PR #26126 完整报告

sgl-project/sglang

[RL] [Spec v2] Use stop-aware seqlen for returned topk metadata

合并时间：2026-05-23 09:13

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26126

执行摘要

- 一句话：修复 speculative decoding 下 routed topk 元数据越界问题
- 推荐动作：值得合并，修复逻辑清晰且已有生产验证（upstream 自 sglang-miles）。

功能与动机

Speculative decoding 可能导致 req.output_ids 比 finished_len 更长，但已有 output_ids_through_stop 正确反映停用后长度。topk 元数据收集未使用该边界，会包含 trailing tokens 的行，返回给用户或下游时产生脏数据。PR 描述指出 'This upstreams the corresponding sglang-miles fix from commit 71bda1af'。

实现拆解

变更仅涉及 batch_result_processor.py，两个方法同步调整：

1. _maybe_collect_routed_experts：将 seqlen 从 req.seqlen 改为 len(req.origin_input_ids) + len(req.output_ids_through_stop)，用于调用 capturer.get_topk 和计算 expected_rows；warning 日志新增 raw_seqlen 参数。
2. _maybe_collect_indexer_topk：同样将 seqlen 从 req.seqlen 改为 len(req.origin_input_ids) + len(req.output_ids_through_stop)。
3. 测试配套：本次未新增测试，仅通过 py_compile 和 git diff --check 验证。

关键文件：

- python/sglang/srt/managers/scheduler_components/batch_result_processor.py（模块 调度器；类别 source；类型 core-logic；符号 _maybe_collect_routed_experts, _maybe_collect_indexer_topk）：核心文件，修改了 routed experts 和 indexer topk 元数据收集时的长度计算逻辑。

关键符号：_maybe_collect_routed_experts, _maybe_collect_indexer_topk

关键源码片段

python/sglang/srt/managers/scheduler_components/batch_result_processor.py

核心文件，修改了 routed experts 和 indexer topk 元数据收集时的长度计算逻辑。

```
# python/sglang/srt/managers/scheduler_components/batch_result_processor.py

def _maybe_collect_routed_experts(self, req: Req):
    if not req.return_routed_experts:
        return
    capturer = get_global_experts_capturer()
    if capturer is None:
        return
    start_len = req.routed_experts_start_len
    # 使用 output_ids_through_stop 计算逻辑长度，而非 raw seqlen
    seqlen = len(req.origin_input_ids) + len(req.output_ids_through_stop)
    req.routed_experts = capturer.get_topk(
        req_pool_idx=req.req_pool_idx,
        seqlen=seqlen,
        req_to_token_pool=self.req_to_token_pool,
        start_len=start_len,
    )
    expected_rows = max(0, seqlen - 1 - start_len)
    if (
        req.routed_experts is not None
        and req.routed_experts.shape[0] != expected_rows
    ):
        # 仍记录 raw_seqlen 用于调试
        logger.warning(
            "routed_experts row-count mismatch for req %s: got %d, expected %d "
            "(seqlen=%d, raw_seqlen=%d, cached_tokens=%d, start_len=%s). "
            "This indicates a silent bug.",
            req.rid,
            req.routed_experts.shape[0],
            expected_rows,
            seqlen,
            req.seqlen,
            req.cached_tokens,
            req.routed_experts_start_len,
        )

def _maybe_collect_indexer_topk(self, req: Req):
    capturer = get_global_indexer_capturer()
    if capturer is None:
        return
    # 同样使用 stop-aware 长度
    seqlen = len(req.origin_input_ids) + len(req.output_ids_through_stop)
    req.indexer_topk = capturer.get_topk(
        req_pool_idx=req.req_pool_idx,
        seqlen=seqlen,
        req_to_token_pool=self.req_to_token_pool,
    )
```

评论区精华

- 暂无高价值评论线程

风险与影响

- 风险：风险较低，仅替换了局部变量 seqlen 的来源，调用 capturer.get_topk 的签名未变；warning 日志兼容旧字段，不影响生产运行。若能补充单元测试覆盖 speculative decoding 场景将更稳健。
- 影响：直接修复 return_routed_experts 或 return_indexer_topk 开启时、speculative decoding 下的元数据越界 bug，影响范围限于该功能路径。
- 风险标记：缺少测试覆盖

关联脉络

- PR #26108 FutureMap: debug-assert that gather sees a stashed value: 同为调度器层 speculative decoding 相关问题，涉及 overlap_utils 和输出边界处理。
- PR #26085 drop FutureIndices wrapper class: 同为 speculative decoding 和调度器重构，修改了 overlap_utils 等文件。