Paratera Demo (UCAS-GPU-LLM-Inference)

By SLDRMK

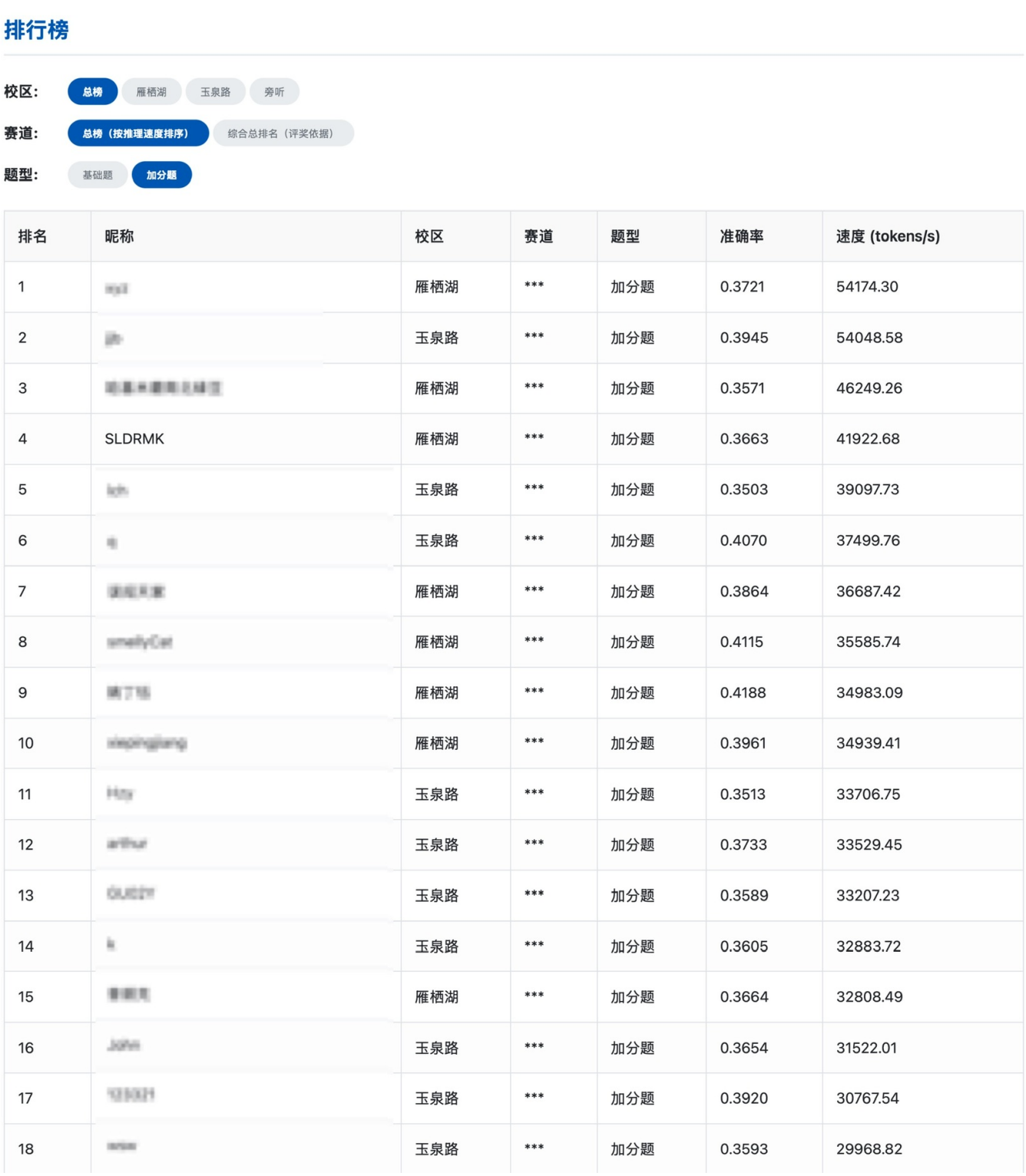
1. 项目简介 (Introduction)

本项目是中国科学院大学2025秋《GPU架构与编程》课程大作业二（LLM Inference Optimization）的推理服务部分。

项目地址：https://github.com/SLDRMK/GPU_LLM_Inference_Paratera

本方案针对 **Qwen/Qwen3-0.6B** 模型在 NVIDIA RTX 5090 (32GB) 环境下的高吞吐推理进行了深度优化。在开发过程中，经历了从 **Qwen3-4B AWQ** 量化模型到 **Qwen3-0.6B 浮点基座 + FP8 KV Cache** 的技术路线切换：

- **初期尝试**：使用 Qwen3-4B-AWQ 模型，利用 vLLM 的 AWQ Marlin 内核进行加速。虽然显存占用得到控制，但在极高并发下（batch size > 256）算力仍是瓶颈，tokens/s 提升受限。最终最好成绩约**16k tokens/s**。
- **最终方案**：切换至参数量更小但架构更优秀的 **Qwen3-0.6B** 基座模型，配合 **FP8 KV Cache** 与 **激进的算子融合编译（torch.compile + Inductor）**。这一转变使得在保持高精度的同时，推理吞吐量（tokens/s）获得了数倍提升，充分释放了 5090 的算力与显存带宽。最终最好成绩约**42k tokens/s**。
- 最终排名情况见下图



排行榜

校区： 总榜 雁栖湖 玉泉路 旁听

赛道： 总榜（按推理速度排序） 综合总排名（评奖依据）

题型： 基础题 加分题

排名	昵称	校区	赛道	题型	准确率	速度 (tokens/s)
1	[illegible]	雁栖湖	***	加分题	0.3721	54174.30
2	[illegible]	玉泉路	***	加分题	0.3945	54048.58
3	[illegible]	雁栖湖	***	加分题	0.3571	46249.26
4	SLDRMK	雁栖湖	***	加分题	0.3663	41922.68
5	[illegible]	玉泉路	***	加分题	0.3503	39097.73
6	[illegible]	玉泉路	***	加分题	0.4070	37499.76
7	[illegible]	雁栖湖	***	加分题	0.3864	36687.42
8	[illegible]	雁栖湖	***	加分题	0.4115	35585.74
9	[illegible]	雁栖湖	***	加分题	0.4188	34983.09
10	[illegible]	雁栖湖	***	加分题	0.3961	34939.41
11	[illegible]	玉泉路	***	加分题	0.3513	33706.75
12	[illegible]	玉泉路	***	加分题	0.3733	33529.45
13	[illegible]	玉泉路	***	加分题	0.3589	33207.23
14	[illegible]	玉泉路	***	加分题	0.3605	32883.72
15	[illegible]	雁栖湖	***	加分题	0.3664	32808.49
16	[illegible]	玉泉路	***	加分题	0.3654	31522.01
17	[illegible]	玉泉路	***	加分题	0.3920	30767.54
18	[illegible]	玉泉路	***	加分题	0.3593	29968.82

注意：本项目仅包含推理服务与评测逻辑；模型的微调训练部分请参考：SLDRMK/GPU_LLM。

---

2. 目录 (Table of Contents)

---

3. 项目特点 (Features)

