PR #25015 完整报告

sgl-project/sglang

Fix Eagle draft decode positions

合并时间：2026-05-13 05:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25015

执行摘要

- 一句话：修复 Eagle 投机解码 draft decode 位置错误
- 推荐动作：建议优先合并。该 PR 修复了一个明显的 off-by-one 错误，逻辑正确，改动量小，风险可控。虽然缺少自动化测试验证，但 PR 作者已通过 debug logging 确认修复。后续应考虑补全 Eagle speculative decoding 的 e2e 测试。

功能与动机

根据 PR body 的描述，draft decode 步骤中 RoPE position IDs 存在偏差：例如在 draft extend for prefill 后，draft decode forward 应使用 position=7，但实际使用了 8（错误），导致后续 KV cache 索引和 attention 计算不正确。PR 作者通过 debug logging 确认了该问题。

实现拆解

1. 移动 position 增量时机（eagle_worker.py & eagle_worker_v2.py）：在 draft_forward 方法的循环中，将 forward_batch.positions.add_(1) 从 step 内最前（设置 input_ids 等之后立即执行）移到 step 内最后（隐藏在 hidden_states = logits_output.hidden_states 之后），确保每个 draft decode forward 调用时 positions 已经是正确的上一个 token 的位置。
2. 恢复 position（eagle_draft_cuda_graph_runner.py）：在 CUDA graph capture 的 run_once() 函数中，draft_forward 会 in-place 修改 forward_batch.positions（增加 speculative_num_steps - 1）。为了不影响后续 capture 或 replay 的起始状态，在 run_once() 末尾添加 forward_batch.positions.sub_(self.eagle_worker.speculative_num_steps - 1) 将 positions 恢复到调用前的值。同时更新了备份注释。

关键文件：

- python/sglang/srt/speculative/eagle_worker.py（模块 投机解码；类别 source；类型 core-logic；符号 draft_forward）：核心 Eagle draft worker v1，修复了 draft_forward 中 positions.add_(1) 的错误时机
- python/sglang/srt/speculative/eagle_worker_v2.py（模块 投机解码；类别 source；类型 core-logic；符号 draft_forward）：Eagle draft worker v2，与 v1 同步修复相同的错误
- python/sglang/srt/speculative/eagle_draft_cuda_graph_runner.py（模块 投机解码；类别 source；类型 core-logic；符号 run_once）：CUDA graph capture 中 restore

positions 的补偿逻辑

关键符号：draft_forward, run_once

关键源码片段

python/sglang/srt/speculative/eagle_worker.py

核心 Eagle draft worker v1，修复了 draft_forward 中 positions.add_(1) 的错误时机

```
# 在 draft_forward 的循环中，调整 position 增量时机
for i in range(self.speculative_num_steps):
    # ... (select_top_k_tokens, 设置 input_ids, out_cache_loc, attn_backend etc.)

    # 之前：positions.add_(1) 在此处（错误），导致当前 forward 使用已递增的位置
    # forward_batch.positions.add_(1) # 已删除

    # Run forward
    logits_output = self.draft_model_runner.forward(
        forward_batch, skip_attn_backend_init=True
    ).logits_output
    # ... (softmax, topk, hidden_states etc.)

    # 现在：将增量移到 forward 之后，使得当前 forward 使用的是正确（未递增）的位置
    forward_batch.positions.add_(1)
```

python/sglang/srt/speculative/eagle_draft_cuda_graph_runner.py

CUDA graph capture 中 restore positions 的补偿逻辑

```
# 在 CUDA graph capture 的 run_once 函数中，备份并恢复 forward_batch 状态
def run_once():
    # Clean intermediate result cache for DP attention
    forward_batch.dp_local_start_pos = forward_batch.dp_local_num_tokens = None
    set_dp_buffer_len(
        global_dp_buffer_len,
        num_tokens,
        forward_batch.dp_padding_mode.is_max_len(),
    )
    set_is_extend_in_batch(False)

    # Backup fields that are modified in-place in `draft_forward`.
    output_cache_loc_backup = forward_batch.out_cache_loc
    hidden_states_backup = forward_batch.spec_info.hidden_states

    ret = self.eagle_worker.draft_forward(forward_batch)

    forward_batch.out_cache_loc = output_cache_loc_backup
    forward_batch.spec_info.hidden_states = hidden_states_backup
    # 新增：恢复 positions，因为 draft_forward 中对其进行了 in-place add_(n-1)
    forward_batch.positions.sub_(self.eagle_worker.speculative_num_steps - 1)
    return ret
```

评论区精华

PR 没有 review 评论，只有作者触发的 /tag-and-rerun-ci 命令和 gemini-code-assist 的 quota 警告。没有实质性讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 回归风险低：改动仅涉及 in-place 操作的顺序调整和一处简单的减法恢复，逻辑清晰，且两个 worker 版本均同步修改。
  - CUDA graph capture 影响：eagle_draft_cuda_graph_runner.py 中的 run_once 函数用于 CUDA graph 捕获和回放，修改后增加了 positions.sub_ 操作，可能影响 graph 的捕获行为（但只影响 capture，不影响 replay，因为 replay 时 forward_batch 是新传入的）。
  - 缺少测试覆盖：PR 仅在 PR 描述中标记了“Run Eagle speculative decoding e2e test”但未实际执行，且没有提交新的测试用例。
- 影响：
  - 用户影响：修复了 Eagle 投机解码 draft 阶段的位置错误，提升生成质量和吞吐。所有使用 Eagle 投机解码的用户受益。
  - 系统影响：无性能开销，仅调整 in-place 操作顺序和一次额外的减法。
  - 团队影响：代码量小，变化集中，易于 review 和回滚。
  - 风险标记：缺少测试覆盖

关联脉络

- PR #25037 spec: STANDALONE skips hidden_states end-to-end (Optional schema + None-safe consumers): 同为 speculative-decoding 相关，涉及 Eagle 工作流的 hidden_states 处理优化
- PR #25030 [Spec] Multi-layer mamba scatter cleanup; fix positional call bug: 同为 speculative-decoding bugfix，涉及 multi_layer_eagle_worker 的位置相关调用
- PR #25038 [Spec] Rename accepted_indices -> accept_indices; drop _token_id suffix per Rule 5: 同一集群的 speculative-decoding 重构 PR