PR #26870 完整报告

sgl-project/sglang

Make unified tree SWA hicache tests faithful to write-through backup

合并时间：2026-06-01 16:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26870

执行摘要

- 一句话：增强 SWA HiCache 单元测试，模拟父优先写通备份和多节点树场景
- 推荐动作：建议测试工程师和缓存模块开发者精读此 PR，学习如何通过模拟父优先备份路径编写更贴合生产逻辑的单元测试。新增的压力测试可作为参考用例，用于验证类似数据结构。

功能与动机

当前 SWA HiCache 单元测试仅备份单个节点（且自动备份禁用），断言基于单叶树形状，既不验证真实的父优先写通备份，也无法容忍序列跨多个树节点。为使测试真实反映生产写通备份行为，需要将备份逻辑改为整个路径祖先优先，并添加多节点深度树压力测试。

实现拆解

1. 重构备份模拟函数 _simulate_backup 和真实备份函数 _backup_node：将原单节点备份改为遍历从目标节点到根节点的完整路径，按祖先优先（父节点先于子节点）顺序执行备份，确保写通语义。同时 _simulate_backup 不再只备份传入节点，而是沿路径依次设置 host_value。
2. 调整受影响的现有测试用例：
   - test_aux_evict_full_locked_leaf_tombstones_aux_only：将插入序列长度从 2 改为 1，使叶子保持在一个 SWA 窗口内，避免叶容量分裂。
   - test_hicache_load_back_restores_data：改用 match_prefix 收集重新加载的前缀值，而非仅断言叶值。
   - test_hicache_partial_match_splits_evicted_backed_up_node：改为通过 last_host_node 定位主机前缀，并从路径键重建前缀 / 后缀断言，消除对固定父 / 子形状的依赖。
3. 新增压力测试 test_swa_deep_tree_backup_evict_loadback_stress：构建多节点深度 SWA 树，依次执行备份 → 回收 → 加载回设备，全程调用 sanity_check 验证树结构不变，确保真实写通场景下缓存一致性。

关键文件：

- test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py（模块 缓存测试；类别 test；类型 test-coverage；符号 test_swa_deep_tree_backup_evict_loadback_stress, insert_swa, _simulate_backup, _backup_node）：唯一变更文件，重构了备份模拟和真实备份逻辑，使其忠于写通顺序，并添加了深度树压力测试。

关键符号：_simulate_backup, _backup_node, test_swa_deep_tree_backup_evict_loadback_stress

关键源码片段

test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py

唯一变更文件，重构了备份模拟和真实备份逻辑，使其忠于写通顺序，并添加了深度树压力测试。

```
# 核心变更 1：_simulate_backup 改为备份整个 root->node 路径（父优先）
def _simulate_backup(self, tree, node):
    """Simulate D->H backup over the whole root->node path (parent-first)."""
    chain = []
    cur = node
    # 从当前节点向根遍历，收集路径上的所有节点
    while cur is not tree.root_node:
        chain.append(cur)
        cur = cur.parent
    # 按祖先优先（reversed chain）顺序备份每个组件
    for ancestor in reversed(chain):
        for ct in (ComponentType.FULL, ComponentType.MAMBA, ComponentType.SWA):
            if ct not in self.cfg.components:
                continue
            cd = ancestor.component_data[ct]
            # 仅在 device 上有值且 host 上尚无备份时执行克隆
            if cd.value is not None and cd.host_value is None:
                cd.host_value = cd.value.clone()

# 核心变更 2：_backup_node 同样改为整路径父优先写通备份
def _backup_node(self, tree, node):
    # Parent-first backup over the whole path: one insert can span several
    # nodes, so a single-node backup would leave an unbacked ancestor.
    chain = []
    cur = node
    while cur is not tree.root_node:
        chain.append(cur)
        cur = cur.parent
    backed_up = 0
    # 从根向叶依次备份，跳过已备份节点
    for ancestor in reversed(chain):
        if ancestor.backuped:
            continue
        backed_up = tree.write_backup(ancestor, write_back=True)
        self.assertGreater(backed_up, 0)
    tree.writing_check(write_back=True)
    # 确保叶节点最终已被备份
    self.assertTrue(node.backuped)
    return backed_up
```

评论区精华

无实质性技术讨论，作者 /ispobock 在 Issue 评论中触发 CI 重跑指令，test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py 测试全部通过。

- CI 重跑确认测试通过 (other): 测试通过，无代码争议。

风险与影响

- 风险：仅修改测试文件，不影响生产代码。风险较低，但需注意：备份模拟逻辑变更可能使旧测试因依赖单节点备份而失败，本次已调整相关断言；新增压力测试覆盖了多节点场景，降低了回归风险。无性能或安全风险。
- 影响：影响范围局限于该单元测试文件，对系统其他模块无影响。提升测试对真实写通行为的模拟准确度，增强缓存模块的可靠性验证。团队应知悉测试行为变化，避免将来误判。
- 风险标记：测试逻辑变更，压力测试新增

关联脉络

- 暂无明显关联 PR