InfoQ 极客传媒

以 KVCache 为中心的云上 LLM 推理软件栈

演讲人：马腾

阿里云 / 高级技术专家、
KVCache.AI 成员
Mooncake/SGLang/RBG/AIGW Committer

AiCon
全球人工智能开发与应用大会

AiCon
全球人工智能开发与应用大会

目录

极客邦科技 2026 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

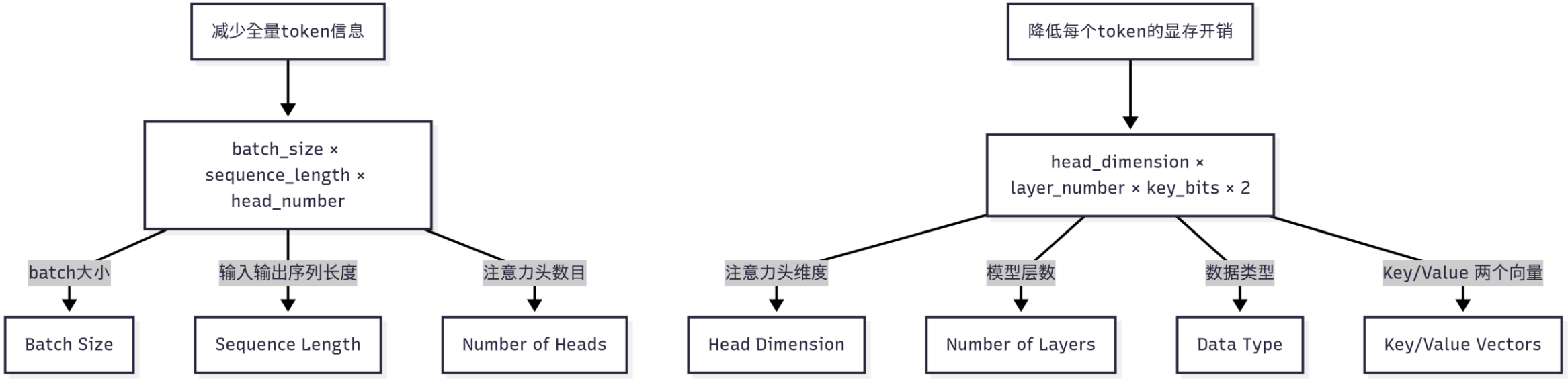
会议时间：12月18-19日

- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

01 大语言模型推理和KVCache

推理过程KVCache计算方案



静态稀疏化
滑动窗口

减少Head数
GQA/MQA

降低维度
MLA

只存部分层
YOCO/CLA

量化
FP8/Int4

高效显存管理
Page Attention
vAttention

例子 $\text{4K context:} \quad \underset{\text{seqlen}}{4000} \times \underset{\text{layer}}{60} \times \underset{\text{head}}{8} \times \underset{\text{dim}}{128} \times \underset{\text{KV}}{2} \times \underset{\text{bf16}}{2} \text{ bytes} = 0.91\text{GB}$

KVCache 缓存对存储系统的挑战

- 每一个 1 token 对应 2 * 层数 * 隐藏维度 = 数十乃至数百 KB 的 KVCache
- 不仅数据量极大，还需要尽可能地快速进行传输不然会导致 GPU 空转

每天数千亿 Token
的大型推理服务

一个 Token
(数 Bytes 级别)

一个 Token 对
应的 KVCache
(数十 KB 级别)

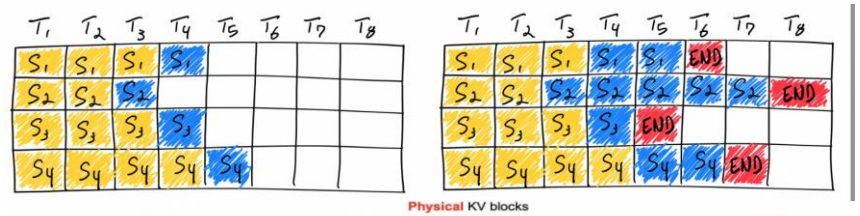
单张 GPU 显存
(数十 GB 级别)

单台机器的内存
(数 TB 级别)

KVCache计算方式演进

Orca

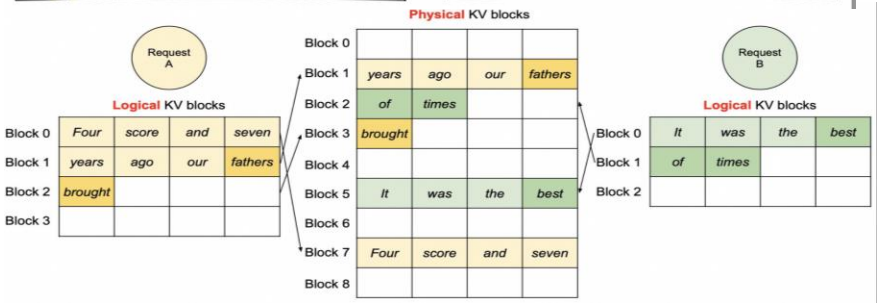
Continus Batching



通过切分任务打满算力

vLLM

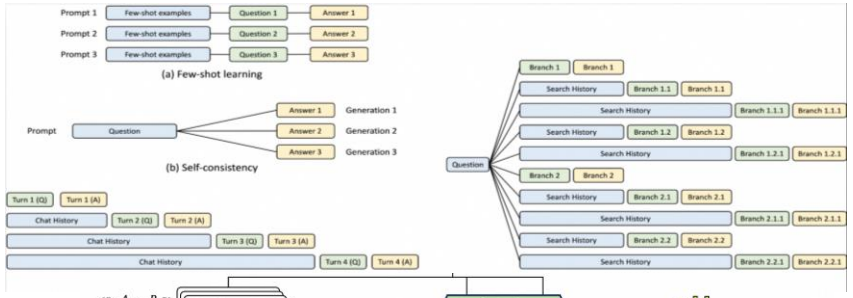
Page-Attention



Page粒度管理减少显存占用

SGLang

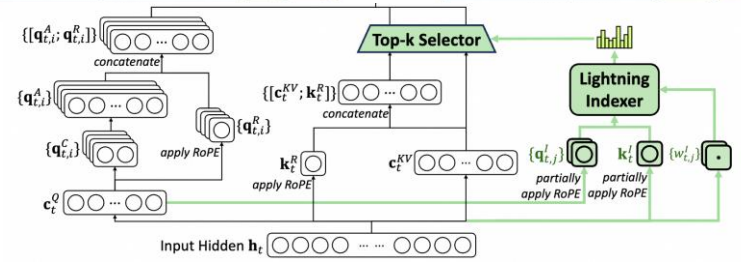
Prefix Attention



尽可能复用KVCache

DeepSeek NSA

Sparse KVCache

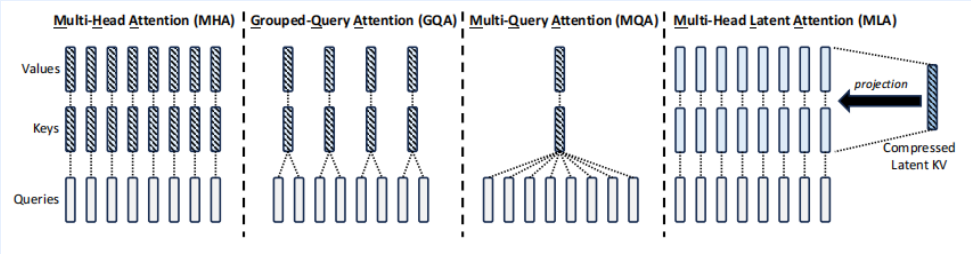


降低KVCache的产生量

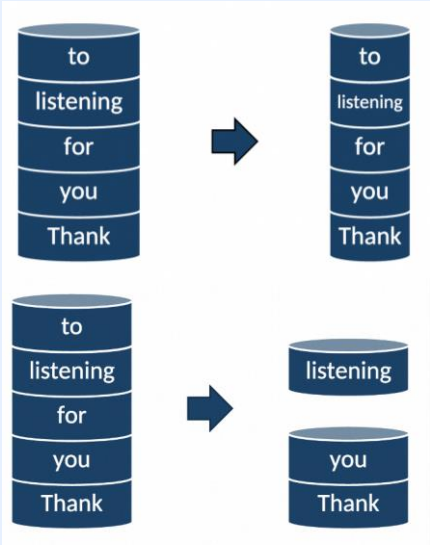
KVCache数据优化技术

模型级别优化

通过算法优化（Head/Dimension）减少KVCache产生量



KVCache压缩

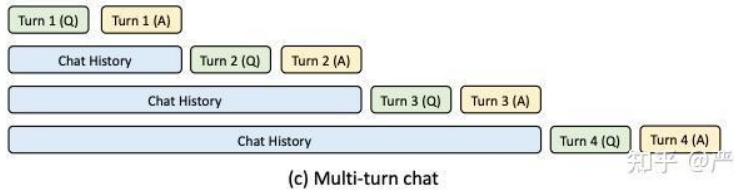


KVCache量化
使用低精度格式

KVCache消除
减少无关重要的数据

KVCache复用（Reuse）

- 在PagedAttention中，KV Cache只是在一个请求内复用，而没有做到跨请求的KV Cache复用
- 在多轮对话的场景下，下一轮的prompt其实刚好就是上一轮的prompt+completion



(c) Multi-turn chat

SGLang

原生支持

通过RadixAttention来实现Prefix Caching

first-come-first-serve，无法达到最优的缓存复用效果

cache-aware scheduling的调度算法

vLLM

Hash RadixAttention的方法，它使用哈希码作为物理KV Block的唯一标识

hash(prefix tokens + block tokens)

