PR #31835 完整报告

sgl-project/sglang

Negotiate PrefillDelayer only after KV-budget admission checks

合并时间：2026-07-22 03:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31835

执行摘要

- 一句话：修复 PrefillDelayer 协商时机，避免混合转发 bug
- 推荐动作：值得精读，特别是调度调度代码的审阅者。该 PR 展示了如何正确放置跨进程协商点，并提供了完整的测试覆盖，设计决策清晰，适合作为耦合组件的参考。

功能与动机

避免 DP attention 下高 KV 利用率时，因协商过早导致 delayer 失效，产生混合 prefill/decode 转发的性能与正确性问题。

实现拆解

1. 移动协商位置：在 PrefillAdder.add_one_req 中，将 negotiate_should_allow_prefill 调用从函数最前面移除，放到所有 KV 预算检查（包括 pre-lock 的 total_tokens/SWA/rem_input_tokens 检查，以及 post-lock 的 _lock_node 重检）之后、init_load_back 之前。
2. chunked 请求协商：在 add_chunked_req 中新增协商调用，固定报告 local_prefillable=True 并忽略判决结果，因为 mid-chunk rank 必须继续本轮 prefill 以免内存泄漏。
3. 补充测试：在 test_prefill_adder.py 中新增 5 个测试用例，覆盖 KV 预算拒绝、锁后重检失败、chunked 请求、延迟判决阻止、允许判决通过等场景，确保协商行为正确。

关键文件：

- python/sglang/srt/managers/schedule_policy.py（模块 调度器；类别 source；类型 core-logic；符号 add_one_req, add_chunked_req）：核心逻辑变更：移动协商位置，新增 chunked 请求协商，修正调度关键路径。
- test/registered/unit/managers/test_prefill_adder.py（模块 测试；类别 test；类型 test-coverage；符号 _RecordingDelayer, init, negotiate_should_allow_prefill, test_delayer_not_consulted_when_kv_budget_rejects）：新增 5 个测试用例和 _RecordingDelayer 模拟类，全面覆盖新协商行为。

关键符号：PrefillAdder.add_one_req, PrefillAdder.add_chunked_req, _RecordingDelayer.init, _RecordingDelayer.negotiate_should_allow_prefill

关键源码片段

python/sglang/srt/managers/schedule_policy.py

核心逻辑变更：移动协商位置，新增 chunked 请求协商，修正调度关键路径。

```
def add_one_req(self, req, has_chunked_req, truncation_align_size):
    # ... 前置检查 (dsa_prefill_cp, max_requests, ignore_eos) ...

    # 计算预算后的截断限制
    chunk_tokens_limit = min(self.rem_chunk_tokens, swa_cap)

    # 协商只在所有 KV 预算检查之后 （NO_TOKEN 的 rank 通过 finalize() 报告不可 prefill）
    if (self.prefill_delayer_single_pass is not None) and (
        not self.prefill_delayer_single_pass.negotiate_should_allow_prefill(
            local_prefillable=True,
            running_batch=self.running_batch.batch_size(),
            max_prefill_bs=self.max_prefill_bs,
            max_running_requests=self.max_running_requests,
            waiting_queue_len=self.waiting_queue_len,
        )
    ):
        return AddReqResult.OTHER

    if req.needs_host_load_back():
        new_indices, req.last_node = self.tree_cache.init_load_back(...)
    # ... 后续预填充逻辑 ...
```

评论区精华

PR 正文中提到了以下关键讨论点：

- 安全性：collective 契约不变，仍每 pass 一次 all-gather，且 _lock_node 内的协商是安全的，因为锁是单线程调度器上的驱逐保护引用计数。
- 已知残余：chunked prefill 的截断 / 对齐 OTHER 分支在协商后仍可能失败，但覆盖这些需要更晚的协商点，会延迟判决，是固有折衷。
  - 协商放置的安全性与 collect 契约不变 (design): 确认安全，无需额外措施。
  - chunked prefill 的残余问题（协商后仍可能失败） (correctness): 接受为已知残余，不在此 PR 修复。

风险与影响

- 风险：主要风险在于移动协商位置可能影响其他依赖早期协商的逻辑，但 PR 通过测试验证了行为正确。add_chunked_req 中固定报告 True 可能导致在极端情况下 delayer 误判，但 mid-chunk 请求必须继续，因此是合理折衷。该变更位于核心调度路径，对 DP attention 模式下的所有模型有影响，但非 DP 场景不受影响。
- 影响：影响范围限定在使用 PrefillDelayer 的 DP attention 部署场景（如 DeepSeek 系列模型）。修复后能正确聚合各 rank 的 prefill 能力，避免混合转发带来的性能下降和潜在错误。对非 DP 模式无影响。团队需要关注该修复可能带来的延迟行为变化，但总体是更符合预期。

- 风险标记：核心调度路径，DP attention 依赖，残余问题未完全解决

关联脉络

- PR #31682 Turn on breakable prefill cuda graph for dp attention by default: 同为 DP attention 调度优化，涉及 prefill 与 decode 分离。