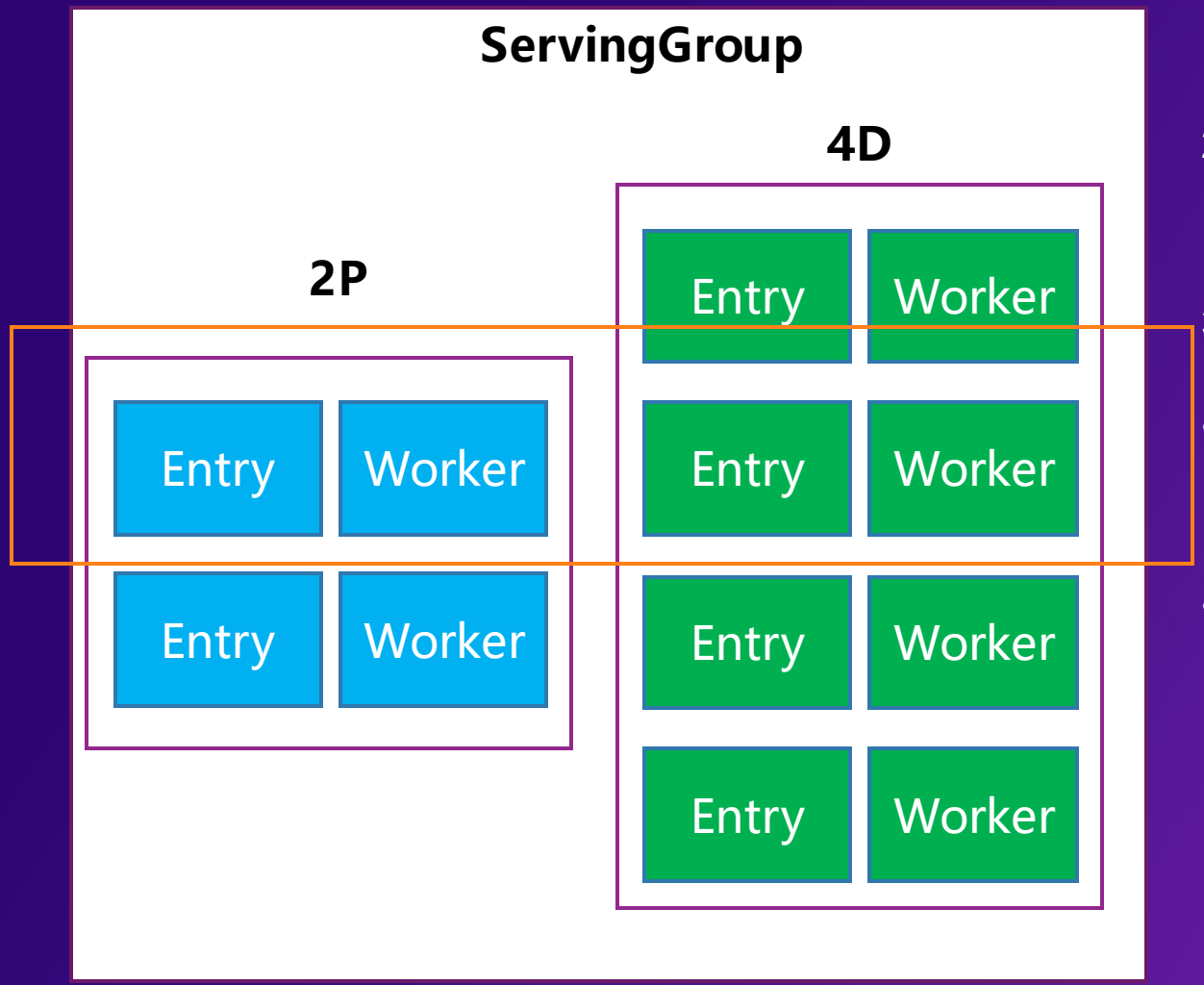
- 每个Role等同于单LWS实例，遵循原子调度。
- 网络拓扑感知调度，同一Role的Worker采用PP、DP，EP等策略，必须调度到同一超节点。

