亚马逊云科技
中国峰会

Mooncake on EFA：万亿参数模型背后的开源服务架构实践

把 RDMA 级 KV Cache 传输带到亚马逊云科技 EFA

王鹤男

资深应用科学家

亚马逊云科技

汪其香

解决方案架构师

亚马逊云科技

Agenda

1. **Mooncake： 开源的 以KVCache 为中心的推理**
2. 亚马逊网络组件**EFA**，对比 **IB / RoCE**
3. **Transfer Engine on EFA 的设计**
4. **与 vLLM / SGLang 的整合 + 训练场景（RL 权重同步）**
5. **Mooncake vs NIXL**
6. **实测数据分享** — Kimi K2.6 / MiMo-V2-Flash / GLM-5.1

Agentic AI 对推理服务提出全新要求

AGENTIC AI RAISES ENTIRELY NEW DEMANDS ON SERVING

01 / TOOL CALLS

模型连续调用上百次工具,意味着 ——

→超长上下文 与 重复 Prefill
单次会话上下文极长,且需反复重新编码

→ 持续高吞吐负载
不再有波峰波谷,而是稳定、不间断的高压力

02 / DEPLOYMENT

传统同进程部署方式开始失效 ——

→Prefill 是 算力受限型(compute-bound)
吃满 GPU FLOPs,长上下文一次性编码

→Decode 是 显存带宽受限型(memory-bound)
吃 VRAM 带宽,逐 Token 生成,延迟敏感

Agentic 推理负载,不是 Q&A 的放大版,而是一种全新的工作负载形态

AGENTIC INFERENCE IS A NEW WORKLOAD SHAPE, NOT A SCALED-UP Q&A.

PREFILL · COMPUTE-BOUND

Prefill算力密集

CHARACTERISTICS

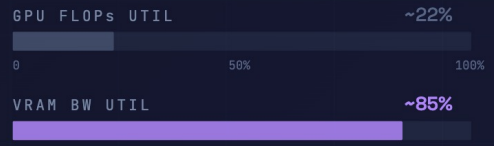
- 一次性大批量计算
- 长上下文一次性编码
- 显存带宽利用率低

吃满 FLOPs

算力是瓶颈,延迟由 token 总数决定

DECODE · MEMORY-BOUND / LATENCY-SENSITIVE

Decode是访存密集/低延迟



CHARACTERISTICS

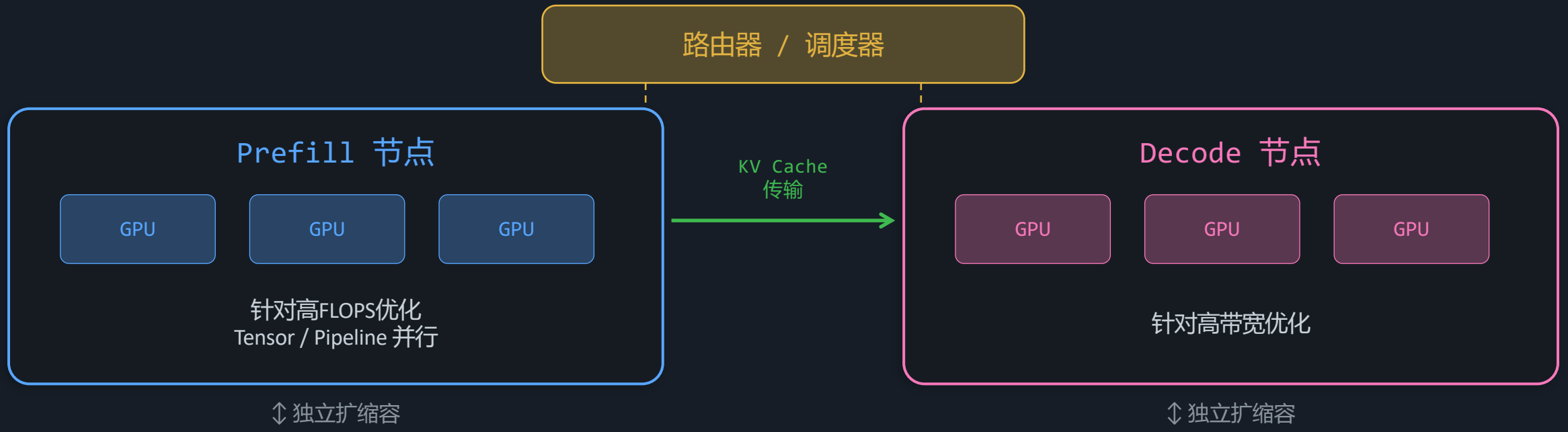
- 逐 Token 串行生成
- 每个 Token 都对延迟敏感
- FLOPs 利用率反而低

