

PR #23696 完整报告

sgl-project/sglang

[BugFix][HiMamba] Fix host-protected node deletion in HiMamba tombstone del

合并时间: 2026-05-01 21:57

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/23696>

执行摘要

- 一句话: 修复 HiMamba 墓碑级联删除忽略 host 引用保护
- 推荐动作: 建议阅读此 PR 以理解多级缓存中 host 引用计数在逐出算法中的关键作用。测试设计值得借鉴: 通过模拟 HiMambaRadixCache 内部状态 (而不依赖完整初始化) 来高效验证边界条件。但需注意当前修复仍是临时方案, 后续应跟踪 UnifiedRadixTree 迁移进展。

功能与动机

此 PR 是 #22592 的后续修复。用户 @riZZzhik 在 #22592 中报告在持续高负载下仍会触发相同的 `AssertionError: parent does not have child key` 崩溃。经分析, `_iteratively_delete_tombstone_leaf` 在级联删除时只检查了 `full_lock_ref` 和 `mamba_lock_ref`, 但忽略了 `host_ref_counter` 和 `host_mamba_ref_counter`, 导致 host 节点 (如正在 prefetch/backup) 被错误删除, 破坏父 -> 子关系。详见 PR body 和评论。

实现拆解

1. 在 `hi_mamba_radix_cache.py` 的 `_iteratively_delete_tombstone_leaf` 方法中, 在 while 循环内新增一组条件判断: 如果当前父节点的 `host_ref_counter > 0` 或 `host_mamba_ref_counter > 0`, 则立即停止级联删除。这确保了正在被 host 引用 (如正在 prefetch/backup) 的节点不会被错误地从树中移除。
2. 在 `mamba_radix_cache.py` 的 `cache_unfinished_req` 方法中, 调用 `self.insert()` 时传入 `chunked=chunked` 参数 (之前遗漏), 以对齐基类 `RadixCache.cache_unfinished_req` 的行为, 避免因缺少该参数导致的分片处理错误。
3. 在 `test/registered/unit/mem_cache/test_mamba_unittest.py` 中新增测试方法 `test_hi_mamba_tombstone_cleanup_respects_host_ref`, 通过构造一个 HiMambaRadixCache 实例并手动设置 `host_ref_counter`, 验证级联删除不会移除受 host 引用的父节点。
4. 在 `test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py` 中新增测试方法 `test_tombstone_cleanup_respects_locked_parent`, 验证 `UnifiedRadixCache._iteratively_delete_tombstone_leaf` 在父节点 `lock_ref > 0` 时也不会错误删除, 确保类似保护存在于统一缓存层。

关键文件:

- python/sglang/srt/mem_cache/hi_mamba_radix_cache.py (模块 HiMamba; 类别 source; 类型 core-logic; 符号 `_iteratively_delete_tombstone_leaf`) : 核心修复文件: 在 `_iteratively_delete_tombstone_leaf` 中新增 `host_ref_counter/host_mamba_ref_counter` 检查, 防止受 `host` 引用的节点被级联删除。
- test/registered/unit/mem_cache/test_mamba_unittest.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 `test_hi_mamba_tombstone_cleanup_respects_host_ref`, `RecordingCacheController`, `init`, `evict_device`) : 新增单元测试 `test_hi_mamba_tombstone_cleanup_respects_host_ref`, 直接验证 `hi_mamba_radix_cache` 中 `host` 引用保护逻辑的正确性。
- python/sglang/srt/mem_cache/mamba_radix_cache.py (模块 Mamba; 类别 source; 类型 core-logic; 符号 `cache_unfinished_req`) : 在 `cache_unfinished_req` 中补传 `chunked` 参数, 对齐基类行为, 避免分片处理错误。
- 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_tombstone_cleanup_respects_locked_parent`) : 新增测试 `test_tombstone_cleanup_respects_locked_parent`, 验证 `UnifiedRadixCache` 中类似的 `lock_ref` 保护, 确保统一缓存层也具备防御措施。

关键符号: `_iteratively_delete_tombstone_leaf`, `cache_unfinished_req`,
`test_hi_mamba_tombstone_cleanup_respects_host_ref`,
`test_tombstone_cleanup_respects_locked_parent`