推理工作负载编排设计2



1. Prefill-Decode分离部署：可以独立伸缩，动态调整Prefill-Decode的比例，更灵活的应对各种复杂的业务场景（如长短句混合、实时推理等）。
2. 多并行范式：TP/PP/DP/EP 等并行模式灵活配置，最大化提升资源利用率和SLO
3. 支持业界主流的推理引擎vLLM、SGLang、Triton、TGI 等
4. 拓扑感知 + Gang 调度：Gang 调度确保ServingGroup/Role“成组原子化”落地，避免资源浪费；拓扑感知调度通过将Role内的一组Pod调度到网络拓扑更优的节点，提升并行计算的数据传输时延。

推理工作负载的调度设计-借助Volcano

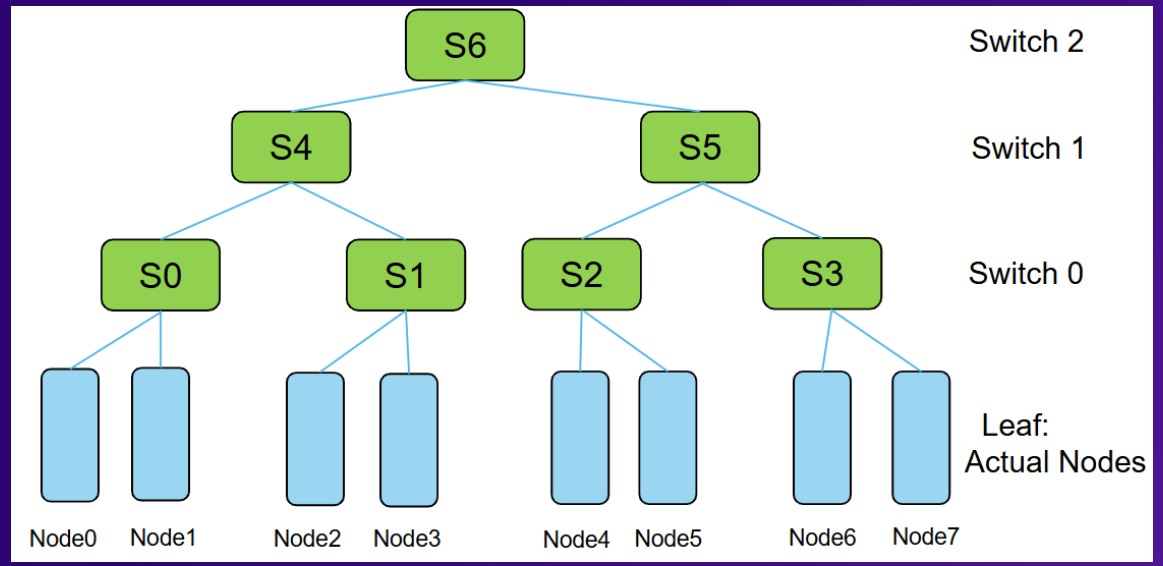


1. 严格Gang调度：2P 4D，12个Pod成组调度。
2. 宽松Gang：至少1P 1D，4个Pod成组调度；其余Role内Entry和Worker成组调度
3. 拓扑感知调度：

- 同Role的内的Entry Worker可能涉及All2All AllReduce通信，不能跨交换机
- Prefill和Decode之间需要传递KV Cache，相对Role内的数据交换对带宽时延要求更低，所以网络拓扑要求可以放松。

Volcano网络拓扑感知调度：超节点抽象

HyperNode: Standardizing Cluster Topologies via Performance Domains



- **Leaf Switch** (s0, s1, s2, s3)： HyperNode成员是真实的集群节点
- **Spine Switch** (s4, s5, s6)：指向其他的超节点

- node0 和 node1 通信效率最高
- node1 and 跨两层交换机，通信效率次之
- node0 and node4 跨三层交换机(s0→s4→s6) 通信效率最低.

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

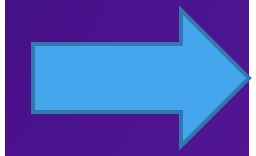
```yaml
apiVersion:topology.volcano.sh/v1alpha1
kind: HyperNode
metadata:
  name: s6
spec:
  tier: 3
  members:
  - type: HyperNode
    selector:
      exactMatch:
        name: s4
```

ModelServing 配置网络拓扑

```yaml
…
servingGroup:
  networkTopology:
    mode: hard
    highestTierAllowed: 1
…
```