吃满 带宽

带宽是瓶颈,延迟由生成步数决定

Prefill-Decode 分离

为不同负载匹配合适的硬件

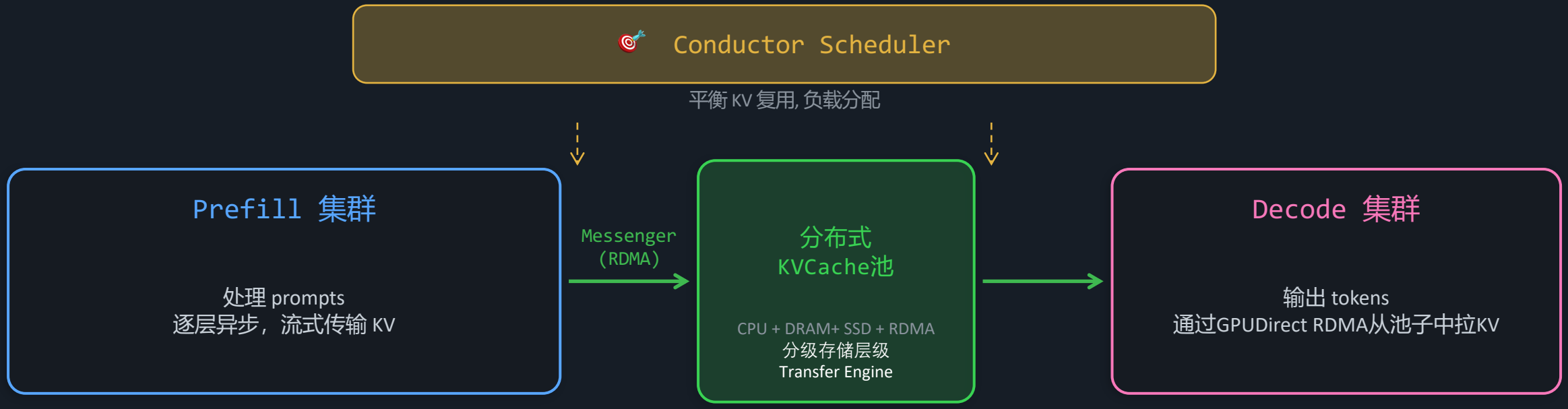


- KV Cache传输是TTFT的关键：KV从Prefill传到Decode的延迟直接计入首Token延迟（TTFT），网络慢则PD分离得不偿失
- 传输量级：单次请求的KV Cache可达数百MB~数GB（取决于模型大小和上下文长度），例如128K上下文的70B模型，单请求KV约2-4GB