关键源码片段

test/registered/unit/mem_cache/test_mamba_unittest.py

新增单元测试 `test_hi_mamba_tombstone_cleanup_respects_host_ref`, 直接验证 `hi_mamba_radix_cache` 中 `host` 引用保护逻辑的正确性。

```
def test_hi_mamba_tombstone_cleanup_respects_host_ref(self):
    # 手工构造 HiMambaRadixCache 内部状态, 避免复杂初始化
    tree = object.__new__(HiMambaRadixCache)
    root = TreeNode()
    parent = TreeNode()
    deleted = TreeNode()

    root.key = RadixKey([])
    parent.key = RadixKey([1])
    deleted.key = RadixKey([2])
    parent.parent = root
    deleted.parent = parent
    parent.value = torch.tensor([1], dtype=torch.int64)
    parent.protect_host() # 设置 host_ref_counter > 0
    root.children[parent.key.child_key(1)] = parent

    # 使用 RecordingCacheController 记录驱逐事件
    class RecordingCacheController:
        def __init__(self):
```

```

        self.device_evictions = []
        self.host_evictions = []
    def evict_device(self, value):
        self.device_evictions.append(value)
    def evict_host(self, value):
        self.host_evictions.append(value)

tree.root_node = root
tree.page_size = 1
tree.full_lru_list = LRUList(mamba=False)
tree.full_lru_list.insert_mru(parent)
tree.cache_controller = RecordingCacheController()
tree.full_evictable_size_ = len(parent.value)
tree.evictable_full_device_leaves = {parent}
tree.evictable_full_host_leaves = set()

result_node, full_evicted, mamba_evicted = (
    tree._iteratively_delete_tombstone_leaf(deleted)
)

# 断言：节点没有被级联删除，父节点仍在树中
self.assertIs(result_node, deleted)
self.assertEqual(full_evicted, 0)
self.assertEqual(mamba_evicted, 0)
self.assertIs(root.children[parent.key.child_key(1)], parent)
self.assertTrue(tree.full_lru_list.in_list(parent))
self.assertEqual(tree.cache_controller.device_evictions, [])
self.assertEqual(tree.cache_controller.host_evictions, [])

```

评论区精华

@riZZZhik 报告在应用 patch 后，中等负载下仍出现相同错误，但随后确认错误由其他因素引起（相关 Issue 评论）。@icepoint666 分析推测另一个潜在 BUG: `_insert_helper_host()` 更新 `last_access_time` 并刷新 `mamba` LRU，但未刷新 `full` LRU 中仍在 `device` 上的节点，可能导致后续 LRU 一致性问题。@riZZZhik 最后确认本 PR 在高负载下测试通过。

- `AssertionError` 在中等负载下仍然出现 (`correctness`): @riZZZhik 后来确认错误由其他因素引起，本 PR 修复有效。但 icepoint666 提出的 `full` LRU 刷新问题未在本 PR 中解决。
- 确认修复在高负载下有效 (`testing`): 修复通过用户实际高负载验证，无回归。

风险与影响

- 风险：主要风险是本修复仅解决了 `host` 引用保护，但未处理 `_insert_helper_host` 不刷新 `full` LRU 的潜在问题 (icepoint666 指出)。该问题可能导致 LRU 顺序错误进而影响逐出策略。不过当前修复是官方声明的临时方案，长期计划是迁移到 `UnifiedRadixTree`。此外，新增的测试覆盖了 `host` 引用保护的关键路径，但未覆盖高并发场景下的完整行为。
- 影响：影响范围限于使用 `HiMambaRadixCache` + `Mooncake` 存储 + `write_through` 策略的用户。修复后预计消除了持续高负载下的断言崩溃。对 `MambaRadixCache` 的修改（传递

chunked 参数) 是对齐基类行为, 不影响现有功能。测试的增强为未来重构提供了安全网。

- 风险标记: host 引用保护遗漏

关联脉络

- PR #22592 (原 PR 标题未知, 但本 PR 是它的后续修复): PR #22592 引入了 HiMambaRadixCache 但遗漏了 host_ref 检查, 导致本 PR 修复的 bug。