<-->

Logical KV blocks

->

Physical KV blocks

DeepSeek

上下文硬盘缓存技术，把预计未来会重复使用的内容，缓存在分布式的硬盘阵列中

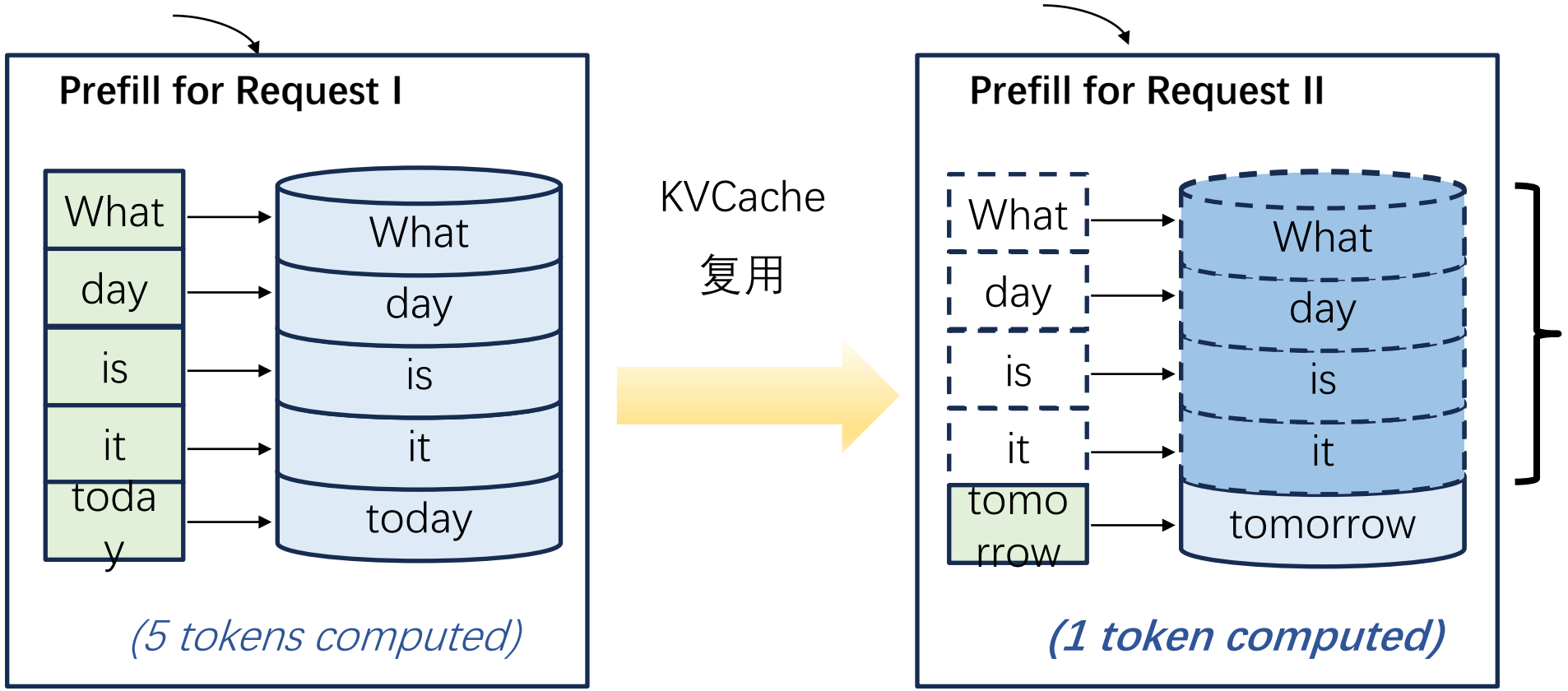
输入（每百万 tokens）		输出（每百万 tokens）
缓存命中	缓存不命中	2元
0.1元	1元	

KVCache共享和复用基本原理

以存换算基本原理： 前缀匹配的中间结果可被复用

“What day is it today”

“What day is it tomorrow”



前四个字符和第一个请求相同，因此**无需重复计算**可以直接复用中间结果

KVCache共享和复用基本原理

- 大模型辅助读论文场景为例

Cool Papers - Immersive Paper Discovery

所有查询可复用

热点论文复用率高

System Prompt: 你是一个论文阅读助手，…

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving

Ruoyu Qin♠♡1 Zheming Li♠1 Weiran He♠
Mingxing Zhang♡2 Yongwei Wu♡ Weimin Zheng♡ Xinran Xu♠2
♠Moonshot AI ♡Tsinghua University

User Question 1: 总结一下这篇论文

User Question 2: 这篇论文关键创新是什么

User Question 3: 有哪些相关研究?

User Question 4: 可以进一步探索什么?

