PR #23456 完整报告

sgl-project/sglang

[SPEC V2] fix: skip stale state updates in spec-v2 overlap

合并时间：2026-05-10 14:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23456

执行摘要

- 一句话：修复 Spec-V2 重叠调度中过时状态更新导致的 KV 记账错误
- 推荐动作：建议阅读核心逻辑变更和 review 讨论，理解异步过时状态处理的设计权衡；全局指标一致性问题可作后续优化方向。

功能与动机

根据 PR 描述和代码注释，overlap 调度中 *decode* 结果可能后到达，但旧代码无条件应用 *accept_lens*，导致已结束请求的 KV 状态被错误延长并产生虚假的推测接受统计。需要跳过对过时请求的状态更新以维护正确性。

实现拆解

1. 在 _resolve_spec_overlap_token_ids 循环中，先根据 *accept_lens* 将 predict_tokens 追加到列表（无论请求状态如何）。
2. 若请求已撤回（req.is_retracted），直接 continue，因为 reset_for_retract() 已清零所有 KV 账目。
3. 若请求已完成（req.finished()），仅将 kv_committed_len 减 1（回滚 pre-claimed bonus slot），跳过统计更新。
4. 正常请求则按原逻辑更新 kv_committed_len、spec_verify_ct、spec_accepted_drafts 及历史分布。
5. 无测试或配置配套改动。

关键文件：

- python/sglang/srt/managers/scheduler_output_processor_mixin.py（模块 调度器；类别 source；类型 core-logic；符号 _resolve_spec_overlap_token_ids）：修改了 _resolve_spec_overlap_token_ids 方法，增加对 retracted 和 finished 请求的状态跳过逻辑，是修复的核心变更。

关键符号：_resolve_spec_overlap_token_ids

关键源码片段

python/sglang/srt/managers/scheduler_output_processor_mixin.py

修改了 _resolve_spec_overlap_token_ids 方法，增加对 retracted 和 finished 请求的状态跳过逻辑，是修复的核心变更。

```
def _resolve_spec_overlap_token_ids(
    self: Scheduler, result: GenerationBatchResult, batch: ScheduleBatch
) -> List[List[int]]:
    """Resolve the padding next token ids for speculative decoding with overlap."""
    assert result.next_token_ids.is_cpu
    assert result.accept_lens.is_cpu

    next_token_ids = result.next_token_ids.tolist()
    accept_lens = result.accept_lens.tolist()
    result.num_accepted_drafts = sum(accept_lens) - len(batch.reqs)
    result.num_accepted_drafts_per_req_cpu = [x - 1 for x in accept_lens]

    predict_tokens = []
    stride = self.draft_worker.speculative_num_draft_tokens

    for i, req in enumerate(batch.reqs):
        # 先填充 predict_tokens，无论请求状态如何，都使用接收的 token 序列
        predict_tokens.append(
            next_token_ids[i * stride: i * stride + accept_lens[i]]
        )

        # 如果请求已经被撤回，reset_for_retract() 已经清空了 committed 和 allocated KV，
        # 所以跳过所有状态更新。
        if req.is_retracted:
            continue

        # 如果请求已经在之前完成，则只需回滚 bonus slot（prepare_for_decode
        时预声明的槽位），
        # 不更新统计信息，避免重复或错误累积。
        if req.finished():
            # -1 because prepare_for_decode pre-claimed the bonus slot.
            req.kv_committed_len -= 1
            continue

        # 正常请求：根据 accept_lens 更新 KV 长度和推测接受统计
        req.kv_committed_len += accept_lens[i] - 1
        req.spec_verify_ct += 1

        accepted_draft_tokens = result.num_accepted_drafts_per_req_cpu[i]
        req.spec_accepted_drafts += accepted_draft_tokens
        req.update_spec_acceptance_histogram(accepted_draft_tokens)

    return predict_tokens
```

评论区精华

gemini-code-assist[bot] 评论了两个 medium 优先级的建议：对于 retracted 和 finished 请求，也应从全局 result.num_accepted_tokens 中减去对应接受数，以保持推测效率指标的一致性。该建议未被采纳，全局指标可能仍有偏差。

- 跳过 retracted 请求时应调整全局 num_accepted_tokens (correctness): 未在本次 PR 中处理，可能留待后续评估。
- 跳过 finished 请求时应调整全局 num_accepted_tokens (correctness): 未在本次 PR 中处理，可能留待后续评估。

风险与影响

- 风险：变更集中于单函数，风险较低。但条件判断错误（如误判正常请求为 finished/retracted）会导致状态更新遗漏，反之则可能仍有腐化。全局指标未调整可能引发监控误解。
- 影响：仅影响 Speculative V2 overlap 调度路径。修复后，已结束请求的 KV 记账不再被错误延长，推测接受统计更准确。用户无感知，但调试时指标更可靠。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR