赵志宇

教育背景

南京大学 2023.09 - 2026.06

大气科学 硕士

南京大学 2019.09 - 2023.06

大气科学 理学学士

GPA: 4.66/5.00 (排名 2/65)

专业技能

- 熟悉 CUDA 编程，了解 LLM 常用算子的优化思路，了解 GPU 硬件的基本结构
- 熟悉 C++ 编程，熟悉 C++11 特性，熟悉 CMake 和 Bazel 等构建工具
- 熟悉 Python 编程，熟练使用 Pytorch 深度学习框架，熟悉 vLLM 推理框架
- 熟悉常见的 LLM 推理加速算法，如 PagedAttention，FlashAttention，KV Cache 等
- 英语流利，可无障碍阅读英文文献和技术资料，CET-6 602

实习经历

商汤科技 | 高性能计算与推理部 | 系统研究实习生 2024 年 5 月 - 2024 年 11 月

工作介绍 参与 PPL serving 和 OPMX 的开发，负责 LLM 推理性能测试系统的开发和性能测试工作。

- PPL 服务调度策略模块优化，QPS 提升 10% (C++)
- PPL 权重转换工具 OPMX 开发 (Python)
- 设计并开发推理性能测试系统 llm.perf，模拟高并发、长文本负载情况下不同推理引擎的性能差异
- 基于 llm.perf 进行大规模性能测试，覆盖 LLaMA 2/3、Qwen2、Deepseek V2 等模型，L40S、A100、A800、H800 等显卡，以及 vLLM、SGLang、LightLLM、PPL 等推理引擎

NVIDIA | 自动驾驶平台团队 | 软件开发实习生 2024 年 12 月 - 2025 年 5 月

工作介绍 NVIDIA Driveworks 软件平台开发。负责搭建代码静态测试与日志白名单基础设施，负责图像处理模块的 Coverity 和 Test coverage。

- 基于 Python 和 Bazel 搭建代码静态测试系统，控制代码质量，接入 CI / CD 系统
- 基于 Python 开发日志白名单系统，实现白名单自动测试、同步和追踪，并对接 Bug 追踪工具
- 编写 CUDA 测试用例，提高 imageprocessing 模块的测试用例覆盖率（Test coverage）
- 修复代码仓库中的 Coverity violations，保证 C++ 代码的规范和安全性

阿里云 | PAI | 系统研究实习生 2025 年 5 月 - 至今

项目经历

推理框架性能测试系统 - llm.perf

项目介绍 设计并实现了一套用于测试真实场景下 LLM 推理框架性能的基准测试系统。支持对多种 LLM 推理框架进行静态和动态性能测试，包括吞吐量和延迟等关键指标的评估。

- 支持多种推理后端（vLLM、LightLLM、SGLang、PPL 等）的性能评估和对比
- 静态推理测试评估引擎原生性能，动态测试评估线上服务整体性能
- 支持固定并发用户数和固定请求速率两种测试场景，通过多线程/多进程模拟大规模并发请求
- 支持全面的性能指标评估体系，包括 QPS（每秒请求数）、TTFT（首字延迟）、TPOT（每 Token 生成延迟）、E2E（端到端延迟）、ITL（Token 间延迟）、TPS（吞吐量）等
- 通过配置文件自定义 batch size、模型、TP、PP、EP、输入输出长度、并发数、缓起时间等参数组合
- 设计统一的测试数据集和评估方法，确保不同后端间的性能对比公平有效

基于 YOLOv8 的工业仪表视觉读表系统

项目介绍 设计并实现了一套能够从 RTSP 视频流中自动提取液位计和指针式气压表读数的工业仪表读表系统。通过 TensorRT 加速和 CUDA 优化，在 NVIDIA Tesla P4 GPU 上实现了高效实时处理。

项目地址 https://github.com/lantel-wm/meter_infer

- 以数千张真实的工业仪表图片作为数据集，训练 YOLOv8n (3.2M) 和 YOLOv8s-seg (11.8M) 模型
- 开发双模型流水线，输入图像依次经过预处理、目标检测、语义分割以及后处理，提取仪表信息
- 设计了图像处理算法，从检测和分割结果获取表盘刻度和指针位置，进而得到仪表读数
- 使用 C++ 实现了支持多路并发的生产者-消费者多线程处理框架，支持多路 RTSP 视频流输入
- 使用 CUDA 加速预处理，TensorRT 优化模型推理速度，整体 8 路视频流并发处理时帧率达到 20 FPS

轻量级 LLM 推理框架

项目介绍 设计并实现了一个轻量级的大语言模型推理框架，支持 CPU 和 CUDA GPU 两种推理后端。通过 CUDA 核函数优化和细粒度的内存管理，提供高效的模型推理体验。
项目地址 https://github.com/lantel-wm/llm_infer

- 实现了完整的 Transformer 架构组件，包括 MHA、RoPE、FFN 等
- 基于 CUDA 开发了高性能算子，包括 GeMM、Rotary Position Embedding、RMSNorm 等
- 实现了 KV-Cache 机制，加速自回归生成过程中的序列处理
- 使用 C++20 和 CUDA 实现核心框架，通过 Armadillo 线性代数库优化了 CPU 后端性能
- 采用模块化设计，提供了统一的 Tensor 抽象和层级操作接口，支持灵活扩展模型结构

CMU 10-714 Deep Learning Systems

项目介绍 使用 Python、C++ 和 CUDA，从零开始 (不借助 numpy) 开发一个类似 Pytorch 的简易深度学习框架

- 实现计算图的动态生成和前向计算，基于拓扑排序实现自动微分系统
- 编写与 Pytorch 类似的神经网络模块，例如 nn.Linear()，nn.LayerNorm1d()，nn.Dropout() 等
- 实现神经网络权重的随机初始化，编写 SGD 和 Adam 优化器
- 基于 C++ 和 CUDA 编写 NDArray 库作为 Tensor 的后端，使用 CUDA 实现了规约、矩阵乘法等操作
- 基于该简易深度学习框架实现了 CNN、Transformer 等常见的深度学习模型的训练