Mooncake：以KVCache为中心的分离式推理

生产就绪的 PD 分离 + 分布式 KV 存储

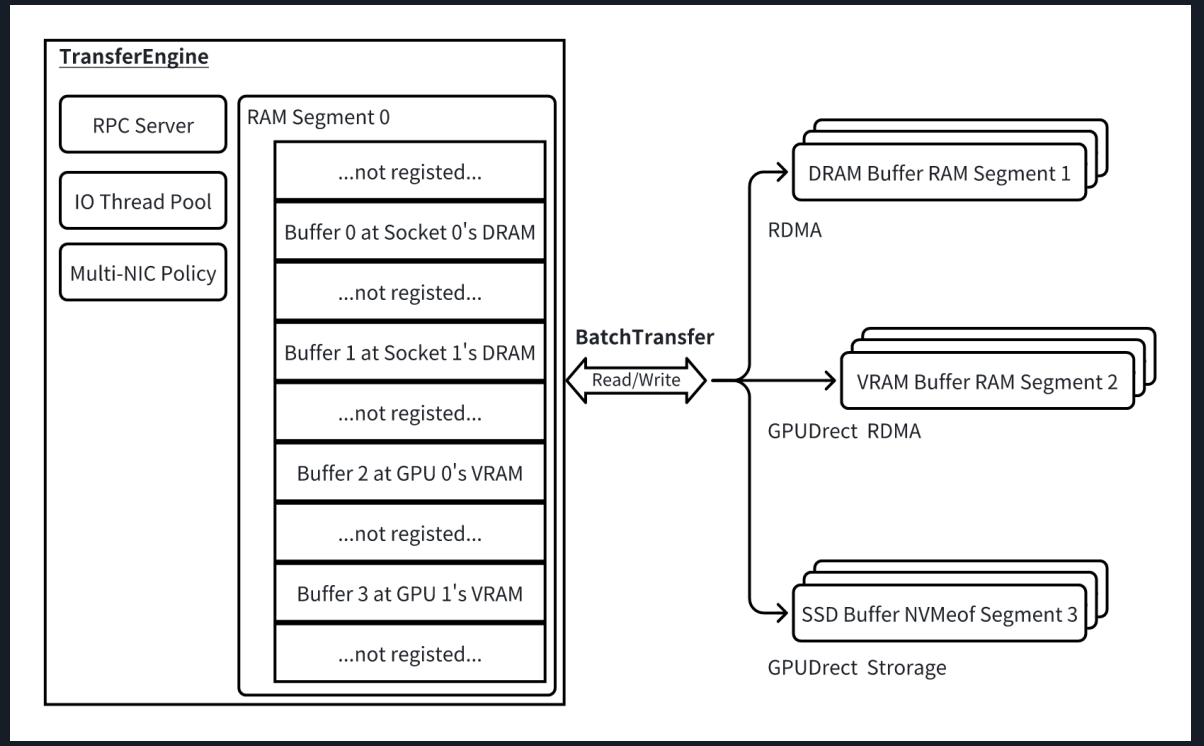
- **USENIX FAST 2025 最佳论文**
- **Transfer Engine**：高性能数据传输框架 —— 为跨存储设备、跨网络链路的批量传输提供统一接口，I/O 延迟远低于 Gloo / TCP



KV cache 不是推理的副产物 — 把 KVCache 作为一等公民，围绕它组织整个系统

Transfer Engine：Mooncake 的数据高速公路

- 统一的零拷贝数据传输抽象，向上层屏蔽底层网络
- 支持多种后端：TCP / RDMA (RoCE/IB) / NVMe-oF / … /
  - **现在 + EFA**
- 关键能力：
  - 多 NIC **聚合带宽**（topology-aware，按 GPU/NUMA 亲和选网卡）
  - 批量传输、零拷贝、GPUDirect RDMA（显存直达网卡，绕过 DRAM）
- 上层调用者无需关心底层是 RoCE 还是 EFA —— **只换一个 protocol 参数**

