PR #25221 完整报告

sgl-project/sglang

[MLX] bench_one_batch: thread --quantization through to MlxModelRunner

合并时间：2026-05-15 00:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25221

执行摘要

- 一句话：修复 MLX bench 量化参数遗漏传递
- 推荐动作：建议合入。这是一个清晰的缺失参数传递修复，影响范围小且已由 reviewer 批准。

功能与动机

PR body 明确指出：_MlxBenchRunner.__init__ 在构造 MlxModelRunner 时未转发 server_args.quantization，导致 --quantization mlx_q4 被静默忽略，模型以 fp16 加载，在 64GB Mac 上运行 32B 模型时 OOM。生产路径 MlxTpModelWorker 已正确转发该参数（#24907），但 bench 工具被遗漏。问题由 @jlee5814 报告。

实现拆解

1. 在 python/sglang/bench_one_batch.py 的 _MlxBenchRunner.__init__ 方法中，于 init_kwargs 字典内增加键 quantization，值为 server_args.quantization。
2. 该字典随后作为关键字参数解包传递给 MlxModelRunner(**init_kwargs)，从而将用户指定的量化选项传递给 MLX 模型加载器。
3. 仅 1 行新增代码，无其他文件修改，无测试变更（现有 MlxModelRunner 的量化单元测试已覆盖该参数路径）。

关键文件：

- python/sglang/bench_one_batch.py（模块 Bench 工具；类别 source；类型 core-logic；符号 _MlxBenchRunner.init）：修复点所在文件，添加了一行 quantization=server_args.quantization 到 init_kwargs 字典中。

关键符号：_MlxBenchRunner.init

关键源码片段

python/sglang/bench_one_batch.py

修复点所在文件，添加了一行 quantization=server_args.quantization 到 init_kwargs 字典中。

```
# _MlxBenchRunner.__init__ 方法的关键部分
init_kwargs = dict(
    model_path=server_args.model_path,
    trust_remote_code=server_args.trust_remote_code,
```

```
        disable_radix_cache=True,
        mem_fraction_static=server_args.mem_fraction_static,
        quantization=server_args.quantization, # 新增：传递量化参数，避免模型以 fp16 加载导致
        OOM
    )
    if server_args.max_total_tokens is not None:
        init_kwargs["pool_size"] = server_args.max_total_tokens
    self.mlx_runner = MlxModelRunner(**init_kwargs)
```

评论区精华

无讨论。Review 由 yeahdongcn 直接批准，无评论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅一行，且生产路径已通过类似方式传递量化参数；若 server_args.quantization 为 None，MLX runner 内部应有默认处理（如不量化），不会引入回归。
- 影响：仅影响 MLX bench 工具 bench_one_batch.py 的使用者：修复后 --quantization 选项在 bench 模式下生效，内存占用正确降低，避免 OOM。不影响生产推理或其他硬件后端。
- 风险标记：暂无

关联脉络

- PR #24907 MLX: add quantization support: 本 PR 是 #24907 的补全，后者在生产路径中正确传递了 quantization，但遗漏了 bench 工具。