可复用长度长

问题短

不同的用户会对同一篇论文进行不同角度的提问，只要能将共享的可复用的部分保存下来多次复用就可以大幅度降低算力开销

03

以KVCache为中心的解耦架构

不同解耦架构带来的优势

Mooncake

将Prefill与Decode解耦，让“算力-吃紧”与“带宽-吃紧”任务各得其所，推理延迟-吞吐双降

PD分离

StepFun

把Attention 计算和 Expert 路由/计算分到异构硬件，CPU缓存KV、GPU稀疏激活专家，MoE模型显存换带宽，成本直降

AF分离

Qwen2.5-VL

把多模态编码、KV生成、自回归解码拆成三阶段流水线，单阶段内存爆降95%，批尺寸×20，长图长视频秒级TTFT。

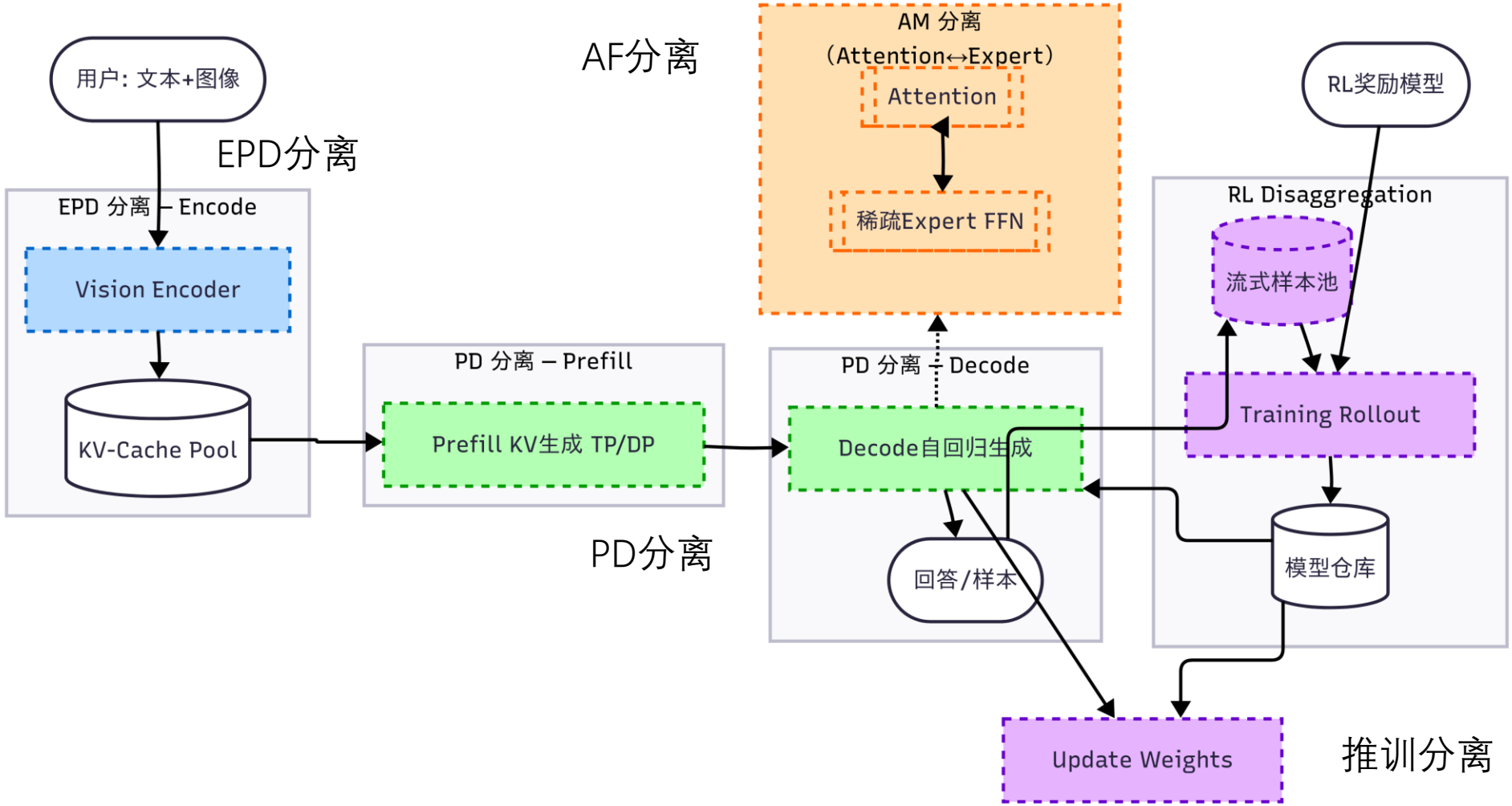
EPD分离

verl

把RLHF中的在线样本生成（推理）与策略梯度更新（训练）彻底分开，独立扩缩容、异构部署，训练吞吐提升2.6×，成本再降30%。

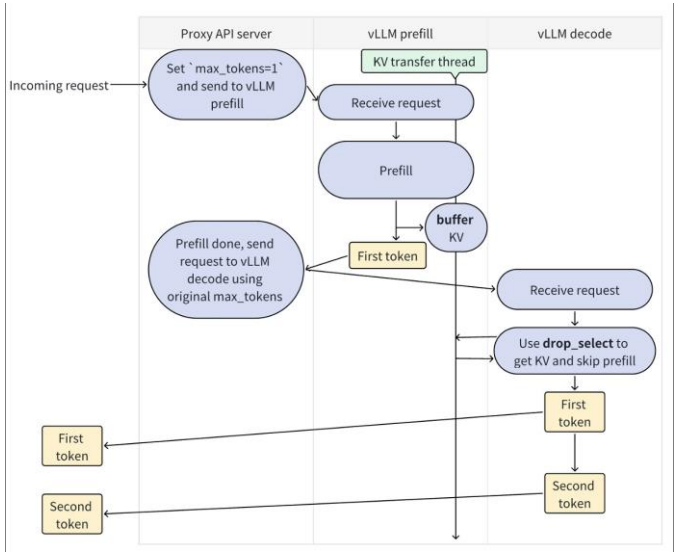
推训分离

不同解耦架构在推理训练框架中的位置

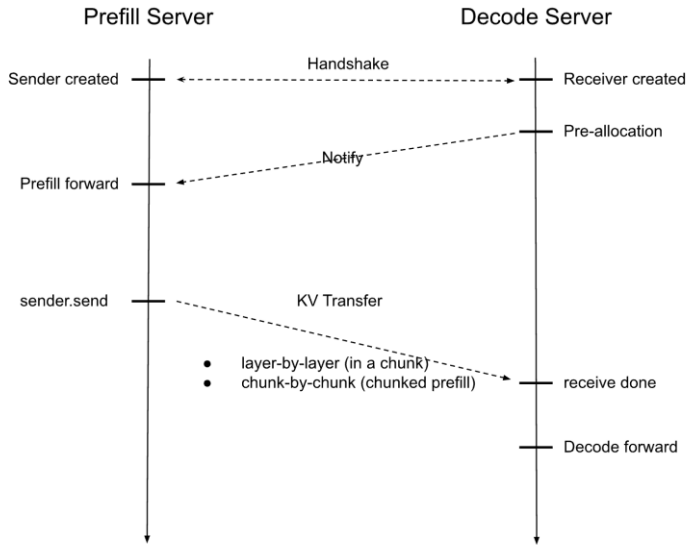


基于解耦架构的KVCache共享和复用

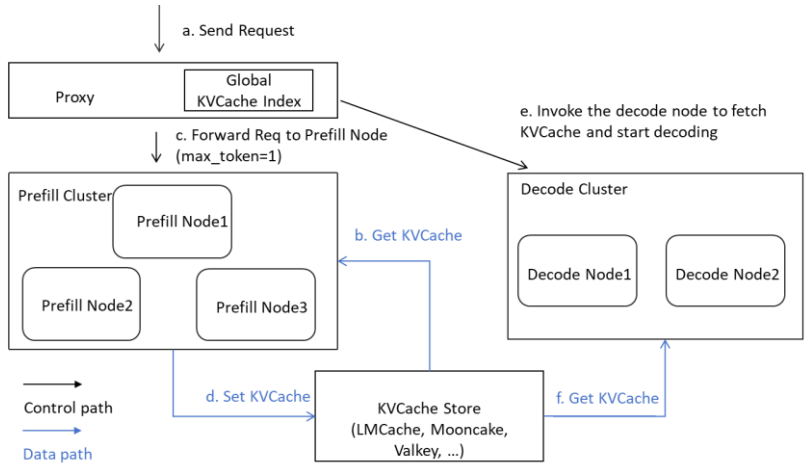
点对点传输（CPU Offload）



点对点传输（GDR）



Store接口

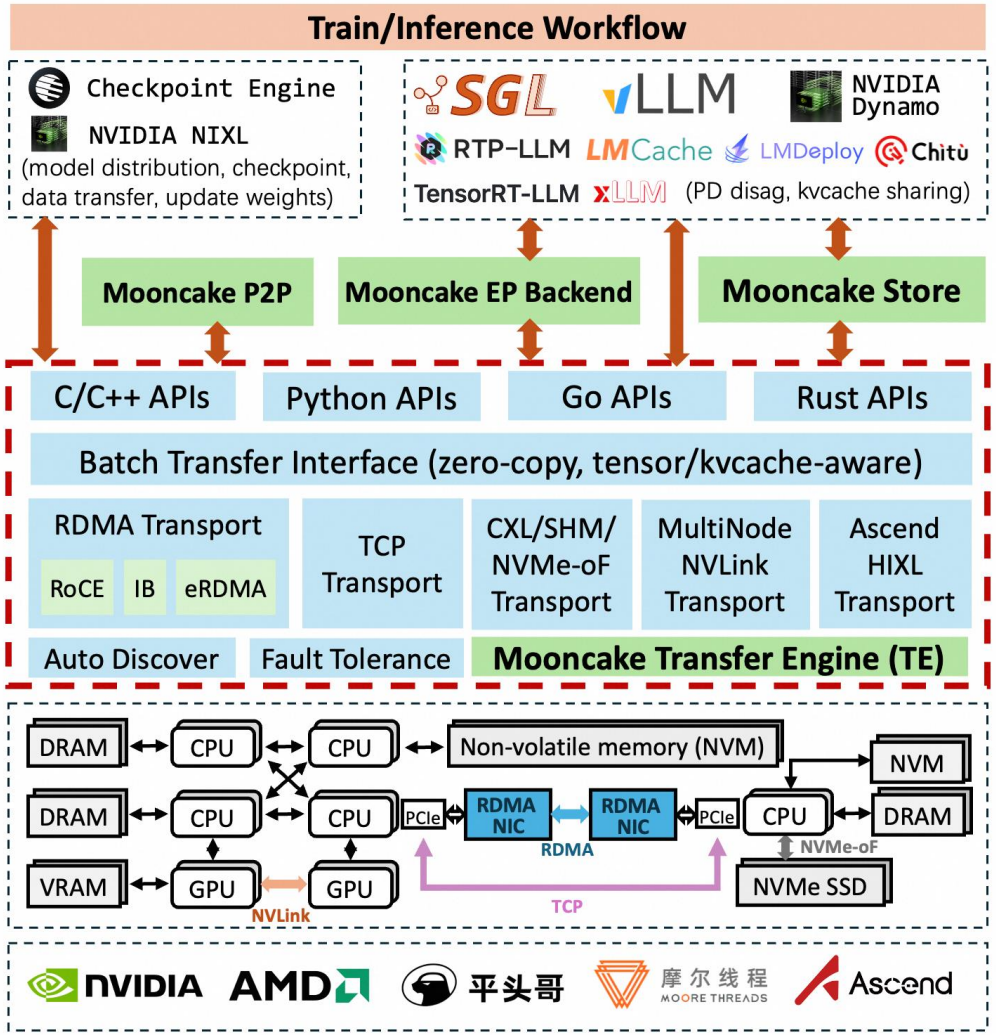


维度	LayerWise	Chunk Level	全量传输
吞吐量	中	高	低
扩展性	强	中	弱
实现难度	复杂（动态调度）	中等（块管理）	简单

04

KVCache开源项目：Mooncake

Mooncake项目架构



Transfer Engine：

- 全链路零拷贝、多网卡池化
- 最高8*400Gbps聚合带宽、拓扑感知
- 故障容错，负载均衡，多协议支持。
- 更充分地发挥高性能网卡的优势，
- 相比 nccl 更加灵活，
- 支持动态拓朴、故障容错

KVCache Store：

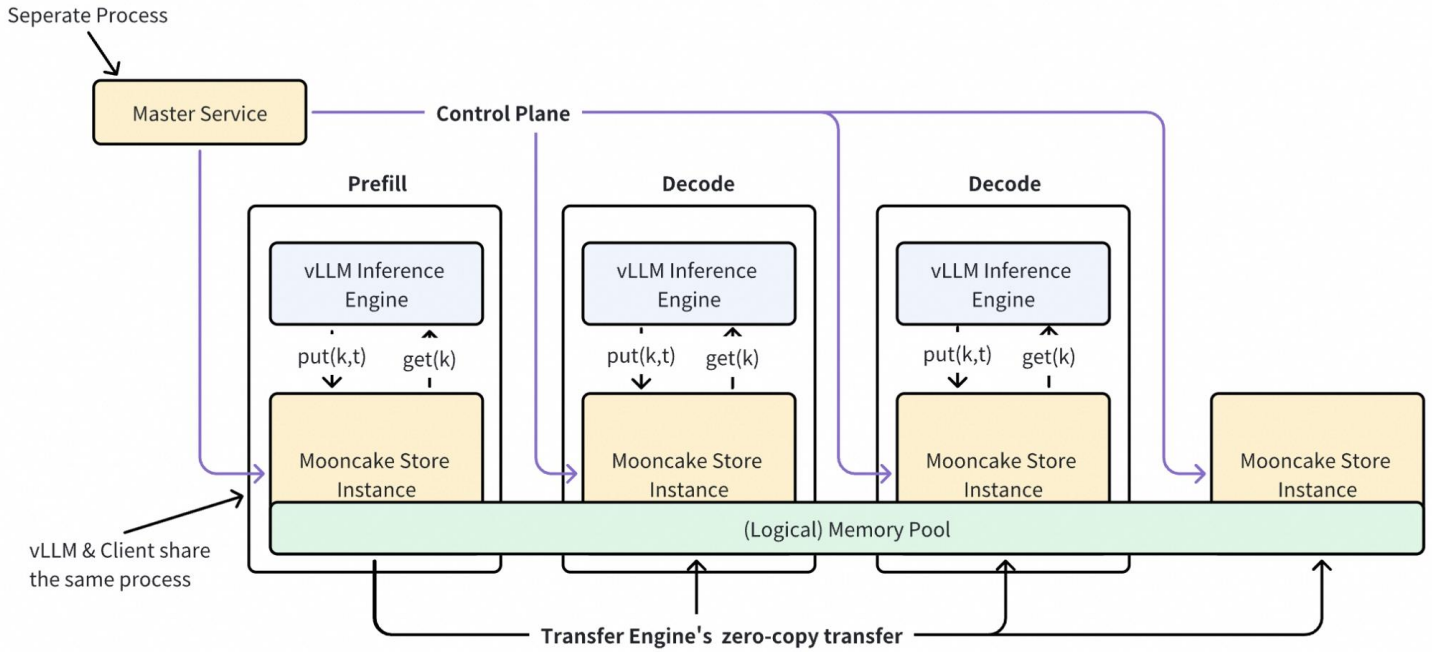
- 充分利用GPU 集群中闲置的内存和带宽
- 省成本的同时降低响应延迟
- 透明多级缓存，VRAM/DRAM/SSD/Remote
- 进一步下沉到底层廉价存储

Production Ready:

- AC2镜像 + K8S部署
- 提供whl包一键安装
- Nightly Build & End-to-end CI Test

Mooncake Store

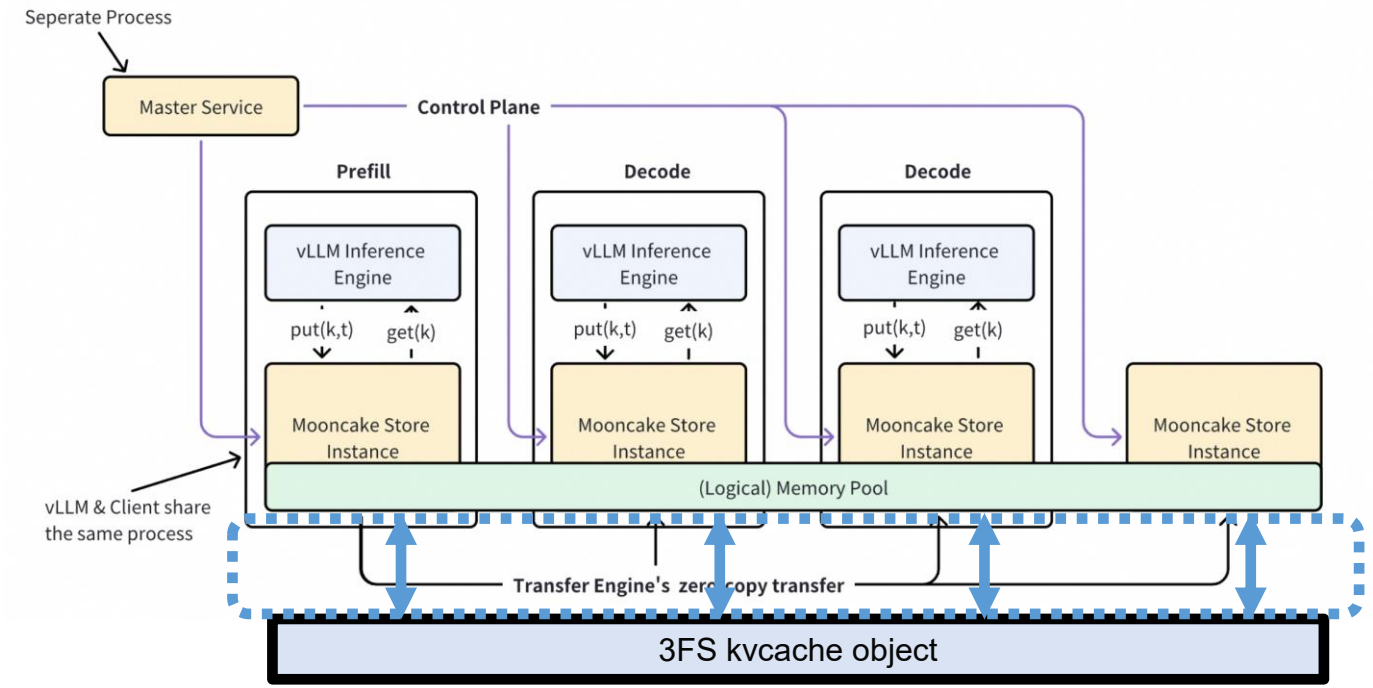
- 与传统缓存系统的区别:
  - Key 由 value 通过哈希计算得到，无需 update操作
  - 支持Lease，没有版本管理的需求
- 灵活性与可定制性:
  - 提供底层对象存储和管理功能。
  - 具体缓存策略由上层框架/用户实现（如 vLLM）



