PR #30181 完整报告

sgl-project/sglang

[MLX] Fix single-token chunked-prefill continuation misrouted as decode

合并时间：2026-07-07 11:31

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30181

执行摘要

- 一句话：修正 MLX 后端单 token 分块预填充延续误路由为 *decode*
- 推荐动作：建议阅读此 PR，以理解状态驱动路由（check decoding_reqs）相比长度启发式的优势，以及通过抽取共享方法同步双路径的设计模式。对于参与 MLX 后端的开发者，合并后可消除隐式数据损坏风险。

功能与动机

当 chunked prefill 最后一块只有一个 token 时，旧逻辑仅通过 seq_len>1 区分 continuation 和 mixed *decode*，使得 1 token continuation 被误判为 *decode*，*decode* 路径不消费输入 token，导致真实 prompt token 被丢弃，生成基于错误的 prompt 继续。此问题在模型预测与实际 token 不一致时暴露。详见 PR body 及 repro 说明。

实现拆解

1. 问题定位：在 MlxTpModelWorker 中，_forward_batch_generation_mlx 和 _async_extend_batch 都使用 seq_len > 1 作为判定条件，导致 1 token chunk 延续进入 *decode* 分支。
2. 抽取共享路由方法：新增 _route_extend_request(self, rid, decoding_rids)，根据 self._mlx_runner.has_request(rid) 和 rid in decoding_rids 返回 prefill、*decode* 或 continuation。
3. 同步路径修改：在 _forward_batch_generation_mlx 中构建 decoding_rids 集合，用路由方法替换原有的长度检查。
4. 异步路径修改：在 _async_extend_batch 中进行相同修改，确保两个路径行为一致。
5. 单元测试：新增 test_tp_worker_routing.py，使用 _FakeRunner mock 模型执行器，测试三种路由决策并覆盖同步 / 异步调用。

关键文件：

- test/registered/unit/hardware_backend/mlx/test_tp_worker_routing.py（模块 路由测试；类别 test；类型 test-coverage；符号 _FakeRunner, TestRouting, TestAsyncWiring, test_continuation_one_token_routes_to_extend）：新增的单元测试套件，使用 mock runner 验证路由决策，覆盖同步和异步路径，是保证修复正确性的关键。
- python/sglang/srt/hardware_backend/mlx/tp_worker.py（模块 MLX 路由；类别 source；类型 core-logic；符号 _route_extend_request, _forward_batch_generation_mlx,

_async_extend_batch）：核心修改文件，新增 _route_extend_request 方法，修改两个 forward 路径使用新路由，消除长度启发式。

关键符号：_route_extend_request, _forward_batch_generation_mlx, _async_extend_batch, _FakeRunner.extend, _FakeRunner.decode_batch

关键源码片段

python/sglang/srt/hardware_backend/mlx/tp_worker.py

核心修改文件，新增 _route_extend_request 方法，修改两个 forward 路径使用新路由，消除长度启发式。

```
def _route_extend_request(self, rid: str, decoding_rids: set[str]) -> str:
    # 如果 runner 还没有这个 request，则是全新的 prefill
    if not self._mlx_runner.has_request(rid):
        return "prefill"
    # 否则检查是否属于当前 batch 中的真实 decode 步骤（由 scheduler 标记）
    if rid in decoding_rids:
        return "decode"
    # 其他情况都是分块预填充的延续（无论长度是否为 1）
    return "continuation"
```

```
# 同步路径中的典型使用
decoding_rids = {r.rid for r in (batch.decoding_reqs or [])}
for i, req in enumerate(reqs):
    ...
    route = self._route_extend_request(req.rid, decoding_rids)
    if route == "continuation":
        next_token = self._mlx_runner.extend(req.rid, req_token_ids, req_new_slots)
        extend_rids.append((req.rid, next_token))
    elif route == "decode":
        decode_rids.append(req.rid)
    else: # "prefill"
        # 执行 prefill 相关逻辑
        ...
```

评论区精华

review 中 yeahdongcn 指出异步路径 _async_extend_batch 同样存在 bug 但未被测试固定，建议添加测试或提炼共享 helper。noob-se7en 采纳建议，将路由决策抽取为 _route_extend_request 方法，两个路径共同调用，并新增异步路径测试用例，确保两种路径都经过回归验证。

- 建议异步路径测试覆盖或提炼共享 helper (testing): 采纳建议：抽取 _route_extend_request 共享方法，同步和异步路径统一调用，并新增异步路径测试用例覆盖。

风险与影响

- 风险：风险较低：重构提炼共享方法后逻辑简单明确，单元测试覆盖了三种关键场景。但风险包括： 1) batch.decoding_reqs 必须由调度器正确维护，否则可能误判； 2) 测试依赖 mock runner，未覆盖完整调度器交互； 3) 仅影响 MLX 后端，不影响其他硬件。
- 影响：
  - 用户：Apple Silicon 用户在特定 prompt 长度下使用 chunked prefill 时，之前可能产生错误推理结果，现在修复。
  - 系统：无性能影响，路由开销极小。
  - 团队：代码重复消除，可维护性提高；新增测试为后续变更提供安全网。
  - 风险标记：MLX 后端特有，Mock 测试（非端到端），依赖 decoding_reqs 正确维护

关联脉络

- 暂无明显关联 PR