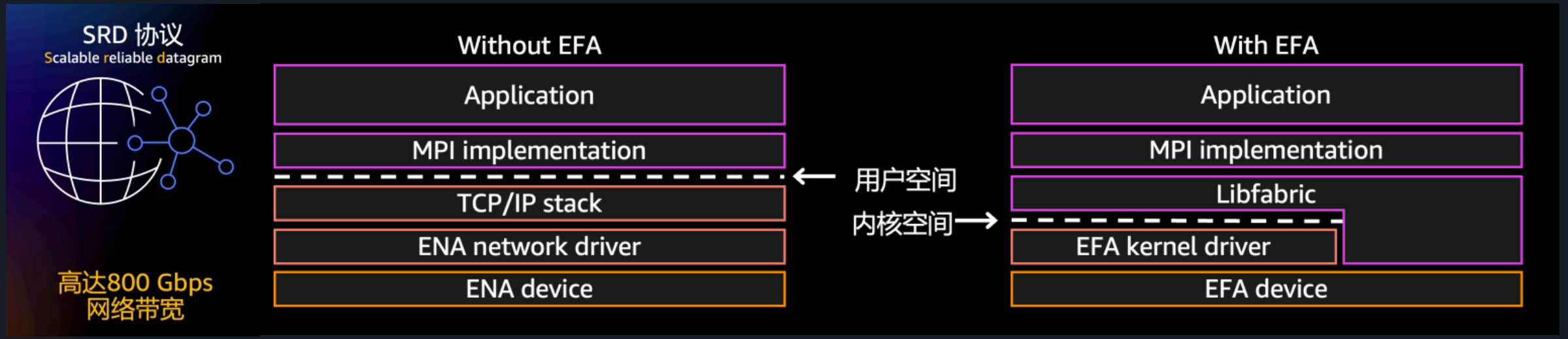


亚马逊云科技
中国峰会

EFA — 跨节点高性能网络

Elastic Fabric Adapter

基于 Nitro 的 EC2 实例中的高性能网络适配器，唯一的 RDMA 级 fabric



- OS Bypass 机制和创新的 SRD协议提供 RDMA 能力，为云 上AI/ML 和 HPC 负载提供低延迟、高吞吐的网络传输，接近本地集群的网络性能

EFA SRD VS RoCE/IB

特性	EFA (libfabric SRD)	RoCE / IB (libverbs)
协议类型	Scalable Reliable Datagram (SRD)	RDMA over Converged Ethernet (RoCE)
端点配置	`FI_EP_RDM`	RC Queue Pair (RC QP)
拓扑连接	**无连接 (多端动态映射，极富弹性)**	有连接 (端对端静态绑定)
节点扩展性	O(1) 极低开销 (无需新建物理 QP 资源)	需要建立多对多 QP 与状态机
单边 RDMA	硬件级单边 RDMA 卸载 (device RDMA)	硬件级单边 RDMA 卸载
编程接口	现代开放 libfabric 接口	传统 ibverbs 接口

- Mooncake 的 RDMA 后端、NIXL、NCCL 的 verbs 路径，做单边 RDMA 都依赖 **RC（Reliable Connected）QP**
- EFA 通过 ibverbs 暴露设备，**有 QP，但只有 UD / SRD 类型,所以要支持Transfer Engine就 要在 libfabric 上重建整个传输路径**

亚马逊云科技
中国峰会

Transfer Engine 支持 EFA的核心设计

从 EFA 的本质出发：无连接 SRD + 异步完成

EFA ≠ IB/RoCE，不能直接套用原 RDMA 后端。以下特性决定了 Transfer Engine必须重构传输路径：

- 无连接的 SRD：没有 QP-to-QP 的 RC 连接，对端通过 AV（Address Vector）槽位寻址 → 连接管理模型完全变了。
- 异步完成 + 弱原语：没有 Write with Immediate，completion 必须靠 CQ 轮询；并发提交时 completion 会丢。