PD分离部署样例

```yaml
apiVersion: workload.serving.volcano.sh/v1alpha1
kind: ModelServing
metadata:
  name: sample
  namespace: default
spec:
  schedulerName: volcano
  replicas: 1 # servingGroup replicas
  template:
    restartGracePeriodSeconds: 60
    gangpolicy:
      minRoleReplicas:
        prefill: 2   # 1 entry + 1 worker
        decode: 2
    roles:
    - name: prefill
      replicas: 4
      # ... additional role configuration
    - name: decode
      replicas: 4
      # ... additional role configuration
```



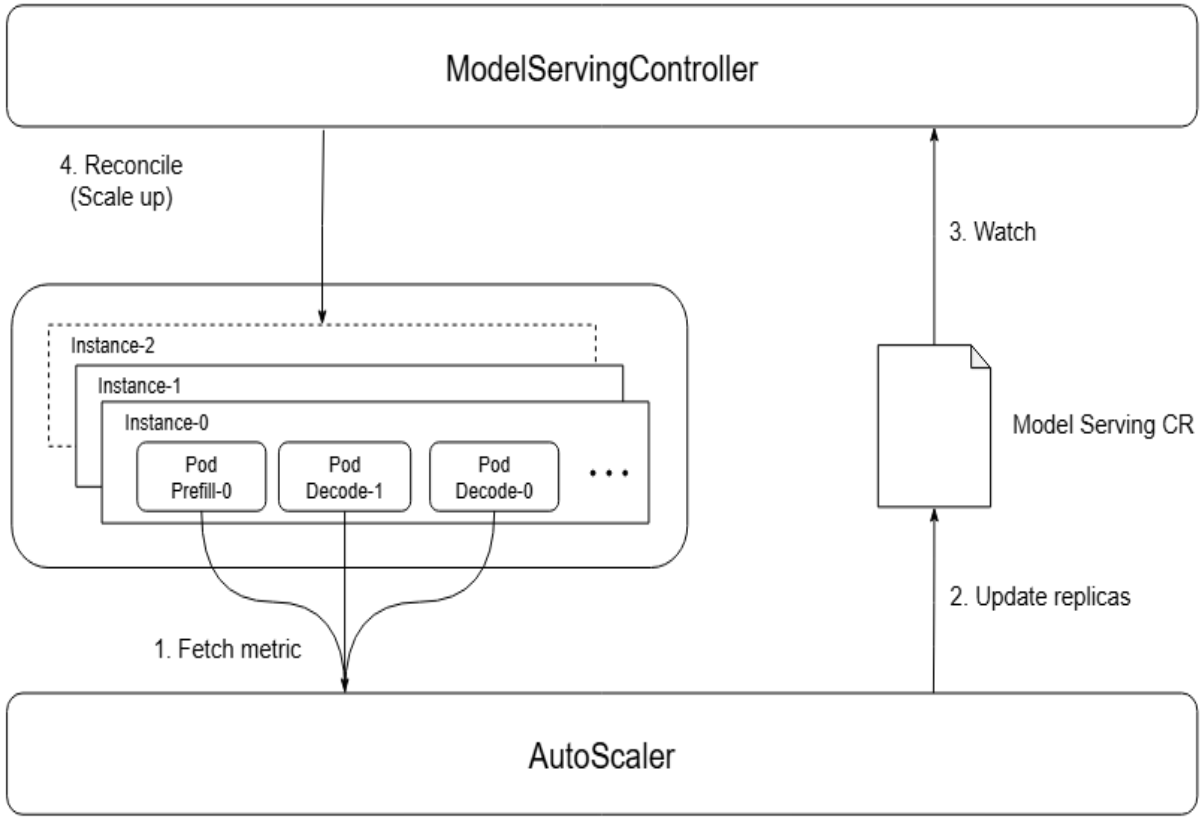
```yaml
apiVersion: scheduling.volcano.sh/v1beta1
kind: PodGroup
metadata:
  name: {modelserving-name}-{servinggroup-index}
  namespace: {modelserving-namespace}
  labels:
    modelinfer.volcano.sh/name: {modelserving-name}-
{servinggroup-index}
  annotations:
    scheduling.k8s.io/group-name: {modelserving-name}
spec:
# MinTaskMember defines minimum pods required for each role
replica
  minTaskMember:
    "prefill-0": 2
    "decode-0": 2
  # ... additional role replicas
  minResources:
```

