PR #24138 完整报告

sgl-project/sglang

[SWA] Ensure we use pre-computed SWA cache location during prefill

合并时间：2026-05-01 15:01

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/24138

执行摘要

- 一句话：修复 prefill 阶段 SWA cache location 被忽略问题
- 推荐动作：建议精读。此 PR 展示了一个典型的“使用预计算值替代重复计算”的优化模式，同时也体现了 review 中发现的“直接引用状态属性 vs 通过 forward_batch 传递”的设计陷阱。对于维护 SWA 或类似缓存机制的同学，该变更和讨论值得学习。

功能与动机

在 sliding-window attention 模型中，prefill 阶段已经通过 SWAKVPool 预计算了 SWA cache location 并存储在 forward_batch.out_cache_loc_swa 中，但 _get_layer_cache_loc 方法始终调用 translate_loc_from_full_to_swa 从完整 cache location 重新转换，忽略了预计算的值。这会导致在不支持重新转换的 CUDA graph 等场景下出现错误。PR 从 meta-llama/prod_inference 上游同步修复。

实现拆解

1. 修改 _get_layer_cache_loc 方法：将参数从 cache_loc: torch.Tensor 改为 forward_batch: ForwardBatch，以便访问 forward_batch.out_cache_loc_swa 和 forward_batch.out_cache_loc。
2. 优先返回预计算的 SWA cache location：当 forward_batch.out_cache_loc_swa 不为 None 时，直接返回该值，避免调用 translate_loc_from_full_to_swa 进行冗余转换。
3. 更新调用点 _fused_fp8_set_kv_buffer：将 self._get_layer_cache_loc(layer, forward_batch.out_cache_loc) 调整为 self._get_layer_cache_loc(layer, forward_batch)，以匹配新的函数签名。

关键文件：

- python/sglang/srt/layers/attention/trtllm_mha_backend.py（模块 注意力层；类别 source；类型 core-logic；符号 _get_layer_cache_loc, _fused_fp8_set_kv_buffer）：唯一修改的文件，核心变更在 _get_layer_cache_loc 方法及其调用点 _fused_fp8_set_kv_buffer，修复 SWA prefill 阶段缓存位置被忽略的 bug。

关键符号：_get_layer_cache_loc, _fused_fp8_set_kv_buffer

关键源码片段

python/sglang/srt/layers/attention/trtllm_mha_backend.py

唯一修改的文件，核心变更在 _get_layer_cache_loc 方法及其调用点 _fused_fp8_set_kv_buffer，修复 SWA prefill 阶段缓存位置被忽略的 bug。

```
def _get_layer_cache_loc(
    self,
    layer: RadixAttention,
    forward_batch: ForwardBatch,
) -> torch.Tensor:
    """Return cache locations in the correct index space for the given layer.

    如果该层是 SWA 层，优先返回预计算的 SWA cache location（`out_cache_loc_swa`），
    避免从完整 cache location 重新转换，后者在 CUDA graph 等场景下可能不准确。
    """
    if self.use_sliding_window_kv_pool:
        _, is_swa = self._swa_kv_pool.layers_mapping[layer.layer_id]
        if is_swa:
            # 如果 forward_batch 已经预计算了 SWA cache location，直接返回
            if forward_batch.out_cache_loc_swa is not None:
                return forward_batch.out_cache_loc_swa
            # 否则从完整 cache location 实时转换（作为 fallback）
            return self._swa_kv_pool.translate_loc_from_full_to_swa(
                forward_batch.out_cache_loc
            )
    # 非 SWA 层，直接返回完整 cache location
    return forward_batch.out_cache_loc
```

评论区精华

Review 中 gemini-code-assist[bot] 指出，如果直接使用 self._swa_kv_pool.swa_loc 作为缓存位置，会引入跨批次使用过期数据（stale data）的风险。具体来说，swa_loc 是 SWAKVPool 的状态属性且是批次相关的，在 ModelRunner._forward_raw 中仅当当前批次提供新值时才会更新，从未被清除。若后续批次未提供新的 swa_loc，该方法将返回上一批次的 swa_loc，可能导致错误的 KV cache 写入或形状不匹配。最终提交的代码已采纳该建议，改用 forward_batch.out_cache_loc_swa（从 ForwardBatch 中获取）代替 self._swa_kv_pool.swa_loc，避免了状态残留问题。

- 使用 forward_batch 传递预计算位置 vs 直接引用 SWAKVPool 状态属性 (design): 最终代码采用 forward_batch.out_cache_loc_swa（由 ForwardBatch 传递），而不是直接引用 SWAKVPool 的状态属性，避免了状态残留问题。

风险与影响

- 风险：风险极低。变更仅涉及一个私有方法 _get_layer_cache_loc 及其调用点，改动量小。主要风险在于：若 forward_batch.out_cache_loc_swa 与 out_cache_loc 的语义不一致（例如在非 SWA 层或 decode 阶段），但现有逻辑已通过 is_swa 判断和 None 检查加以保护，不会误用。此外，该路径仅在 use_sliding_window_kv_pool 为 True 时生效，不影响

非 SWA 模型。

- 影响：影响范围限于使用 TRTLLM 注意力后端且启用 sliding-window KV pool 的模型（如某些 Blackwell 平台上的模型）。对用户透明，但可修复在 CUDA graph 等场景下 SWA 模型 prefill 阶段未使用预计算缓存位置导致的正确性问题。无性能回退，因为免去了多余的坐标转换调用反而可能带来轻微性能提升。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR