

PR #23537 完整报告

sgl-project/sglang

[BugFix] correct host leaf status check from evicted to backedup

合并时间: 2026-04-30 01:26

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/23537>

执行摘要

- 一句话: 修正 host leaf 状态检查条件 bug
- 推荐动作: 建议精读此 PR 的讨论和实现, 理解 HiCache 驱逐状态机的微妙之处。该修复展示了分布式缓存中状态字段语义必须精确对应业务含义的重要性。

功能与动机

原代码使用 `child.evicted` 判断子节点状态, 但 `evicted` 仅表示节点是否从 GPU 驱逐, 不表示其 host 备份仍存在。在 `write_back` 策略下, 当某个 token 的 host 数据因内存压力被清除后重新加载至 GPU, 其 `backedup` 仍为 `True` 而 `evicted` 为 `False`, 导致父节点被错误加入 `evictable_host_leaves`, 最终引发断言错误。关联 Issue #19212 描述了该崩溃场景。

实现拆解

修改 `python/sglang/srt/mem_cache/hiradix_cache.py` 中 `_update_host_leaf_status` 方法的条件判断, 将 `child.evicted` 替换为 `child.backedup`。此变更确保仅当子节点仍在 host 有备份数据时, 父节点才不会被当作可 host 驱逐的叶子节点。

关键文件:

- `python/sglang/srt/mem_cache/hiradix_cache.py` (模块 缓存管理; 类别 source; 类型 core-logic; 符号 `_update_host_leaf_status`): 核心缓存管理逻辑文件, 修复 host leaf 状态判断条件

关键符号: `_update_host_leaf_status`

关键源码片段

`python/sglang/srt/mem_cache/hiradix_cache.py`

核心缓存管理逻辑文件, 修复 host leaf 状态判断条件

```
def _update_host_leaf_status(self, node: TreeNode):
    # 节点未被驱逐或仍有锁引用时, 从可驱逐 host 集合中移除
    if not node.evicted or node.lock_ref > 0:
        if node in self.evictable_host_leaves:
            self.evictable_host_leaves.remove(node)
        return
```

```
# 遍历子节点：若有任一子节点在 host 仍有备份数据 (backupid 为 True) ,
# 则当前节点不能作为可驱逐的 host 叶子 (因为需要保留其子节点在 host 的数据)
for child in node.children.values():
    if child.backupid: # 修复：原为 child.evicted, 错误地将仅从 GPU 驱逐
        # 但仍在 host 有备份的子节点视为“可忽略”, 导致父节点被错误加入
        if node in self.evictable_host_leaves:
            self.evictable_host_leaves.remove(node)
    return

# 所有子节点均无 host 备份, 则当前节点可加入可驱逐 host 叶子集合
if node not in self.evictable_host_leaves:
    self.evictable_host_leaves.add(node)
```

评论区精华

Reviewer @hzh0425 在评论中提供了详细的失败链路分析, 补充了 `backupid` 为 `True` 但 `evicted` 为 `False` 的具体场景, 确认了修复的正确性。两位 reviewer 均批准了 PR。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：变更仅为单行条件替换, 逻辑正确性有 reviewer 确认和 Issue 场景支撑。回归风险在于若其他逻辑依赖 `_update_host_leaf_status` 的旧行为, 但此方法仅在 `dec_lock_ref` 和 `evict` 中调用, 且用途明确, 回归可能性小。
- 影响：直接修复 HiCache `write_back` 策略下的崩溃 bug, 影响使用分层缓存且策略为 `write_back` 的用户。不影响 `write_through` 策略或其他模块。
- 风险标记：单行变更回归风险低

关联脉络

- PR #21125 [HiCache] feat: add draft KV cache backing for L2/L3: 同为 HiCache 功能线, 修改了相同模块的缓存管理逻辑