- 系统特性:
  - 提供KV复用率，TTFT等指标
  - etcd可靠性服务，高可用模式
  - 数据持久化能力
  - 提供分层存储能力（NFS/3FS）
  - 提供Restful API/独立服务
  - 多语言接口/零拷贝能力

Mooncake Store 分层存储

- 存储：辅存上的KVCache object如何组织？（映射关系，抽象粒度？）
- 接入：DFS的能力如何“优雅”地接入store？（写盘时机，与evict结合？）
- 管理：逻辑层谁来负责管理3fs上的kvcache？（client or master管理？）



- **问题1**：get每次只进行单个kvcache数据的读取，不能充分利用3FS的并发读取性能？
- 解决思路：使用**batchget**操作，多线程同时读取多个kvcache数据。

- **问题2**：使用通用的POSIX接口需要经过fuse客户端，过程中存在多次上下文切换和数据拷贝
- 解决思路：引入3fs高性能读写接口**USRBIO支持**，并进行优化，以实现高吞吐、低延迟的数据读写。

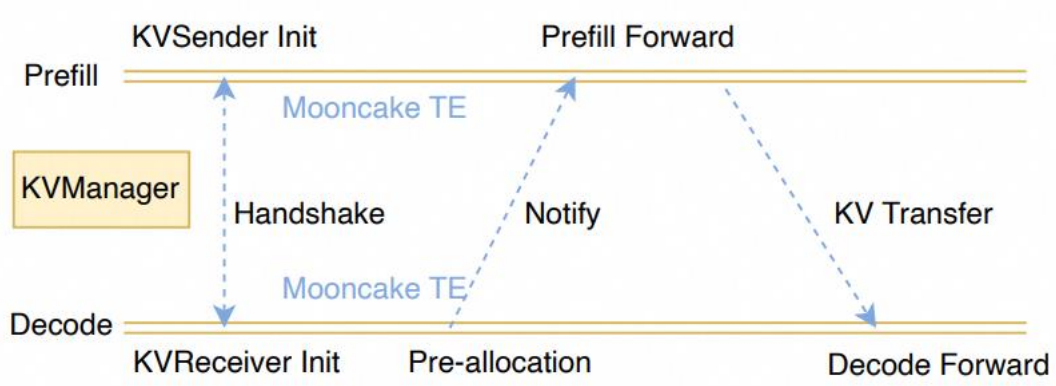
- **问题3**：client端负责元数据管理，带来了数据一致性问题，查询文件系统也会带来性能开销
- 解决思路：将ssd的metadata元数据信息管理从client迁移至master，**使master统一管理disk和memory两种类型的元数据信息**。

Mooncake对于大模型推理生态支持

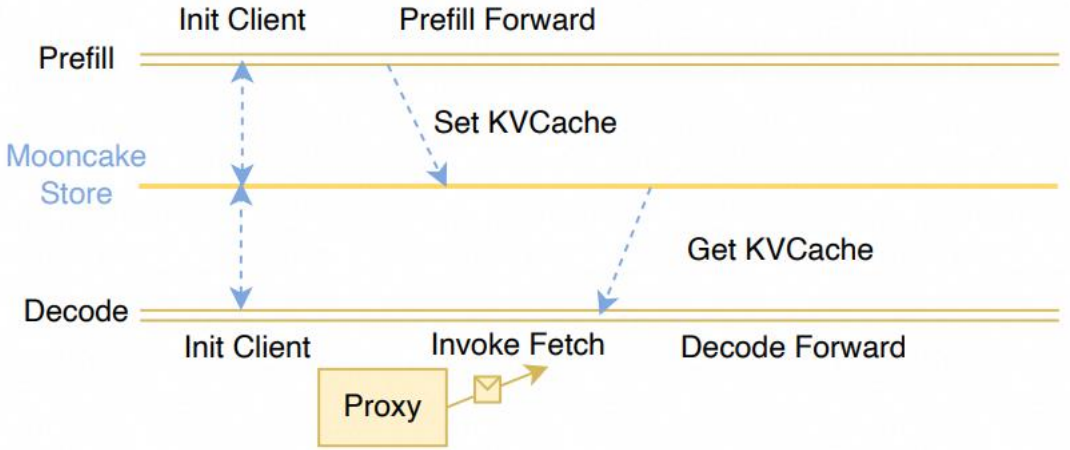
Feature	Project	Type	Transfer	KVCache Store	Ckpt Engine
Inference	SGLang	Inference	✓	✓	✓
	vLLM V0	Inference	✓	✓	✓
	vLLM V1	Inference	✓	✓ (w/ LMCache)	✗
	LMDeploy	Inference	✓	✓	✗
	Chitu	Inference	✓	✗	✗
	RTP (Alibaba)	Inference	✗	✓	✗
Middleware	TBase (Ant)	Middleware	✓	✗	✗
	Dynamo	Framework	✓ (w/ Nixl)	✗	✗
	LMCache	Middleware	✗	✓	✗
RL Training	Slime	RL Training	WIP	WIP	✓

