PR #24743 完整报告

sgl-project/sglang

fix(cuda_graph): zero out_cache_loc_swa on pad and use int32 (hybrid-SWA accuracy fix)

合并时间：2026-05-09 18:22

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24743

执行摘要

- 一句话：修复 hybrid-SWA 精度回归，零化填充索引并修复 dtype
- 推荐动作：建议立即合并此 PR。它修复了一个关键的精度回归，变更简洁且经过良好推理。开发者在 hybrid-SWA 模型上工作时值得仔细阅读此 PR，以理解 CUDA Graph 填充路径下索引管理的陷阱。

功能与动机

PR #23552 为 hybrid-SWA 模型添加了预计算的 out_cache_loc_swa 缓冲，以加速 CUDA Graph 捕获。但引入了一个正确性问题：在 populate_from_forward_batch 中，当批次大小从 raw_bs 填充到 bs 时，out_cache_loc_swa 未被清零，导致上一次 replay 留下的虚假 SWA 索引使填充位置的 token 错误地写入活跃请求的 SWA 槽位，损坏 KV 缓存。用户可见的症状是 MiMoV2.5 Pro 上的精度显著下降，如 AIME24-25 从 80.3 降至 90.3（实际上是修复后提升）。此外，dtype 不匹配问题在 #23552 的 review 中被提出但推迟修复。

实现拆解

1. 在 populate_from_forward_batch 中清零填充区域：在 cuda_graph_runner.py 的填充分支（if bs != raw_bs:）中，新增对 self.out_cache_loc_swa 的 zero_() 调用，确保填充位置的索引为 0（SWA sentinel 值），从而避免 KV 缓存损坏。此举与 piecewise_cuda_graph_runner.py 中已有的正确行为保持一致。
2. 统一 out_cache_loc_swa 的 dtype 为 int32：在 DecodeInputBuffers.create()（cuda_graph_runner.py）和 PiecewiseCudaGraphRunner.__init__()（piecewise_cuda_graph_runner.py）中，将分配 out_cache_loc_swa 时的 dtype 从 torch.int64 改为 torch.int32，以匹配下游函数 translate_loc_from_full_to_swa 和 set_kv_buffer 的期望类型，避免隐式类型转换带来的潜在问题。

关键文件：

- python/sglang/srt/model_executor/cuda_graph_runner.py（模块 CUDA Graph 运行器；类别 source；类型 data-contract；符号 DecodeInputBuffers.create, DecodeInputBuffers.populate_from_forward_batch）：核心修复文件：在 populate_from_forward_batch 中新增 out_cache_loc_swa.zero_()，并将 out_cache_loc_swa 的 dtype 从 int64 改为 int32。

- python/sglang/srt/model_executor/piecewise_cuda_graph_runner.py（模块 CUDA Graph 运行器；类别 source；类型 data-contract；符号 PiecewiseCudaGraphRunner.init）：配套修复：将 out_cache_loc_swa 的 dtype 从 int64 改为 int32，保持一致性。

关键符号：DecodeInputBuffers.create, DecodeInputBuffers.populate_from_forward_batch, PiecewiseCudaGraphRunner.init

关键源码片段

python/sglang/srt/model_executor/cuda_graph_runner.py

核心修复文件：在 populate_from_forward_batch 中新增 out_cache_loc_swa.zero_()，并将 out_cache_loc_swa 的 dtype 从 int64 改为 int32。

```
# DecodeInputBuffers.create() 中分配 out_cache_loc_swa（line 184-188）
out_cache_loc_swa = (
    torch.zeros((max_num_token,), dtype=torch.int32) # 原为 torch.int64，改为与下游一致的 int32
    if is_hybrid_swa
    else None
)

# populate_from_forward_batch() 中填充分支（line 289-301）
if bs != raw_bs:
    self.seq_lens.fill_(seq_len_fill_value)
    self.out_cache_loc.zero_()
    # 新增：清零 out_cache_loc_swa，防止残留索引损坏 KV 缓存
    if self.out_cache_loc_swa is not None:
        self.out_cache_loc_swa.zero_()
    if self.mamba_track_indices is not None:
        self.mamba_track_indices.zero_()
    if self.mamba_track_mask is not None:
        self.mamba_track_mask.fill_(False)
```

评论区精华

Reviewer@hnyls2002批准了PR，并CC@merrymercy以跟进后续的int32/int64清理工作。未发现其他争议或未解决的问题。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更范围小（仅 2 个文件，共 8 行新增 / 修改），并且清零逻辑和 dtype 一致性与 piecewise runner 已使用的模式完全一致。没有引入新的配置或 API，不会对非 hybrid-SWA 模型产生影响。回归风险低，因为修复的是填充路径，在非填充情况下行为不变。

- 影响：对用户而言，修复了 MiMoV2.5 Pro 等 hybrid-SWA 模型在启用 CUDA Graph 时的精度回归，提升显著（如 AIME24-25 提升 10 个百分点）。对系统而言，仅影响 CUDA Graph 场景下的 SWA 模型，不改变其他功能。对团队而言，解决了此前 review 中遗留的 dtype 问题，减少技术债务。影响程度：中等重要性，但针对特定模型的高精度场景。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #23552 feat: pre-compute out_cache_loc_swa in DecodeInputBuffers for hybrid-SWA fast path: 本 PR 引入了 out_cache_loc_swa 预计算，但遗漏了填充清零和 dtype 问题，是当前 PR 修复的直接前序。
- PR #24617 fix(fa3): translate page table to SWA loc in EAGLE3 topk>1 spec metadata: 同为 hybrid-SWA 相关修复，涉及 SWA 页表翻译，属于同一功能域。