滚动升级

	R-0	R-1	R-2	R-3	Note
Stage1	❎	❎	❎	❎	Before rolling update
Stage2	❎	❎	❎	⏳	Rolling update started, The replica with the highest ordinal (R-3) is updated
Stage3	❎	❎	⏳	✅	R-3 is updated. The next replica (R-2) is now being updated
Stage4	❎	⏳	✅	✅	R-2 is updated. The next replica (R-1) is now being updated
Stage5	⏳	✅	✅	✅	R-1 is updated. The last replica (R-0) is now being updated
Stage6	✅	✅	✅	✅	Update completed. All replicas are on the new version

1. ServingGroup是完成推理服务的最小单元。ModelServing创建的ServingGroup具有稳定的编号0,1,2…N-1
2. 当前Kthena支持按照ServingGroup粒度进行滚动升级。
3. 滚动升级策略中有Partition可以控制灰度的滚动升级，类似StatefulSet

弹性扩缩容

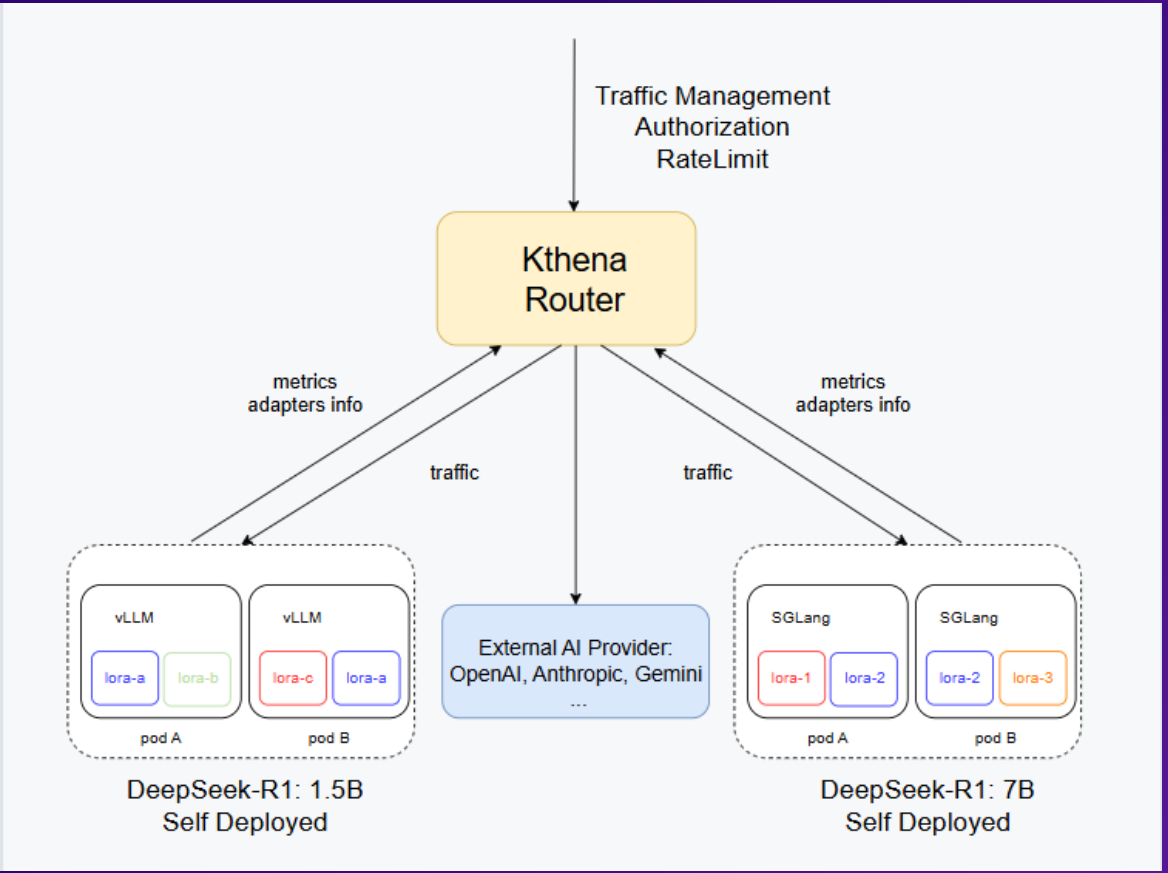


1. 支持同构、异构扩缩
2. 同构扩缩：类似KPA，支持Stable和Panic两种模式。基于ASP策略和Metrics，调整ModelServing的Replicas
3. 异构扩缩：同一个模型可以部署在不同的加速卡上，以及使用不同的推理引擎，异构扩缩可以根据加速卡的成本和业务处理能力，通过整数求解器规划最优的硬件资源配置。