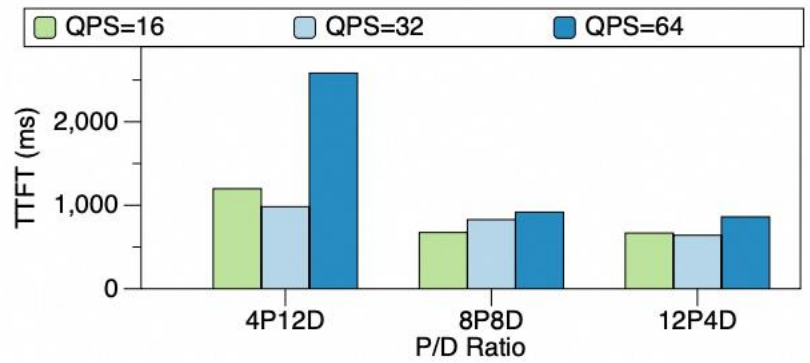
Mooncake 对于大模型推理生态支持（PD分离）

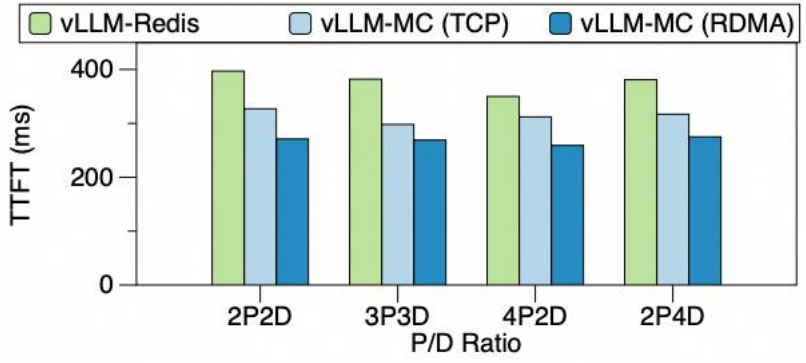
基于Transfer Engine的SGLang PD分离



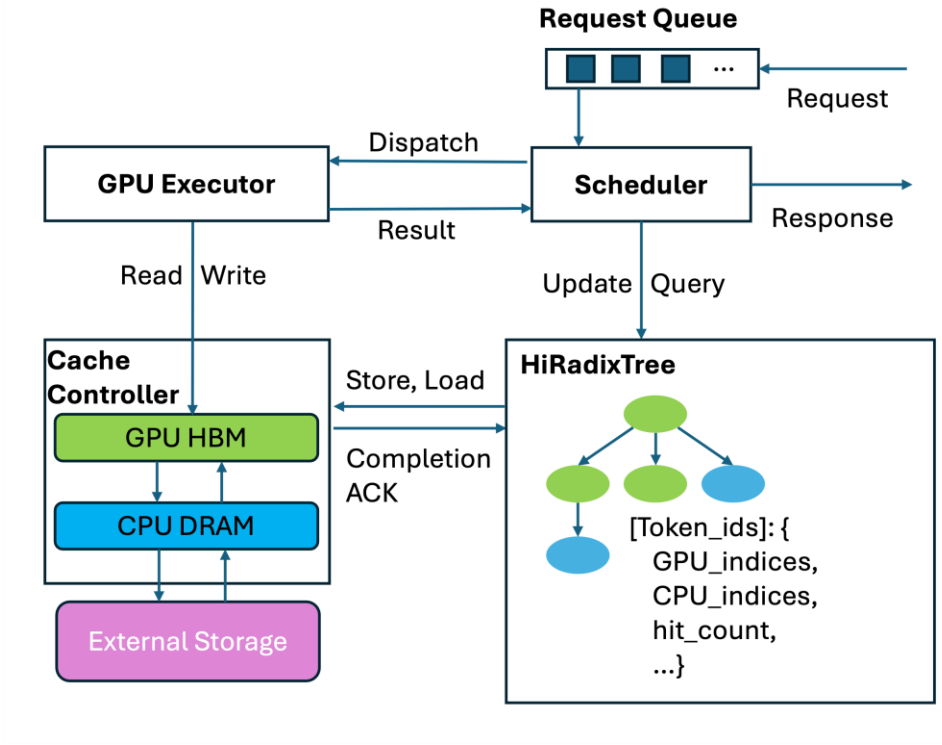
基于Store的vLLM PD分离



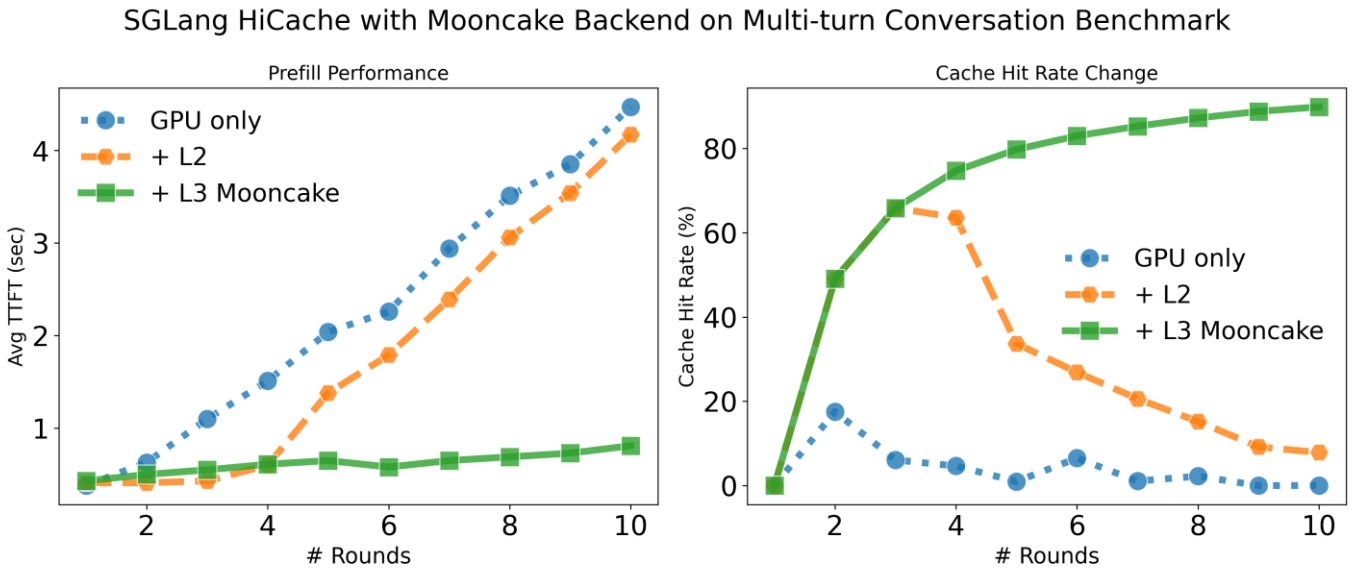




Mooncake Store + HiCache（KVCache Offloading）



- **流水线掩盖：**将 GPU 计算与 KV 数据传输（H2D/D2H）重叠执行，智能预取**隐藏 I/O 延迟**
- **零拷贝：**利用 Mooncake 的 RDMA 机制与 GPU Direct 技术，**绕过 CPU 拷贝**，降低通信开销



引入Mooncake作为L3，实现**跨机 KV 共享**与持久化，支持无限长上下文（Infinite Context），解决多实例间的重复计算问题
HiCache高性能IO Kernel

Mooncake Store + LMCache/vLLM (Orchestration)

Mooncake × LMCache

LMCache x Mooncake: Unite to Pioneer KVCache-Centric LLM Serving System

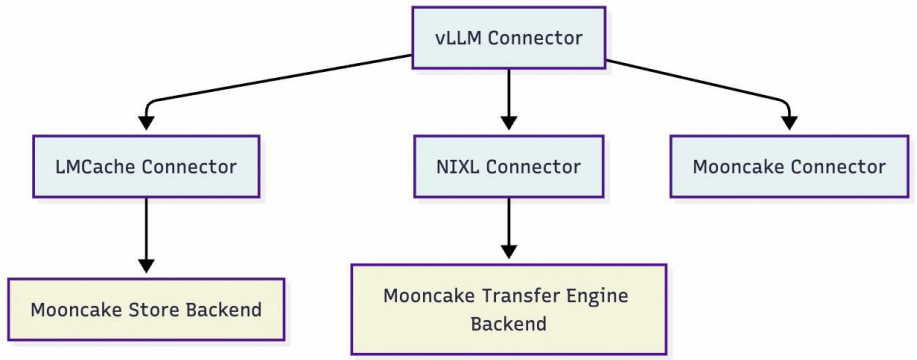
By **Mooncake and LMCache Team**

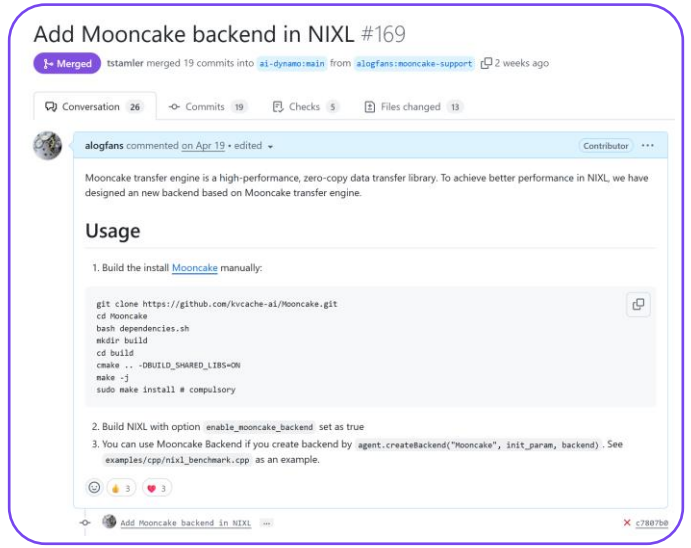
Posted on May 8, 2025

Metric	Cold Start (First Round)	Cache Hit (Second Round)	Improvement
Average TTFT	21,707.62 ms	6,708.39 ms	↓ 69.1%
P50 TTFT	22,102.51 ms	7,253.38 ms	↓ 67.2%
P90 TTFT	38,170.54 ms	11,128.26 ms	↓ 70.9%
Average TPOT	368.12 ms	140.17 ms	↓ 61.9%
P50 TPOT	362.08 ms	132.98 ms	↓ 63.3%
P90 TPOT	632.90 ms	221.93 ms	↓ 64.9%
Request Throughput (req/s)	1.11	3.23	↑ 191.0%
Output Token Throughput (tok/s)	71.24	202.91	↑ 184.8%
Total Token Throughput (tok/s)	10,899.84	31,665.01	↑ 190.5%

Two methods Support vLLM v1

- LMCache Storage Backend KVCache Reuse
- NIXL Plugin for P/D





06

云原生下的KVCache

当前挑战：显存墙与架构演进

当前挑战：显存墙与架构演进

- **显存压力：** 在长上下文和高并发场景下，KVCache 显存占用常超 70%，单机 GPU HBM 和 CPU DRAM 已难以为继 。
- **架构演进：** 推理架构正从单体向分布式演进（Prefill-Decode 分离、KVCache 外置），以实现跨请求缓存共享与弹性伸缩 。
- **运维痛点：** 传统的 K8s Workload 难以处理推理角色间的强依赖与拓扑感知，且滚动升级常导致缓存丢失与性能抖动 。

核心解决方案：RBG + Mooncake

- **Mooncake (存储引擎)：** 分布式 KVCache 存储，提供高吞吐、低延迟的 L3 缓存服务 。
- **RBG (编排引擎)：** RoleBasedGroup 是面向 AI 推理的 K8s 原生 API，统一管理多角色协同、部署与弹性 。
- **价值主张：** 将 Mooncake 作为 RBG 编排下的补充角色，实现性能最大化（L3 缓存）与生产级稳定性（原地无感升级） 。

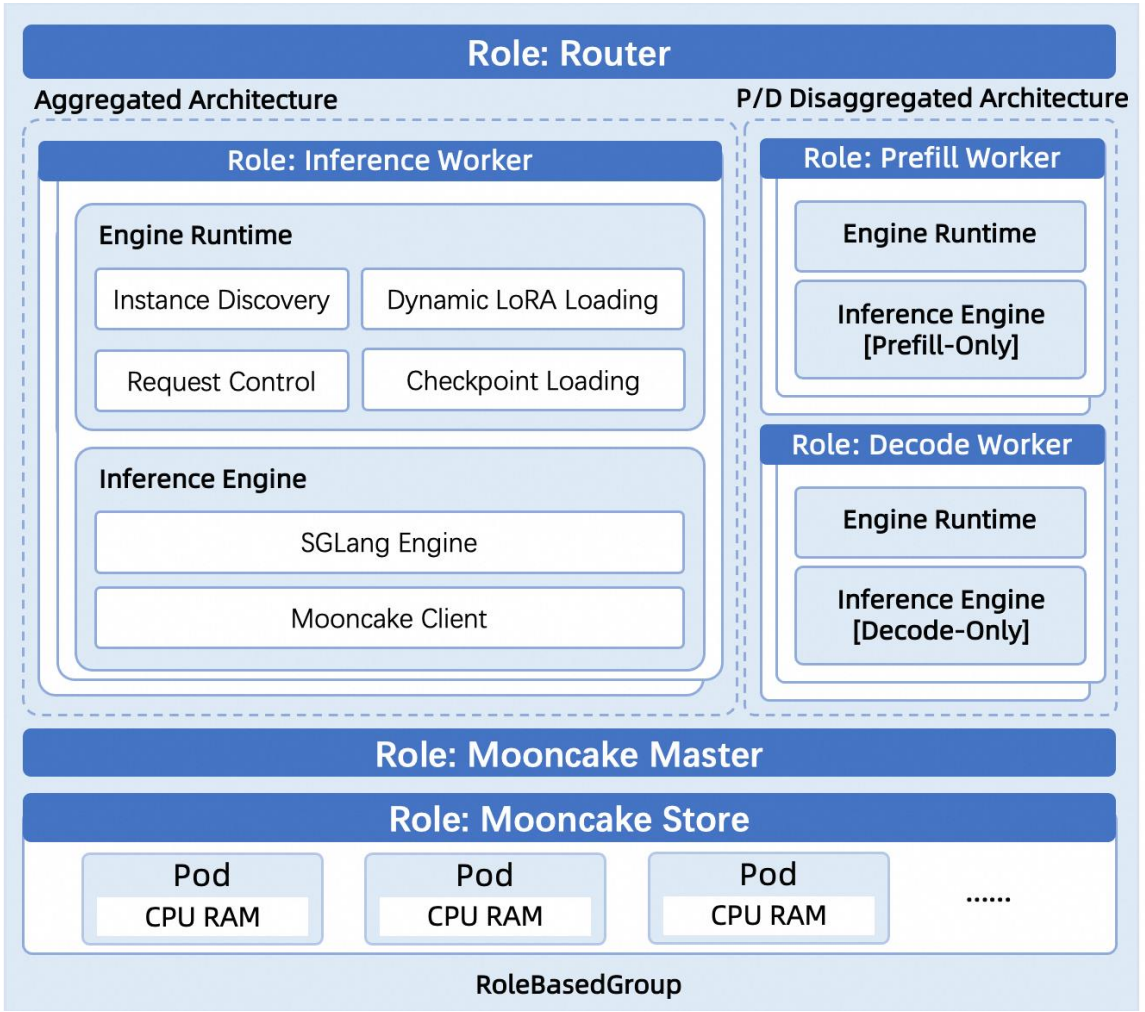
核心技术双引擎：Mooncake 与 RBG

技术深度解析——存储加速与智能编排

组件	Mooncake (分布式 KVCache 存储)	RoleBasedGroup (弹性角色编排)
定位	SGlang 的高性能分布式 L3 存储后端	视推理服务为“角色有机体”而非孤立 Pod 的编排引擎
核心特性	**RDMA 加速 + 零拷贝：**实现高带宽、低延迟访问 **智能预取：** 支持 GPU 直传，最大化 I/O 效率 **PD 分离支持：** 提升大规模集群 Token 吞吐量	**SCOPE 能力框架：** • **S**table: 拓扑感知的确定性运维 • **C**oordination: 跨角色协同（如部署、升级） • **P**erformance: 拓扑感知调度 (NVLink/PCIe 优先)

