PR #29352 完整报告

sgl-project/sglang

[bug2] skip swa recovery on locked full kv

合并时间：2026-07-01 07:07

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/29352

执行摘要

- 一句话：修复 SWA 恢复时覆盖锁定 Full KV
- 推荐动作：值得精读，尤其是对 radix cache 和 SWA 组件交互边界感兴趣的同学。新增的 _restore_device_value_with_locked_full 和 _recover_tombstone_keeping_locked_full 设计体现了在锁定资源下做部分恢复的通用思路。建议同时阅读关联 PR #29353 理解完整修复上下文。

功能与动机

PR body 说明："This PR fixes an SWA recovery path that could overwrite Full KV still protected by an active request. SWA may try to recover a tombstoned SWA component from fresh insert data. That recovery must not replace the node's Full KV if the Full component is still locked." 属于一系列 radix cache 正确性修复的一部分（源自 #29349）。

实现拆解

1. swa_radix_cache.py：在 _insert_helper 的 Branch 1 和 Branch 2（SWA tombstone 恢复分支）中增加 node.full_lock_ref > 0 判断。当 Full 被锁定时，调用新增方法 _recover_tombstone_keeping_locked_full 保留原 Full KV 值，仅从传入 KV 恢复 SWA 组件并更新映射，同时释放传入的 Full KV 切片。
2. unified_cache_components/swa_component.py：新增 _restore_device_value_with_locked_full 方法，实现类似逻辑：利用旧 Full 值调用 set_full_to_swa_mapping 建立映射，释放传入 Full KV，再恢复 SWA 设备值。在 update_component_on_insert_overlap 的两个 Branch 中插入对应的锁检查分支。
3. allocator/swa.py：统一 set_full_to_swa_mapping 中的 dtype 转换：始终将 full_indices 转为 int64，swa_indices 转为 mapping 表 dtype，移除之前针对 NPU 的条件分支。
4. 测试：新增 test_swa_unfinished_recovery_preserves_locked_full_value，构造节点 SWA 被 evict、Full 被锁定的场景，调用 cache_unfinished_req 触发恢复，断言 Full KV 保持不变且 SWA 正确恢复。

关键文件：

- python/sglang/srt/mem_cache/swa_radix_cache.py（模块 SWA 缓存；类别 source；类型 core-logic；符号 _recover_tombstone_keeping_locked_full）：核心修复文件，在传统

SWA 缓存插入路径中增加 locked-Full 保护逻辑

- python/sglang/srt/mem_cache/unified_cache_components/swa_component.py（模块 SWA 组件；类别 source；类型 core-logic；符号 _restore_device_value_with_locked_full）：统一缓存 SWA 组件的插入重叠处理，增加 locked-Full 恢复方法
- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_swa_unfinished_recovery_preserves_locked_full_value）：新增单元测试验证 locked-Full 保护场景
- python/sglang/srt/mem_cache/allocator/swa.py（模块 分配器；类别 source；类型 core-logic；符号 set_full_to_swa_mapping）：修复 set_full_to_swa_mapping 的 dtype 转换，避免在 locked-Full 路径中因类型不匹配报错

关键符号：_recover_tombstone_keeping_locked_full, _restore_device_value_with_locked_full, set_full_to_swa_mapping, test_swa_unfinished_recovery_preserves_locked_full_value

关键源码片段

python/sglang/srt/mem_cache/swa_radix_cache.py

核心修复文件，在传统 SWA 缓存插入路径中增加 locked-Full 保护逻辑

```
# python/sglang/srt/mem_cache/swa_radix_cache.py
# Branch 1: 所有 SWA token 未 evicted，直接恢复
if swa_evicted_seqlen <= total_prefix_length:
    if node.full_lock_ref > 0:
        # Full KV 仍被活跃请求锁定，保留原 Full 值，
        # 仅从传入 value 恢复 SWA 并更新映射，
        # 同时释放传入的 Full 切片避免泄漏。
        self._recover_tombstone_keeping_locked_full(
            node, value[:prefix_len]
        )
    else:
        # 经典路径：释放原 Full token，用新值覆盖节点
        self.token_to_kv_pool_allocator.free(node.value[:prefix_len])
        node.value = value[:prefix_len].clone()
        node.swa_tombstone = False
        self.swa_lru_list.insert_mru(node)
        self.swa_evictable_size_ += len(node.value)
elif swa_evicted_seqlen < total_prefix_length + prefix_len:
    # Branch 2: 部分 SWA token evicted，需 split 后恢复
    start_update_idx = swa_evicted_seqlen - total_prefix_length
    if node.full_lock_ref > 0:
        # 先 split，然后对后缀部分调用 locked-Full 恢复
        self._split_node(node.key, node, start_update_idx)
        self._recover_tombstone_keeping_locked_full(
            node, value[start_update_idx:prefix_len]
        )
        self.token_to_kv_pool_allocator.free(value[:start_update_idx])
```

```
    else:
        # 经典路径：释放被覆盖的 Full 切片，split 后设置新值
        self.token_to_kv_pool_allocator.free(
            node.value[start_update_idx:prefix_len]
        )
        self._split_node(node.key, node, start_update_idx)
        node.value = value[start_update_idx:prefix_len].clone()
        self.token_to_kv_pool_allocator.free(value[:start_update_idx])
        node.swa_tombstone = False
        self.swa_lru_list.insert_mru(node)
        self.swa_evictable_size_ += len(node.value)
```