模型预热、Lora管理、指标标准化

1. 大模型参数量巨大，实时下载影响启动速度（分钟级），浪费GPU、NPU资源
2. Kthena提供了模型预下载的能力
3. 针对不同的引擎vLLM，SGLang，原始的Metrics不同，Kthena Runtime提供了标准化的指标，供Autoscaler或者网关使用
4. Lora适配器生命周期管理， Kthena Runtime通过API接口自动热加载和卸载Lora

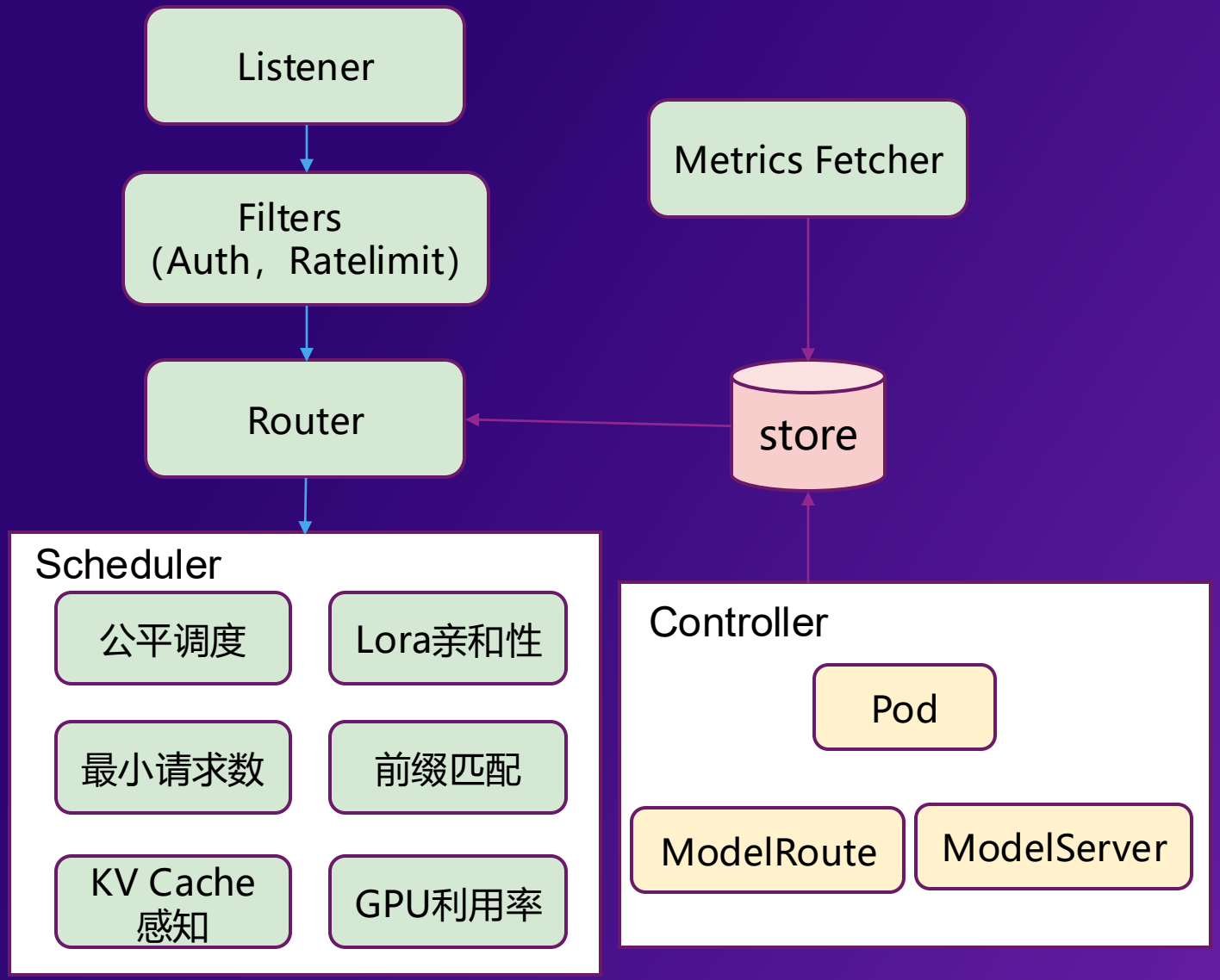
模型感知的智能路由



轻量、易用、高性能的Router，支持自部署的多模型调度和三方厂商的LLM服务路由

1. 模型感知路由：利用来自推理引擎（vLLM、SGLang、TGI）的实时指标来做出智能路由决策
2. LoRA 亲和性调度：智能处理动态 LoRA 适配器加载和卸载
3. 高级调度算法：包括前缀缓存感知、KV 缓存感知和公平性调度等。
4. PD分离：对 xPyD（x-prefill/y-decode）部署模式的原生支持
5. Token级限流：支持过去时间窗内的Token使用量限制，符合当前MaaS服务计费规则

Router核心模块



1. Listener：管理 HTTP/HTTPS 侦听器并处理指定端口上的传入流量。它为不同协议提供灵活的配置，可以绑定多个地址来服务各种类型的请求。
2. Controller：List-Watch Pod 和自定义资源 (CR)，例如 ModelRoute 和 ModelServer。
3. Filters: 包含认证和限流过滤器，限流支持Token级本地、全局限流
4. 调度当前支持多种算法组合使用，其中KV Cache感知的算法对TTFT提升最优帮助
5. 公平调度：优先处理token用量小的用户请求

Router API

```yaml
apiVersion: networking.serving.volcano.sh/v1alpha1
kind: ModelRoute
metadata:
  name: deepseek-multi-models
  namespace: default
spec:
  modelName: "deepseek-multi-models"
  rules:
  - name: "premium"
    modelMatch:
      headers:
        user-type:
          exact: premium
    targetModels:
    - modelServerName: "deepseek-r1-7b“
  - name: "default"
    targetModels:
    - modelServerName: "deepseek-r1-1-5b"
```

```yaml
apiVersion: networking.serving.volcano.sh/v1alpha1
kind: ModelServer
metadata:
  name: deepseek-r1-7b
  namespace: default
spec:
  workloadSelector:
    matchLabels:
      app: deepseek-r1-7b