RBG 的设计理念： “角色（Role）”即一等公民，解决大模型推理作为“有状态、强拓扑”应用的矛盾

部署架构：Router + RBG + SGLang + Mooncake Store



系统角色构成

- **SGLang Router：** 统一入口与流量调度，智能分发至 Prefill 或 Decode 后端
- **Prefill Worker：** 计算密集型，负责 Prompt 前向计算并生成初始 KVCache
- **Decode Worker：** 延迟敏感型，依赖 KVCache 进行 Token 逐个解码
- **Mooncake Store/Master：** 独立外置存储角色，持久化存储 KVCache

协同工作流

- 所有角色通过 RBG 进行紧密集成与协同
- 利用 RDMA 网络在计算节点与 Mooncake 存储节点间高速传输 KVCache
- Router 根据负载动态调度，实现动态批处理与连续批处理

解决行业难题——原地无感升级 (In-place Update)

痛点：传统的滚动升级风险

- Mooncake 作为有状态服务，传统 Pod 重建会导致内存中 KVCache 丢失
- 后果：活跃会话中断，被迫重新 Prefill，导致 P99 延迟毛刺飙升，吞吐量断崖式下跌

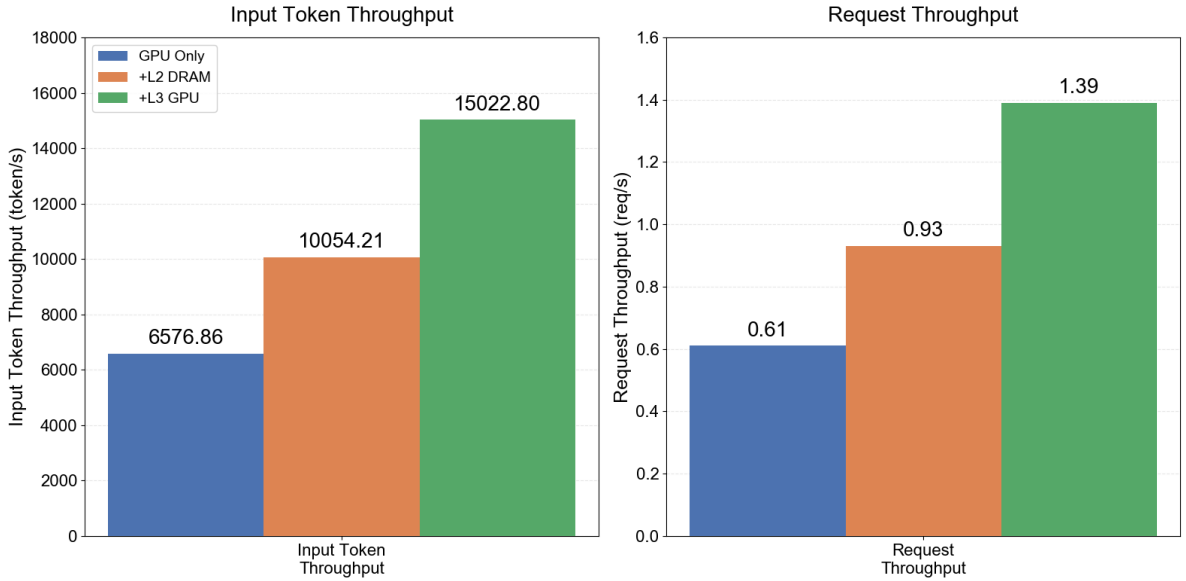
解决方案：RBG 原地升级 + 持久化

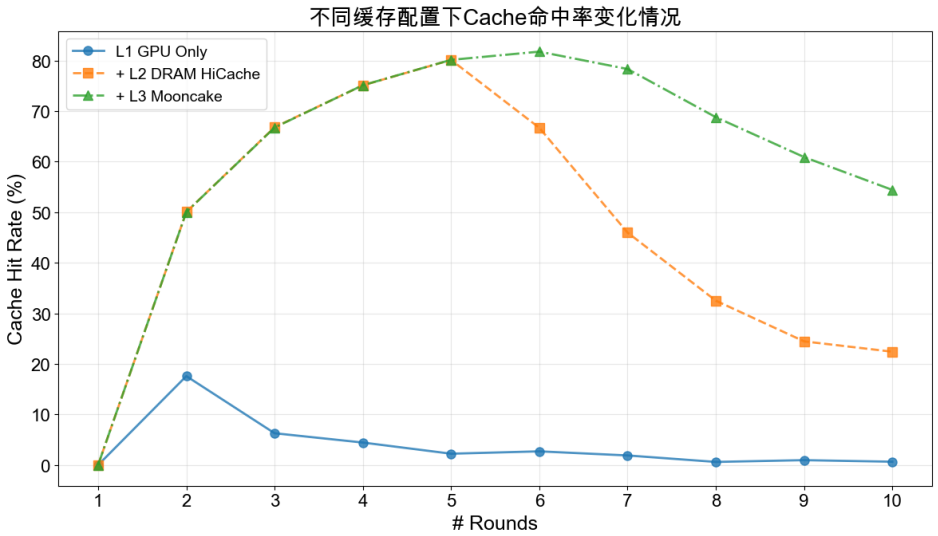
- **Mooncake 本地持久化：**支持KVCache 快照至共享内存/本地盘，进程重启后快速恢复数据
- **RBG 原地升级策略：**K8s 原生 API 实现原地替换容器镜像，复用节点资源，避免 Pod 重建

