刘亦飞

liuyifei5@xiaohongshu.com | scholar.google/YifeiLiu | github.com/NephrenCake

教育经历

上海交通大学 | 工科硕士 | 计算机系，计算机学院 *@EPCC* 2023.09—2026.03

南京工业大学 | 工科学士 | 计算机系，计算机学院 2019.09—2023.06

实习经历

华为云 | 大模型推理优化实习生 *@XDS Team* 2024.09—2025.12（预计）

xDeepServe 是支持 NPU 的大模型推理框架。本人在该项目中主要做出以下贡献:

- **负责 Batch API 的设计、开发、测试和部署**。完成 Batch API 相关业务逻辑，设计并实现异步高性能可恢复的 Batch 作业执行单元。
- **探索 Batch API 背后支撑的高吞吐推理优化**。参与 PP 并行优化的设计、讨论、实现与测试。主导设计两项基于计算分区的高吞吐推理优化。
- **负责早期 MoE 专家负载均衡系统的设计与开发**。采集分析专家热点信息，设计并实现专家放置调度策略、专家加载和路由机制。
- 参与 Attn-MoE 分离系统的设计、讨论与开发。
- 参与异步调度器的设计与讨论。

科研经历

目前有 1 篇一作 CCF-A、3 篇一作/共一在投（两篇已发布至 ArXiv）、2 篇合作在投、2 篇负责的工作正在进行中。

基于 GreenCTX 的大模型在离线推理混部 | 大模型推理优化 2025.08—至今

- 发现现有推理引擎服务在离线请求时存在严重资源浪费，无法同时满足在线请求的低延迟和离线请求的高吞吐要求。
- 利用低级 API（Nvidia Green Context）对 GPU 空分复用为两个独立计算分区，实现了计算资源动态分配、显存资源弹性伸缩、计算任务统一调度的在离线混部系统。

基于 GreenCTX 的大模型 Attn-FFN 算子并行推理 | 大模型推理优化 2025.05—至今

- 发现现有大模型串行推理的本质弱点，对 GPU 的时分复用导致 Attn 算子下 SM 空闲、FFN 算子下带宽空闲。
- 利用 GreenCTX 设计并实现了动态计算分区，为 AF 算子提供并行且隔离的异构计算性能。
- 实验结果表明我们的方法对比 SGLang，相同 TPOT 下吞吐提升 1.5-2x。

通过概率需求建模高效服务 LLM 应用 | 大模型推理优化 2024.09—2025.04

“Hermes: Efficient Serving of LLM Applications with Probabilistic Demand Modeling”（一作/[CCF-A]TACO）

- 发现现有推理引擎服务 LLM agent 时效率低下，忽略应用内的请求相关性，难以支持多样的调度目标和资源配置。
- 针对现有系统对 LLM agent 不感知的问题提出概率需求建模，从请求调度、资源供给两方面优化端到端系统性能。
- 实验结果表明 Hermes 对比现有系统，平均完成时间降低 70%，P95 完成时间降低 80%，DDL 满足率提升一倍。

通过弹性公平队列对深度学习训练任务进行高效公平调度 | 训练集群调度优化 2024.02—2024.08

“Ares: Fair and Efficient Scheduling of Deep Learning Jobs with Elastic Fair Queuing”（一作/[CCF-A]TACO）

- 发现现有的集群系统对 Scaling Curve 不感知，导致调度策略在效率和公平上均表现不佳。
- 研究内生弹性和扩张曲线，提出 Elastic Fair Queuing，在保证训练任务性能可预测性的前提下提供最大加速比。
- 实验结果表明 Ares 对比现有系统，平均完成时间降低 20%，不公平比例降低 40%。

用于异构 AI 工作负载的统一高效集群缓存 | 分布式缓存优化 2023.06—2024.09

“Athena: A Unified, High-Efficacy Cache for Heterogeneous AI Workloads”（共一/在投/ArXiv）

- 发现现有系统对上层混杂数据流访问模式不感知，导致系统性能不佳。
- 提出分层访问抽象，隔离不同应用的访问流，从缓存空间管理和缓存内容管理两方面优化系统性能。
- 实验结果表明 Athena 对比现有系统，缓存命中率提高 55%，平均完成时间降低 52%。

专业技能

- **语言**: 不受特定语言限制。熟练使用 Python、Pytorch。了解 CUDA、Triton、C++、Rust、Golang、Java。
- **框架**: 掌握 vLLM、SGLang 框架，并能在其上进行二次开发。
- **工具**: Linux、Git、Docker、Vim、Shell。
- **经验**: 对大模型推理优化有项目经验，了解大模型推理常用优化方法（如 Continuos Batching、PagedAttention、Chunked-Prefill、PD-Disaggregation、分层 KV Cache 缓存、TP/PP/EP/DP 并行策略、调度策略等）。