python/sglang/srt/mem_cache/unified_cache_components/swa_component.py

统一缓存 SWA 组件的插入重叠处理，增加 locked-Full 恢复方法

```
# python/sglang/srt/mem_cache/unified_cache_components/swa_component.py

def _restore_device_value_with_locked_full(
    self,
    node: UnifiedTreeNode,
    full_value: torch.Tensor, # 节点已有的锁定 Full KV
    incoming_full_value: torch.Tensor, # 插入请求的新 Full 切片
) -> None:
    """当 Full 被锁定时，保留原有 Full 值，只从 incoming 恢复 SWA。"""
    allocator = self.cache.token_to_kv_pool_allocator
    # 将 incoming Full 转换为 SWA 索引
    swa_value = self._translate_full_to_swa(incoming_full_value)
    # 建立从原 Full 到新 SWA 的映射
    allocator.set_full_to_swa_mapping(full_value, swa_value)
    # 清理 incoming 的 Full 索引映射（避免残留）
    allocator.full_to_swa_index_mapping[incoming_full_value.to(torch.int64)] = 0
    # 释放 incoming 的 Full 分配（不需要保留）
    allocator.full_attn_allocator.free(incoming_full_value)
    # 将 SWA 值写入节点并恢复 LRU 状态
    self._restore_device_value(node, swa_value)

# 在 update_component_on_insert_overlap 中使用（简略）：
if swa_evicted_seqlen <= total_prefix_len:
    if full_cd.lock_ref > 0:
        self._restore_device_value_with_locked_full(node, full_cd.value, value_slice)
        return 0
    # ... 经典路径
```

评论区精华

1. 正确性担忧：hanming-lu 反馈 "the server crashes for me"，要求先解决 crash 再合并。yaof20 解释修复源自内部 RL 任务验证，并指出需结合事务回滚 PR（#29353）才能完全避免不一致。后续进一步调试和补充修改后 crash 问题解决。

2. 精度影响：hanming-lu 提出 "minor accuracy implications ... full and swa kv are from different forward"，但认为没有更干净的方法且与 mamba radix cache 已有做法一致，最终接受。
3. 同步修改 swa_radix_cache.py：hanming-lu 要求对传统的 SWA radix cache 应用完全相同逻辑，并附上 diff patch。yaof20 采纳后修改并通过测试。

- 修复导致 crash，需要先验证正确性 (correctness): yaof20 解释修复源自内部 RL 任务，后续补充调试和修改后 crash 问题解决。
- 精度影响：Full 和 SWA 来自不同 forward (correctness): yaof20 认为与 mamba radix cache 现有做法一致，且可以接受。hanming-lu 最终同意并合并。
- 需要为 swa_radix_cache.py 应用相同逻辑 (design): yaof20 采纳补丁并修改、提交，后续测试通过。

风险与影响

- 风险：
  1. 精度风险：Full 和 SWA KV 可能来自不同 forward 导致推理结果微小差异，但该做法与 mamba radix cache 已有策略一致，风险可控。
  2. 依赖事务回滚：单独应用本 PR 可能在某些边界情况（match_prefix 在缺少 SWA 时拒绝匹配）下仍存在不一致，需与 #29353 事务回滚配合使用。
  3. 锁定状态误判：full_lock_ref > 0 的逻辑是否正确涵盖所有锁定场景？新增的单元测试只覆盖特定配置（page_size=1, sliding_window_size=4），其他配置可能需额外验证。 - 影响：直接解决使用 SWA（Sliding Window Attention）并启用 radix cache 时可能发生的 KV 数据损坏问题，提升推理稳定性。影响所有在 sglang SRT 中使用滑动窗口注意力的模型（如 DeepSeek 系），在并发请求共享前缀场景下避免跨请求 KV 泄露。 - 风险标记：可能精度影响，需要依赖事务回滚，锁定状态判断关键

关联脉络

- PR #29349 Parent PR: radix cache bug fixes (split): 本 PR 是 #29349 的拆分 3/5，源自同一修复系列。
- PR #29353 transactional rollback change: 讨论中提及需要配合使用以避免 match_prefix 不一致，属于同一完整修复方案。