环节	RDMA (ibverbs RC)	EFA (libfabric SRD)
连接对象	每对端一个 RdmaEndPoint，**含 N 个 RC QP（默认 2）**	**每 NIC 一个共享端点**，无连接
加一个对端	建 QP + 握手 + **4× ibv_modify_qp** 状态机	**O(1) fi_av_insert**（一个 AV 槽位）
扩展压力	QP = 对端 × NIC × 2 → **需端点驱逐**（FIFO/SIEVE）	共享端点，**无 QP 爆炸、无驱逐**

RDMA 把复杂度放在建连接（QP 状态机 + 驱逐）；EFA 无连接把这块消掉了，代价是 completion / 排序 / 背压要自己管 —— 这正是决策 ①（共享端点）和 ②③（自己锁、自己背压）的由来。

Transfer Engine 支持 EFA 的 核心设计

把"无连接 SRD + 异步完成"翻译成对上层透明的、生产可用的高性能传输层

SRD 共享端点 （Shared Endpoint）

- 每张网卡只建 1 个 ep，对端通过 AV 槽位索引，不再 per-connection 建 ep。

- 收益：冷启动 4×（99ms → 26ms）、warmup 15×（17s → 1.1s） 提速，连接元数据内存占用同步下降。

每网卡提交序列化 （Per-NIC Submit Serialization）

- 实测 FI_THREAD_SAFE 不够：多线程并发 fi_write 会丢 completion。

- 方案：每张网卡一把提交锁，提交路径串行化；CQ 轮询仍可并行。

- 是正确性而非性能优化——丢 completion 会直接挂死上层调度。

GPUDirect 拼图：FI_MR_HMEM + PTE-aware 大 MR 拆分

FI_MR_HMEM 打通 GPU 显存注册（否则直接 EFAULT）；

针对单 MR 上限（p5en 192GB）和 PTE上限（~24M），按页大小自动拆段并在所有 NIC 全覆盖。

→ 500GB KV 池：34 → 108 GB/s（~3×），任意池大小都能满带宽。

请求级 Round-Robin + WR 配额控制

- 流量按请求粒度 round-robin 到多张 EFA NIC，避免单卡热点。

- 全局约束：threads × batch ≤ NICs × 256，把上层并发收敛到硬件 WR 上限内，杜绝 ENOMEM / 队列溢出。

在客户实际负载中持续优化

亚马逊云科技
中国峰会

与 vLLM / SGLang 的整合

vLLM：只改 KV connector 的一个参数

- **Prefill实例:**

```
vllm serve <model> -tp 8 --port 8010 \
  --kv-transfer-config '{"kv_connector":"MooncakeConnector",
    "kv_role":"kv_producer",
    "kv_connector_extra_config":{"mooncake_protocol":"efa"}}’
```

- **Decode 实例：** 同上， kv_role 改成 kv_consumer ， 端口换成 8020
- 用 vllm-router 把 prefill / decode 串起来，policy round_robin
- 关键坑：PREFILL_HOST / DECODE_HOST 必须是 可达私网 IP，不能是 127.0.0.1
  - router 会把这个地址转发给对端做 KV 握手，localhost 会 Connection refused

切到 EFA：把mooncake_protocol 从 rdma 改成 efa，仅此而已。

SGLang：MOONCAKE_PROTOCOL=efa，已合入上游

- SGLang 的 PD-disagg Mooncake 集成 **从 MOONCAKE_PROTOCOL 读传输协议**
- SGLang PR #25083 已合入 SGLang main —— 新版本直接可用，无需打补丁
- 只需一个环境变量：

```
export MOONCAKE_PROTOCOL=efa
sglang serve <model> --tp 8 --dp 2 \
  --disaggregation-mode prefill \      # decode 节点改成 decode
  --disaggregation-transfer-backend mooncake \
  --disaggregation-bootstrap-port 8998
```

- 旧版本（PR #25083 之前）才需要 patch_sglang_efa.sh（幂等，可重复跑）
- 默认无需调那些 MC_* 旋钮 —— #1944 重构后在常见 PD 负载下都冗余了