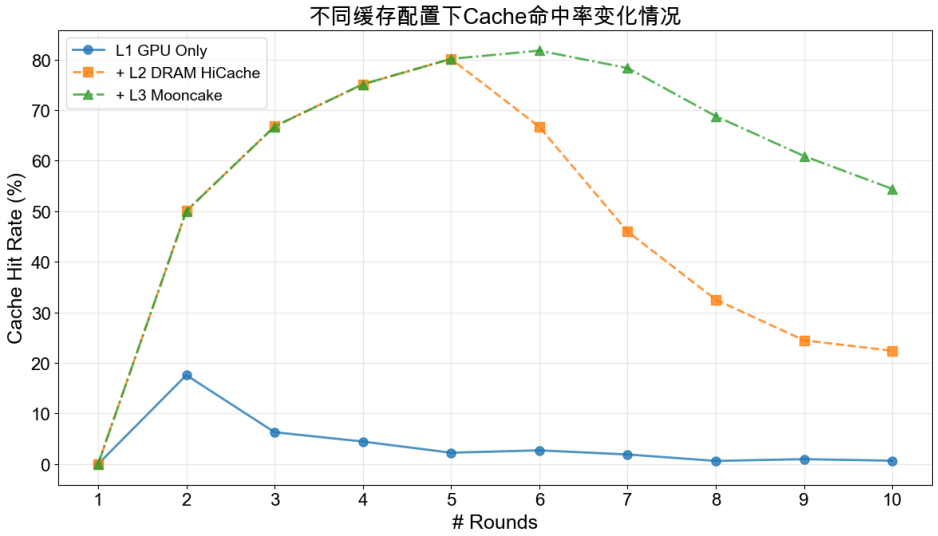
最终效果

- **升级无感，服务不抖：** 彻底避免缓存失效导致的重计算，保障端到端稳定性
- **运维自动化：** 将复杂的分布式协同升级转化为标准化的运维能力

ACK场景下部署效果







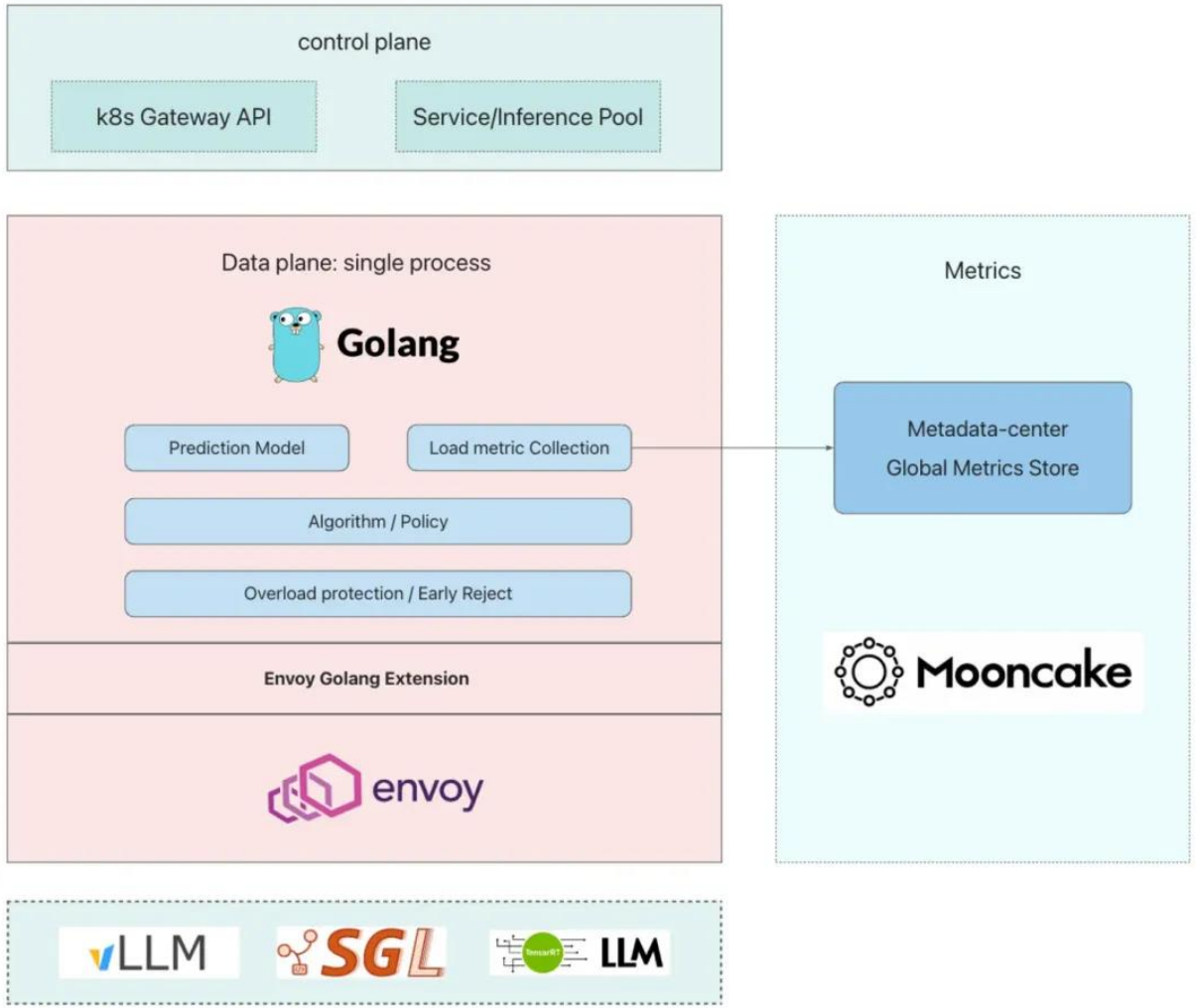
结论

引入 L3 Mooncake 缓存后，能够显著提升缓存命中率，大幅降低延迟并提升系统整体吞吐量，实现性能与成本的平衡

06

KVCache智能化调度

AIGW网关架构

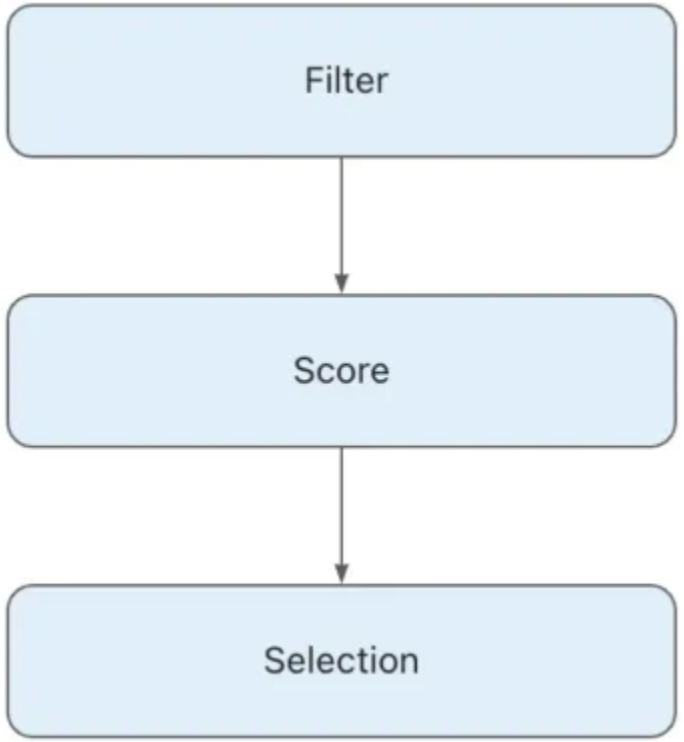


- 灵活且强大的 Envoy Golang 扩展
  底层 Proxy 能力基于标准 Envoy 和 Istio 构建
  使用 Golang 来实现复杂调度的能力
- 基于Metadata Center 实现准实时负载指标收集
- 大规模场景的多因子综合权重负载均衡策略
- 水平可扩展的高可用架构设计。

多因子综合权重负载均衡策略

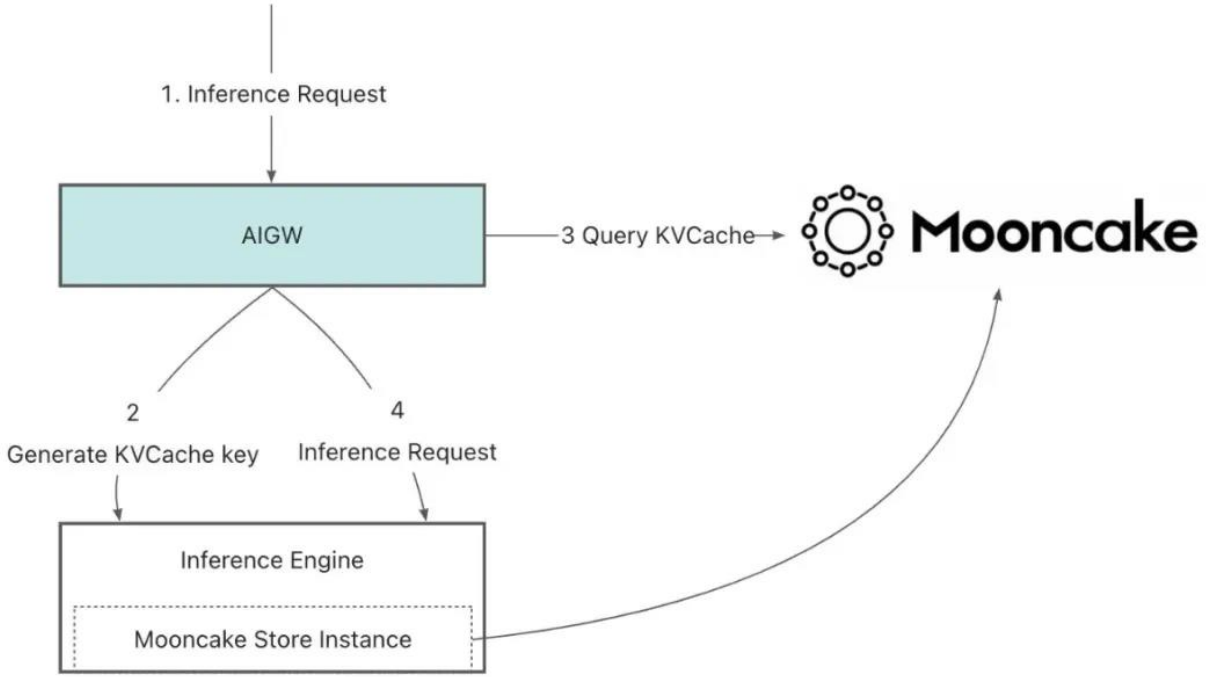
score = W1*cache_ratio - W2*request_load - W3*prefill_load

- **cache_ratio**: Prefix Cache Hit Rate
- **request_load**: Number of Requests
- **prefill_load**: Length of Prompt in Prefill Phase



- 综合权重算法：缓存命中率、请求数量、 Prefill负载三个
- 选取Top K节点：缓存命中率越高、请求数量越低、Prefill负载越轻的节点得分越高
- 整体策略导向为KVCache亲和性优先和Prefill优先，同时兼顾请求数量的均衡性。

精确KVCache Aware算法



问题背景

近似 cache-aware 的方案，引入一定的 KVCache 命中率误差

工作流程

- 由引擎暴露一个新接口来生成 KVCache key
- 网关通过调用这个接口获取到 KVCache key,
- 从 Mooncake master 节点查询不同节点KVCache 命中率

