

PR #27446 完整报告

sgl-project/sglang

Fix PP is_fully_idle missing in-flight microbatches

合并时间: 2026-06-07 17:38

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/27446>

执行摘要

- 一句话: 修复 PP 下 is_fully_idle 忽略在途微批次
- 推荐动作: 该 PR 值得精读, 特别是调度器 idle-gating 逻辑的细节和 PP 下竞态条件的分析。修复方案简洁, 测试设计精妙, 适合作为类似竞争条件修复的参考。

功能与动机

PR body 指出: `is_fully_idle()` cannot see in-flight PP microbatches. 在流水线并行下, 调度器将在途微批次保存在 `self.mbs` 中; 当最后一块 prefill chunk 被调度后, `chunked_req` 已被清空, `running_batch` 等结构也为空, 但微批次仍然在流水线中飞行。此时若处理 `/flush_cache`, 它会通过 idle 检查并重置 KV 缓存, 导致后续处理批结果时触发 `assert node is self.root_node` 错误。这是 PP+ 分块预填充部署调用 `/flush_cache` 时可能复现的 bug。

实现拆解

1. Scheduler idle-gating 修复: 在 `python/sglang/srt/managers/scheduler.py` 中, 提取 `_pp_microbatches_drained()` 方法, 在原有 `running_mbs` 检查基础上, 额外检查 `self.mbs` 中所有微批次槽是否为空。is_fully_idle() 改用该方法, 确保 idle 判断考虑所有在途微批次。
2. 测试框架请求可见性修复: 在 `python/sglang/test/scripted_runtime/context/queries.py` 的 `_get_all_reqs()` 中, 当 PP 开启时, 遍历所有微批次槽 (`mbs`、`last_mbs`、`running_mbs`), 防止在途请求被遗漏导致 `finished` 状态提前为 `true`。
3. 测试框架重置逻辑修复: 在 `python/sglang/test/scripted_runtime/scheduler_hook.py` 的 `_reset_engine_state()` 中, 改用 `scheduler.is_fully_idle()` 判断引擎是否安静, 并在超时时报出 `RuntimeError`, 避免静默 `flush_cache`。
4. 新增回归测试: 在 `test/registered/chunked_prefill/test_scripted_core_4gpu.py` 中新增 `test_pp_flush_cache_during_inflight_chunk_results`, 该测试精确构造竞争窗口 (队列和当前槽清空但微批次在途), 执行 `flush_cache` 并断言请求能正常完成。

关键文件:

- `python/sglang/srt/managers/scheduler.py` (模块 调度器; 类别 source; 类型 core-logic ; 符号 `_pp_microbatches_drained`): 核心修复文件, 新增 `_pp_microbatches_drained` 方法并修改 `is_fully_idle` 调用。

- test/registered/chunked_prefill/test_scripted_core_4gpu.py (模块 测试用例; 类别 test; 类型 test-coverage; 符号 test_pp_flush_cache_during_inflight_chunk_results, _script_flush_during_inflight_chunk_results) : 新增回归测试, 精确验证修复效果。
- python/sglang/test/scripted_runtime/scheduler_hook.py (模块 测试钩子; 类别 test; 类型 test-coverage) : _reset_engine_state 改用 is_fully_idle 驱动等待, 并在超时抛出异常。
- python/sglang/test/scripted_runtime/context/queries.py (模块 查询工具; 类别 test; 类型 test-coverage) : _get_all_reqs 在 PP 模式下遍历所有微批次槽, 确保在途请求可见。

关键符号: _pp_microbatches_drained, _reset_engine_state, _get_all_reqs, test_pp_flush_cache_during_inflight_chunk_results, _script_flush_during_inflight_chunk_results

关键源码片段

python/sglang/srt/managers/scheduler.py

核心修复文件, 新增 `_pp_microbatches_drained` 方法并修改 `is_fully_idle` 调用。

```
def is_fully_idle(self, for_health_check=False) -> bool:
    idle = (
        self.running_batch.is_empty()
        and self.chunked_req is None
        and not self.dllm_manager.any_staging_reqs()
        and (self.last_batch is None or self.last_batch.is_empty())
        and (self.cur_batch is None or self.cur_batch.is_empty())
        and (not self.enable_overlap or len(self.result_queue) == 0)
        and self._pp_microbatches_drained()
    )
    # ... 其余检查 ...
    return idle

def _pp_microbatches_drained(self) -> bool:
    """检查所有PP微批次是否都已排空。"""
    if self.ps.pp_size == 1:
        return True # 单机无 PP
    return all(x.is_empty() for x in self.running_mbs) and all(
        mb is None or mb.is_empty() for mb in self.mbs
    )
```

评论区精华

gemini-code-assist[bot] 提出两条建议: 在 `_pp_microbatches_drained` 和 `_get_all_reqs` 中, 使用 `getattr` 和默认空列表来访问 `running_mbs`、`mbs`、`last_mbs` 等属性, 避免在调度器初始化不完全时引发 `AttributeError`。这些建议未在 PR 中采纳或解决, 但 PR 已合并。

- 防御性访问 `mbs/running_mbs` (design): 未采纳, PR 已合并。

风险与影响

- 风险：
 - 回归风险：修改了 `is_fully_idle()`，该函数用于多个场景（`flush_cache`、`HiCache attach/detach`、健康检查等）。但通过保留 `pp_size==1` 短路和新增测试，风险较低。
 - 性能影响：循环检查所有微批次槽（最多 `pp_async_batch_depth + 1` 个），仅在 `idle` 判定时执行，非热点路径，影响可忽略。
 - 防御性不足：`gemini-code-assist[bot]` 提出的 `AttributeError` 风险在现有初始化顺序下不会触发，但未来重构可能暴露。
- 影响：
 - 用户影响：修复了 PP+ 分块预填充场景下 `flush_cache` 导致 KV 缓存损坏的 bug，对使用 RL 权重更新等流程的用户至关重要。
 - 系统影响：`idle-gating` 行为更准确，使所有依赖 `idle` 状态的操作更加安全。
 - 团队影响：新增的回归测试为同类竞态条件提供了可复现的验证方法。
 - 风险标记：PP `idle-gating` 变更，竞争条件修复，KV 缓存损坏风险

关联脉络

- PR #27445 Complete server warmup before scripted runtime scripts start: 本 PR 基于 #27445 引入的测试框架，并修复了一个被其暴露的竞争条件。