那 NIXL 呢？—— KV 传输不止一个选择

- **NIXL（NVIDIA Inference Xfer Library）**：NVIDIA 的 KV 传输库，Dynamo 的原生后端，亚马逊云科技官方已支持

- **NIXL 是 传输层**；**Mooncake 是 完整的服务架构**（PD 分离编排 + 三层 KV 存储 + Transfer Engine）—— 二者层级不完全对等

- 在 EFA 上的现状（我的实测）：
  - NIXL 默认的 verbs / UCX 路径 同样卡在 ibverbs QP —— 在 EFA 上 起不来
  - NIXL 的 libfabric backend 可以跑：去掉 --enable-sleep-mode 后，Median TTFT 946ms，与 Mooncake 同档
- **两条路在 EFA 上都能走通**，选择更多是 **生态与架构** 层面的，而非"能不能用"

Mooncake vs NIXL：怎么选

- **为什么中国客户偏好 Mooncake**：不只是传输快，而是 PD 编排 + KVCache 分级复用 一整套，且与 vLLM / SGLang 上游深度打通
- 我的工作让 **这一整套** 能直接跑在 EFA 上 —— 客户不必为了上亚马逊云科技而换掉熟悉的架构

维度	Mooncake	NIXL
定位	完整服务架构（PD 编排 + 3 层 KV 存储 + 传输）	传输库（KV/张量搬运）
来源	Moonshot AI 开源，FAST 2025 最佳论文	NVIDIA，Dynamo 原生后端
EFA 默认路径	libfabric（本工作）原生支持	verbs/UCX 起不来；需切 libfabric backend
社区	中国客户主流选择，vLLM/SGLang 深度集成	随 Dynamo / NVIDIA 生态推广
KV 复用 / 分级缓存	内建（VRAM→DRAM→SSD 三层）	不在其职责范围

不止推理：训练场景也在用 Mooncake

- **RL 后训练（post-training）** 里有一个关键路径操作：**rollout 权重同步**
  - 每轮训练后，要把 trainer 的新权重同步给 inference engine 才能继续 rollout
  - 同步期间 **trainer 和 inference 都空转** —— 直接拖慢整体 RL 吞吐
- 传统做法用 **NCCL broadcast**：单源广播，模型越大、跨机越多，带宽越成为瓶颈
- SGLang 的新方案：**用 Mooncake Transfer Engine 做 RDMA 点对点（P2P）零拷贝** 权重传输
  - 启动时注册一次内存，避免每次序列化 CUDA IPC handle；去 GIL，多线程并行

来源：https://lmsys.org/blog/2026-04-29-p2p-update · miles + sglang-miles + Megatron/FSDP + Mooncake

训练侧：1T 模型权重同步快 7×

- **SGLang P2P（Mooncake RDMA）vs NCCL broadcast** · 以下为 lmsys 公布数据，测于 InfiniBand

模型	规模	节点	NCCL	RDMA·IB (Mooncake)	加速
Kimi-K2-fp8	1T (64B active)	32	53.3s	7.2s	7.37×
GLM-5	744B	16	58.3s	8.5s	6.88×
Qwen3-235B	235B	8	10.8s	3.2s	3.40×
GLM-4.5-Air	106B	4	5.0s	2.6s	1.90×

- 收益对 **大 MoE + 高专家并行 + 多机** 最显著（小模型 / 少节点可能反而更慢）
- **我在 2 台 EFA p5.48xlarge 上复现了这条链路**（miles + sglang-miles + Megatron + Mooncake）—— 链路跑通，逐项 EFA 性能数字补测中