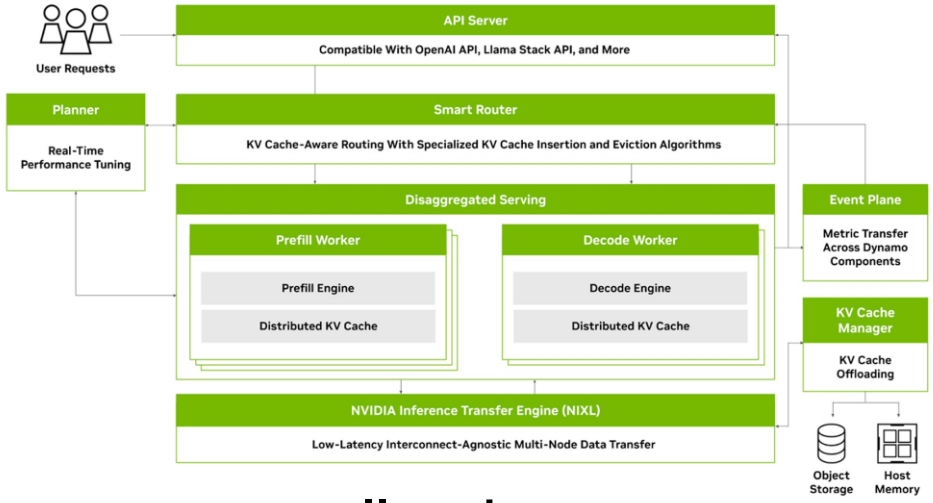
对比于独立的 KVCache Indexer 组件

- 通过复用 Mooncake Store Master
- 减少 KVCache Location 信息的传输
- 减少一个独立组件的维护

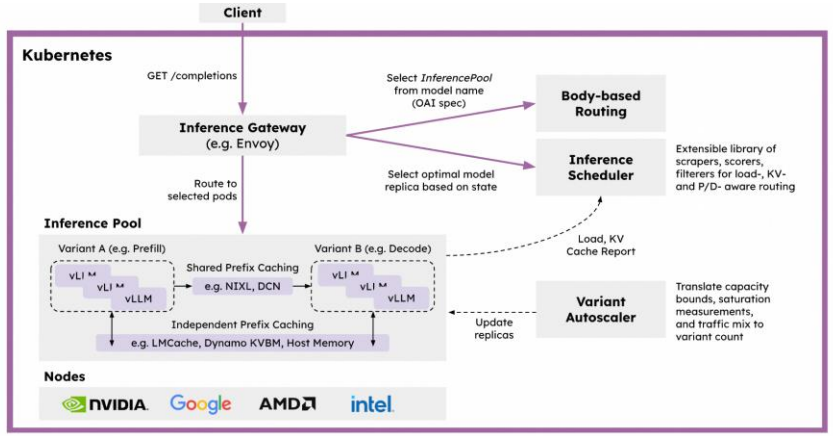
05 Mooncake开源生态

已有AI Production Stack开源生态

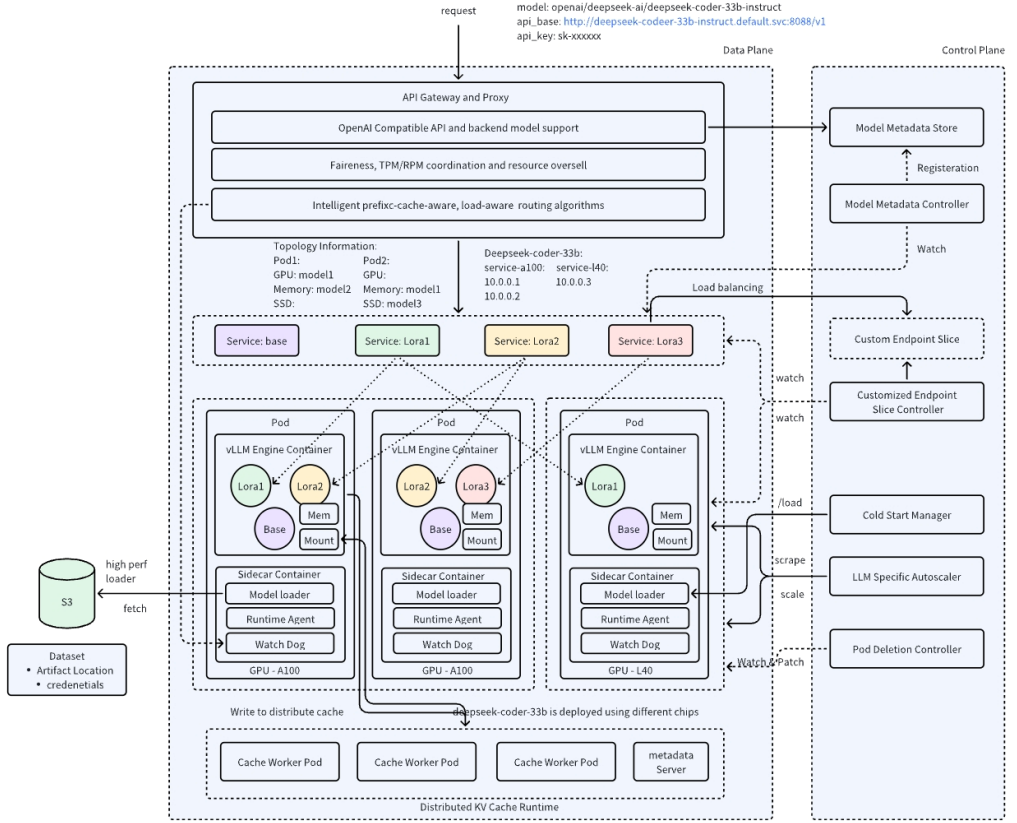
Dynamo



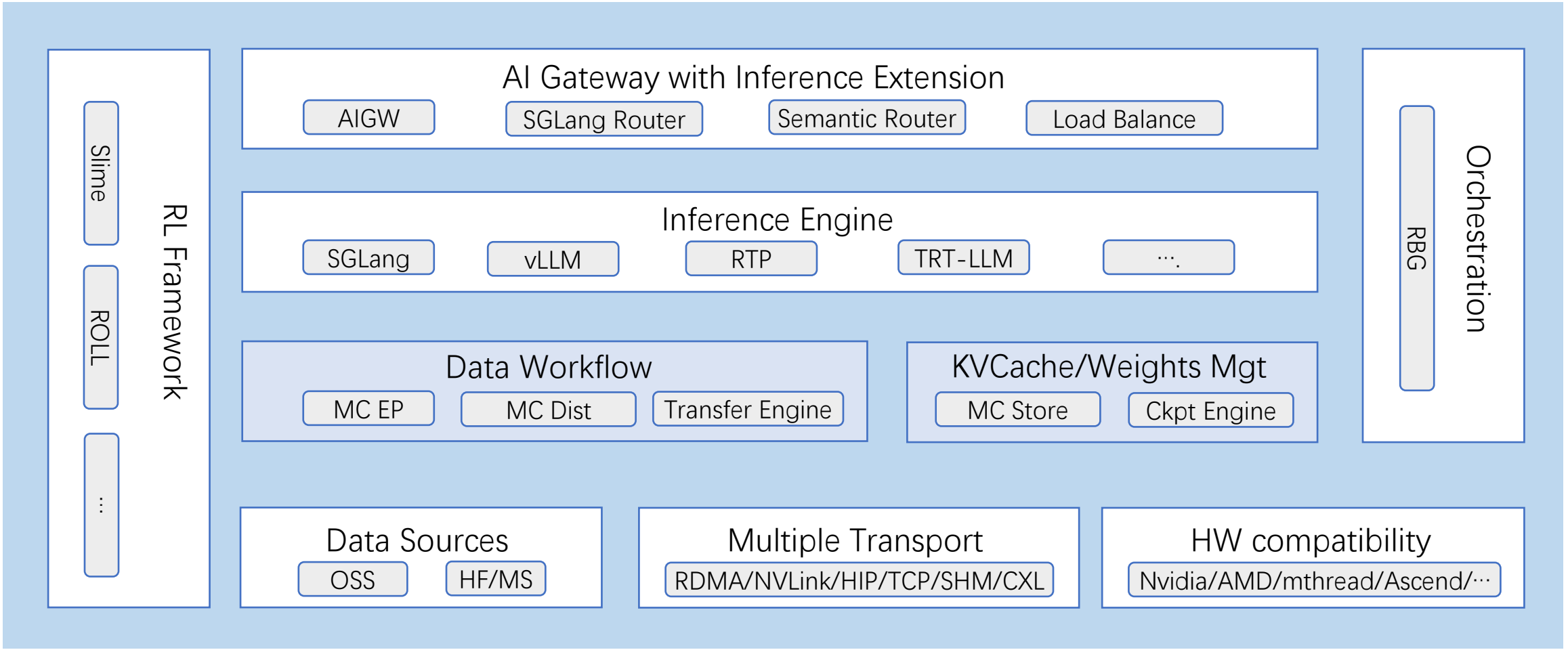
llm-d



AIBrix



Mooncake AI Production Stack



Mooncake开源发展时间线

从标杆应用 → 到产业推广

2024 年 3 月Kimi 凭借长文本处理能力出圈
Mooncake 上线并经受住流量剧增的考验

2024 年 6 月 Mooncake 技术报告公布
引发业界广泛讨论

2024 年 11 月 Mooncake 开源，应用于阿里/蚂蚁
与 vLLM/SGLang 携手共建分离式大模型推理平台

2025 年 2 月
USENIX FAST 最佳论文

被 GTC 2025 Keynote 黄仁勋专门介绍的 Dynamo
分布式推理系统所使用

4.1K ☆
100 Contributors

支持Kimi 1.5/2, 支持Ascend
最早支持DeepSeek大规模推理

Kimi 底层推理平台

- 承载其近四千万的月活用户
- 以 KVCache 为中心的 P/D 分离架构
- 通过以存换算的思想，提升 Kimi 吞吐 75% 以上

Mooncake + vLLM SGL
LMCache LMDeploy RTP-LLM

- 和国际最主流的开源推理框架共建分布式推理架构的标准参考实现

USENIX FAST 最佳论文

kvcache-ai/Mooncake --月之暗面和清华大学共同发起，多家大模型和 Infra 厂商联合共建的开源项目

Moonshot AI　清华 MADSys 实验室
9#AI SOFT　HUAWEI　阿里云
火山引擎　蚂蚁集团 ANT GROUP　趋境科技 APPROACHING.AI
金山云　面壁智能 MODELBEST　NVIDIA
and more ···

极客邦科技 2026 年会议规划

促进软件开发及相关领域知识与创新的传播

参会咨询

查看会议

4月 · 北京 · 1200人

QCon 全球软件开发大会

会议时间：4月16-18日

- Agentic Engineering
- AgentOps
- 下一代模型架构与推理优化
- AI 原生基础设施
- 知识工程实践
- AI 安全

6月 · 上海 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：6月26-27日

- AI Infra 系统工程
- 多 Agent 协作与实践
- 多模态融合
- 模型训练与推理创新
- 数据平台与特征服务

8月 · 深圳 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：8月21-22日

- Agentic AI
- 轻量化与高效推理
- 多模态应用
- AI + IoT 场景实践
- AI 工业化落地

10月 · 上海 · 1200人

QCon 全球软件开发大会

会议时间：10月22-24日

- AI Agent
- Vibe Coding
- 智能可观测
- 推理基建
- 模型攻防
- AI x 创造力

12月 · 北京 · 1000人

AiCon 全球人工智能开发与应用大会

会议时间：12月18-19日

- 大模型架构创新
- 多模态 AI 产业融合
- 具身智能
- AI for Science
- 大模型安全

InfoQ 极客传媒

06

以KVCache为中心的生态总结

06

开源生态一些思考

InfoQ 极客传媒

THANKS

探索 AI 应用边界

Explore the limits of AI applications

AiCon

全球人工智能开发与应用大会

Mooncake广泛的开源影响力

vLLM

- 被最广泛被使用的推理引擎，应用于各大云厂商 - vLLM
- 其分布式推理实现基于 Mooncake 实现

v0.8.3 Latest

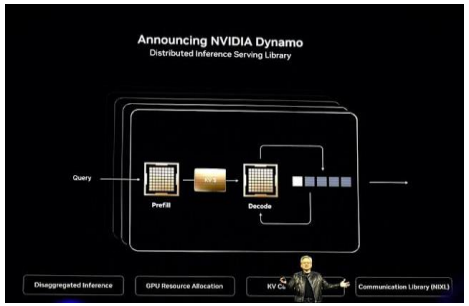
github-actions released this 2 days ago · 57 commits to main since this release · v0.8.3 · 296c657

Cluster Scale Serving

- Support XpYd disaggregated prefill with MooncakeStore (#12957)

NVIDIA Dynamo

- 黄仁勋在 GTC 2025 Keynote 中重点发布的分布式推理系统 – Dynamo
- 其架构参考 Mooncake 设计并在文档中专门致谢



Acknowledgement

We would like to acknowledge several open source software stacks for motivating us to create Dynamo.

- vLLM and vLLM-project
- SGLang
- DistServe
- Mooncake

- *Memory bottlenecks*: Large-scale inference workloads demand extensive capacity. KV cache offloading across memory hierarchies (HBM, DDR, N memory limits and speeds up latency. (Mooncake, AIBrix, LMCache)

SGL

- xAI 底层推理引擎，被广泛应用于 DeepSeek 推理 - SGLang
- 和 Mooncake 合作实现其分布式推理架构

LMSYS Org @lmsysorg

SGLang has achieved a milestone with full support for PD Disaggregation, thanks to the MoonCake team, Teng Ma, Shangming Cai, Xuchun Shang and Yuan Luo were instrumental in this achievement. Special thanks to Atlas Cloud for their support with the H100s cluster. Let's go! 🚀

KIMI SGL Mooncake

Serving Kimi K2 with SGLang and OME

Deploying Kimi K2 with PD Disaggregation and Large-Scale Expert Parallelism on 128 H200 GPUs

by: The Mooncake Team, July 20, 2025

Introduction: Deploying the Most Advanced Open-Source MoE Model Kimi K2 is currently the most advanced open-source Mixture-of-Experts (MoE) model available. Released by Moonshot AI in 2025, it features: 1 trillion total parameters 32 billion activated parameters per token 384 experts with dynam...