本项目基于 vLLM 引擎进行了深度的参数调优与编译优化，核心服务代码位于serve.py中，主要特点包括：

1. **极致的编译优化 (Aggressive Compilation)**：
   - 启用 `torch.compile` (Inductor backend) 的最高优化等级 (Level 3)。
   - 定制 `cudagraph_capture_sizes`，专门针对评测集的 Batch Size (358) 录制全图 CUDA Graph，消除 Python launch overhead。
   - 开启 `combo_kernels`、`epilogue_fusion` 与 `max_autotune`，最大化算子融合收益。
2. **FP8 KV Cache 加速**：
   - 虽然模型权重保持 BF16/FP16 精度以确保生成质量，但 KV Cache 采用 `fp8_e4m3` 格式存储。
   - 在 RTX 5090 上显著减少了显存读写带宽压力，允许更大的 Batch Size 与更长的 Context。
3. **高性能 Attention 后端**：
   - 强制指定 `TRITON_ATTN` 与 `FLASHINFER` 后端。
   - 启用 `use_cudnn_prefill` 与 `use_prefill_decode_attention` 分离，针对 Prefill 与 Decode 阶段分别优化。
4. **异步调度与全量预热**：
   - 开启 `async_scheduling`，减少 CPU 调度带来的 GPU 空闲间隙。
   - 在服务启动阶段（Health Check 前）执行一次与评测规模完全一致（Batch=358）的预热，确保 CUDA Graph 与 Memory Pool 在评测开始前即处于最佳状态。
5. **精简的依赖管理**：
   - Docker 镜像基于 `vllm-openai:v0.13.0`，严格对齐生产环境版本。
   - `requirements.txt` 采用兼容性配置，避免破坏基础镜像构建好的 CUDA/PyTorch 依赖栈。
6. **仿真评测系统**
   - 使用`http`服务进行健康检查和问答服务
   - eval_official_http.py模拟评测系统批量推送提问，进行`RougeL score`计算和推理速度（`tokens/s`）等，方便进行调试。

---

4. 常用命令行指令 (Quick Start)

1) 构建镜像

建议使用 `--no-cache` 重新构建以确保代码更新生效，并清理可能存在的旧镜像层。

```
# 1. 停止并删除可能存在的旧容器
sudo docker ps -a | grep paratera-demo | awk '{print $1}' | xargs -r sudo
```

```
docker stop
sudo docker ps -a | grep paratera-demo | awk '{print $1}' | xargs -r sudo
docker rm

# 2. 删除旧镜像（可选，释放空间）
sudo docker rmi paratera-demo:latest

# 3. 重新构建镜像（推荐使用 --no-cache 确保环境最新）
# 在项目根目录下执行（无需梯子，模型下载脚本使用国内 ModelScope 源）
sudo docker build --network host --no-cache -t paratera-demo:latest .
```

> **说明**：构建过程中会自动执行 `download_model.py`，将 Qwen3-0.6B 模型下载到镜像内的 `/app/Qwen3-0.6B` 目录。

2) 启动容器

推荐运行命令（已调优参数）：

```
sudo docker run --rm --gpus all --ipc=host --ulimit memlock=-1 --ulimit
stack=67108864 \
  -p 8000:8000 \
  paratera-demo:latest
```

> **说明**：
>
> - `serve.py` 内部已内置了针对 0.6B 模型与 5090 显卡的最佳默认参数（Batch=384, FP8 KV, Compilation Level 3 等），直接运行即可。
> - 服务启动时会进行约 10-30 秒的模型加载与编译预热，请等待日志出现 `Uvicorn running on http://0.0.0.0:8000` 后再发起请求。

3) 本地评测

使用提供的 `eval_official_http.py` 脚本进行 ROUGE-L 正确性验证与 tokens/s 吞吐量测试。

前置准备： 需要将镜像内的模型拷出（或本地已有），以便评测脚本加载 Tokenizer 统计 token 数量。

```
# 假设本地已有 Qwen3-0.6B 模型在 ~/data/models/Qwen3-0.6B
# 或者是第一次运行，先从镜像里拷出来：
cid=$(docker create paratera-demo:latest)
docker cp "$cid":/app/Qwen3-0.6B ./Qwen3-0.6B
docker rm "$cid"
```

运行评测：

```
# 安装评测依赖
pip install -r requirements.txt
```

```
# 启动评测（假设服务已在 8000 端口运行）
python eval_official_http.py \
  --no_start_server \
  --server_url http://127.0.0.1:8000 \
  --model_path ./Qwen3-0.6B \
  --repeat 1 \
  --request_chunk_size 384
```

---

5. 致谢 (Acknowledgments)

感谢中国科学院大学《GPU架构与编程》赵地老师课程组提供的平台与指导。

感谢 vLLM 开源社区提供的卓越推理框架支持。

感谢 ModelScope 社区提供的模型托管服务。

感谢 Qwen 团队开源的优秀模型。

感谢VincentYang(哈基米磨南北绿豆), 名井南等同学的技术交流与支持。