同一套 Mooncake Transfer Engine —— 推理走 KV、训练走权重同步；
IB 上有公布数据，EFA 链路我已复现。

亚马逊云科技
中国峰会

PD 分离实测

测试一：万亿参数模型Kimi K2.6 · EFA KV 传输实测

- **模型**：Kimi K2.6（1.1T MoE，MLA 注意力）—— Mooncake 的"主场"模型
- **测法**：用 K2.6 真实 KV cache 尺寸（MLA：61 层 × 576 维/token）跑 EFA GPU-to-GPU 点对点传输
- **关键洞察**：万亿模型 + MLA → 单请求 KV 意外地小

上下文	单请求 KV (fp8)	EFA 传输带宽	单请求延迟（理论）
4K	~134 MiB	776 GB/s	~0.17 ms
32K	~1.05 GiB	776 GB/s	~1.4 ms
128K	~4.2 GiB	776 GB/s	~5.4 ms

"万亿参数" ≠ "KV 巨大" —— MLA 让 K2.6 的 128K 上下文单请求 KV 也才 ~4 GiB，EFA 上传输只需几十 ms。

测试一： c=1 零排队实测，把 KV 传输成本剥离出来

- 2×p5en.48xlarge，K2.6 INT4，c=1（零队列）： TTFT 只剩 prefill + 握手 + KV 传输；把协议从 EFA 降级到 TCP 即可单独量化传输成本；对比NIXL-EFA
  - **单机 ≈ Mooncake-EFA ≈ NIXL-EFA**：三者 TTFT 全程几乎重合 → PD 的 EFA KV 传输开销确认可忽略（256K/8.8GB 也只多~100ms，占 27 秒 0.4%）
  - **EFA 比 TCP 快 ~116~211ms**（RDMA 绕内核/零拷贝 vs 内核栈）——这就是 EFA 的价值；大输入差值反而缩小（256K 仅+50ms）→ **Mooncake 分层流水，传输藏在 prefill 后面**

输入	单机 TP8 （ms）	Mooncake-EFA （ms）	Mooncake-TCP （ms）	NIXL-EFA （ms）	KV 量 (fp8)
1K	256	156	155	155	34 MB
8K	595	525	683	529	274 MB
32K	2055	2118	2274	2097	1.1 GB
64K	4512	4559	4770	4578	2.2 GB
256K	28589	28684	28734	28690	8.8 GB

测试二：MiMo-V2-Flash · PD 分离 vs 单机

- **模型**：MiMo-V2-Flash FP8 · **1P + 1D**，2× p5en.48xlarge
- **测法**：decode 端全开 EAGLE MTP3 推测解码，KV cache 走 Mooncake over EFA

对比维度	单机（prefill+decode 同卡）	PD 分离（EFA 传 KV）
TPOT @ 16K/1K, rate=4	39.73 ms	5.77 ms（快 ~7×）
TTFT 端到端增量（PD vs 单机）	基线	+0.4~0.7 s
decode 饱和吞吐（4K/1K, rate=inf）	受 prefill 抢占	~12.4k tok/s / 节点

- 这 +0.4~0.7s 里，**真正的 KV 网络传输只有几十 ms**（数百 MB ÷ 数百 GB/s）
- 其余是 **SGLang PD 的调度 / 握手 / 路由开销** —— 固定协议成本，与 KV 大小无关

PD 分离把 decode 从 prefill 抢占中解放 → TPOT 快一个数量级；EFA 传 KV 本身只要几十 ms —— 网络完全不是瓶颈，收益远大于成本。

测试三：GLM-5.1-FP8 · EFA vs 本地 GPU +IB

- **模型**：GLM-5.1-FP8（754B MoE，MLA/DSA，FP8）
- **测法**：**2P + 2D**，4× p5en.48xlarge（每台 16× EFA）
  - KV cache 走 **Mooncake over EFA**
  - MoE all-to-all 走 **UCCL-EP**（DeepEP 的 EFA 实现）
- **对比基线**：客户本地 GPU + InfiniBand 集群（DeepEP）的真实 benchmark 值
- 客户压测口径完全对齐：3K / 60K / 120K 输入，输出 1K，开环到达 + 固定 seed