  workloadPort:
    port: 8000
  model: "deepseek-ai/DeepSeek-R1-Distill-Qwen-7B"
  inferenceEngine: "vLLM"
  trafficPolicy:
    timeout: 10s
---
apiVersion: networking.serving.volcano.sh/v1alpha1
kind: ModelServer
metadata:
  name: deepseek-r1-1-5b
  namespace: default
spec:
  workloadSelector:
    matchLabels:
      app: deepseek-r1-1-5b
  workloadPort:
    port: 8000
  model: "deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B"
  inferenceEngine: "vLLM"
  trafficPolicy:
    timeout: 10s
```

极致的性能提升

Plugin Config	Throughput	Latency(s)	TTFT(s)
Least Request + KVCacheAware	**32.22**	**9.22**	**0.57**
Least Request + Prefix Cache	23.87	12.47	0.83
Random	11.81	25.23	2.15

基于 Kthena Router 的调度插件架构，在长系统提示场景（如 4096 tokens）下，采用“KV Cache 感知 + 最少请求”策略相较随机基线：

- 吞吐可提升约 2.73 倍
- TTFT 降低约 73.5%
- 端到端时延降低超过 60%

PART 05

总结与展望

Roadmap

Router：

- 通过多Listener隔离同名的Model
- 兼容Gateway API
- 支持代理公共的MaaS服务
- 公平调度增强：智能自适应的排队策略
- 负载均衡算法：支持精准的负载感知，消除推理引擎排队引发KV Cache失效

Orchestration：

- Group和Role融合的精细Gang调度及拓扑亲和性调度
- Role API层面支持LWS API
- ModelBooster提供一键部署主流大模型的最佳模板
- AutoScaler支持Prefill Decode角色独立扩缩，以适应不同场景下最佳性能要求

欢迎参与Kthena

Github

Slack

Website：https://kthena.volcano.sh/

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时代的用户洞家与需求发现

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+ 业界大咖 汇聚产品大咖的真知灼见

40+ 创新案例 开拓全新AI+产品视角

10+ 分论坛 涵盖产品到商业增长的全链路

800+ 听众规模 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