同样的 GPU，换成亚马逊云科技 + EFA，到底差多少？—— 答案是：不差，甚至更好。

TTFT 全面优于客户本地，稳定性极强

- **3 轮连续稳定性测试均值（同款 GPU 配置，DP=16 + 投机解码 MTP）：**

场景 (ISL/OSL)	指标	客户本地 IB	亚马逊云科技 EFA	对比
3K / 1K	Mean TTFT	4030 ms	2571 ms	好 36%
60K / 1K	Mean TTFT	8417 ms	5772 ms	好 31%
120K / 1K	Mean TTFT	11903 ms	11547 ms	持平略优
60K / 1K	Mean TPOT	14.7 ms	18.4 ms	高 ~26%（见注）

- 3 轮 2652/2652 请求 100% 成功，核心指标 CV < 3%；全程 0 transfer-failed、0 EFA CQ error
- TPOT 略高源自 EFA/UCCL-EP 尾部分布更宽 + MTP accept rate 0.65 vs 客户 ~0.85，非软件栈 bug

Key Takeaways

1. **Agentic AI → 持续高吞吐 → PD 分离**，但 PD 分离把代价转移到了**网络**
2. **亚马逊云科技上唯一的 RDMA 级 fabric 是 EFA**
3. **多个关键设计决策**，核心几条源自 EFA 的本质（无连接 SRD + 异步完成）：
   - ① **SRD 共享端点**：每网卡一个 ep，对端是 AV 槽位 → 冷启动 4×、warmup 15× 提速
   - ② **每网卡提交序列化**：FI_THREAD_SAFE 不够，并发 write 会丢 completion
   - ③ **按请求 round-robin 分发 + WR 配额**：threads × batch ≤ NICs × 256
   - ④ GPUDirect 拼图：FI_MR_HMEM + **PTE-aware 大 MR 拆分**（500GB 池 3×）
   - ⑤ 生产硬化：DP>1 全键、双 libfabric、loopback 短路…
4. **性能**：p6最新高性能计算实例能达到780 GB/s GPU-to-GPU，**~97.5% 线速**
5. **生态**：vLLM / SGLang 只换一个 protocol 参数，改动已进上游
6. **不止推理**：同一个引擎也用于 RL 训练的权重同步（1T 模型同步快 ~7×），EFA 上均已跑通

下一步 & 愿景

- **进行中的 PR**：重连时驱逐 stale-peer + 更详细的诊断日志（#2063）
- **更好的训练支持**：RL 权重同步、checkpoint 分发等训练侧路径在 EFA 上的稳定性与吞吐打磨，补齐逐项 EFA 性能数据
- **持续优化**：更多客户场景下的长跑稳定性、滚动升级期的握手健壮性
- **更广的硬件覆盖**：新代 EFA 实例上线即支持
- **与分层缓存正交互补**：跨机 KV 走 EFA，本机溢出的 KV 还可经 SGLang HiCache 下沉到 CPU RAM / NVMe —— 两条路彼此独立、可叠加
- **与社区共建**：把亚马逊云科技上的最佳实践沉淀进 Mooncake / vLLM / SGLang 上游文档
- **愿景**：让全球开发者 **直接在亚马逊云科技 AI 基础设施上部署 Mooncake**，构建面向 Agentic AI 时代 的生产级服务系统

亚马逊云科技
中国峰会

谢谢大家！

亚马逊云科技
中国峰会

Session 1 :Mooncake on EFA：万亿参数模型背后的开源服务架构实践

扫描上方二维码
填写调查问卷

亚马逊云科技
中国峰会

Session 1 :Mooncake on EFA：万亿参数模型背后的开源服务架构实践

扫描上方二维码
填写调查问卷

亚马逊云科技
中国峰会

Session 1 :Mooncake on EFA：万亿参数模型背后的开源服务架构实践

扫描上方二维码
填写调查问卷