

PR #33777 完整报告

sgl-project/sglang

[HiCache] write_back: reclaim duplicated host copy first under host pressure

合并时间: 2026-08-08 06:59

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/33777>

执行摘要

- 一句话: HiCache write_back 优先回收重复 Host 副本
- 推荐动作: 值得精读。重点关注惰性跟踪 + 实时复查 + digest 跨 TP 校验的设计模式, 以及两轮扫描对“即将被 demote 的 evictable D-leaf”的保留策略。建议后续补充针对 _reclaim_full_host_duplicates 两轮策略与 digest 匹配的独立单元测试, 以增强回归保障。

功能与动机

PR body 开宗明义: 'This is to maximize the cache capacity benefit of the write-back policy.' 在 write_back 策略下, 设备端已有 Full KV 的节点, 其 Host 副本是完全冗余的; 优先回收它们可以在不损失任何可用缓存容量的前提下缓解 Host 压力, 从而最大化写回策略的有效容量收益。这正是本次变更的核心动机。

实现拆解

实现按以下步骤拆解:

1. 树节点状态扩展: UnifiedTreeNode 新增 load_back_pending_id 字段, 用于标记正在进行的 H->D load-back 读取该节点 host 槽位; UnifiedTreeCore 初始化与 reset 中新增 full_host_duplicates (插入有序字典, 保证跨 TP 受害选择确定性) 和 write_back_duplicate_reclaim_digest (滚动哈希, 用于跨 TP 一致性校验)。
2. 双份状态跟踪: 新增 _update_duplicate_tracking 和 _is_settled_full_host_duplicate, 在 backup ack、节点分裂 (_split_node)、恢复被驱逐节点 (_unevict_node_on_insert) 等“双份诞生点”注册; 注销是惰性的, 依赖 _can_reclaim_full_host_duplicate 在回收时实时复查。节点删除 (_remove_leaf_from_parent) 时同步清理, 避免幽灵条目。
3. 回收路径改造: drive_host_eviction 在 write_back 且目标为 BASE_COMPONENT_TYPE (FULL) 时, 先调用 _reclaim_full_host_duplicates 回收双份副本, 再走常规 LRU 驱逐。回收采用两轮扫描: 第一轮跳过 evictable_device_leaves 中的节点 (其设备层即将被 demote, 主机层可能是唯一副本), 第二轮才允许回收; 每回收一个节点, 更新 write_back_duplicate_reclaim_digest。_release_full_host_duplicate 只释放 FULL 的 host 层, aux 组件的 host 切片仍归各自 pool 的 LRU 管理。
4. DMA 保护与 ack 解除: commit_load_back 时为所有涉及 host 槽位读取的节点打上 load_back_pending_id; loading_check 在 ack 后通过新增接口 finish_load_back 清除锚点根路径上的标记。节点分裂时该标记会继承给拆分出的新片段, 保证 ack walk 不受影响。

5. 跨 TP 一致性校验: `_sync_hicache_ready_counts` 与 `loading_check` 将 `[digest, -digest]` 追加到原有的 `all_reduce` 张量中, 利用 MIN 归约后在所有 rank 上断言 `count_values[-2] == -count_values[-1]`, 一旦回收受害顺序分歧立即失败, 机制与既有 `ready-count` 同步共用一次通信。
6. 测试配套: `test_unified_radix_cache_unittest.py` 的 `_simulate_backup` 增加 `_update_duplicate_tracking` 调用, 使现有备份模拟与真实 `ack` 行为一致, 从而让既有单元测试覆盖新跟踪逻辑。未新增针对回收策略的独立断言。

关键文件:

- `python/sglang/srt/mem_cache/unified_cache/unified_tree_core.py` (模块 缓存核心; 类别 `source`; 类型 `core-logic`; 符号 `_update_duplicate_tracking`, `_is_settled_full_host_duplicate`, `_reclaim_full_host_duplicates`, `_can_reclaim_full_host_duplicate`) : 核心实现所在: 新增双份跟踪、两轮回收、DMA 保护与 `digest` 更新等全部关键逻辑。
- `python/sglang/srt/mem_cache/unified_cache/unified_tree_core_interface.py` (模块 缓存接口; 类别 `source`; 类型 `core-logic`; 符号 `finish_load_back`, `write_back_duplicate_reclaim_digest`) : 接口层新增 `finish_load_back` 抽象方法和 `write_back_duplicate_reclaim_digest` 类属性, 是所有 `TreeCore` 实现必须遵守的契约。
- `python/sglang/srt/mem_cache/unified_radix_cache.py` (模块 缓存编排; 类别 `source`; 类型 `core-logic`; 符号 `_sync_hicache_ready_counts`, `loading_check`) : 编排层将 `digest` 校验 piggyback 到既有 `all_reduce`, 并在 `loading_check` 的 `ack` 路径调用 `finish_load_back` 解除 DMA 保护。
- `test/registered/unit/mem_cache/test_unified_radix_cache_unittest.py` (模块 单元测试; 类别 `test`; 类型 `test-coverage`) : 测试辅助函数 `_simulate_backup` 增加 `duplicate_tracking` 注册调用, 使模拟备份与真实 `ack` 行为一致。

关键符号: `_update_duplicate_tracking`, `_is_settled_full_host_duplicate`, `_reclaim_full_host_duplicates`, `_can_reclaim_full_host_duplicate`, `_release_full_host_duplicate`, `finish_load_back`, `drive_host_eviction`, `loading_check`, `_sync_hicache_ready_counts`

评论区精华

该 PR 没有 review 讨论线程, reviewer `stmatengss` 直接 APPROVED 且未附文字意见。唯一的沟通是维护者 `hanming-lu` 在合并前的评论: 'failed tests are unrelated, merging', 表明 `pr-test-extra` 的 CI 失败被判定与本次变更无关, 直接放行合并。整体上, 这个功能变更的评审过程比较简洁, 核心设计权衡 (两轮扫描、惰性跟踪、`digest` 校验) 未在评论区展开讨论。

- CI 失败判定与合并 (other): 维护者判定失败测试与本次变更无关, 直接合并; 无更多技术争论。

风险与影响

- 风险:

1. 回收路径性能: `_reclaim_full_host_duplicates` 在 `drive_host_eviction` 同步路径内遍历 `full_host_duplicates`, Host 压力期间可能引入额外 CPU 开销, 但仅在 `write_back` 且 FULL 组件压力时触发, 量级可控。
2. 跨 TP 一致性断言: `digest` 要求所有 TP rank 对回收受害者的选择顺序完全一致; 任何因 `ack` 完成顺序、节点创建顺序差异导致的不一致都会直接触发 `AssertionError` 终止进程。这是有意的强检查, 但真实环境若存在 rank 间事件顺序抖动, 可能暴露为偶发崩溃。
3. 惰性跟踪的遗漏风险: `full_host_duplicates` 依赖各事件点调用 `_update_duplicate_tracking` 或 `_remove_leaf_from_parent` 清理; 若未来新增修改节点状态的路径遗漏调用, 可能出现漏回收或误回收。当前 `_split_node` 只对 `new_node` 调 `_update_duplicate_tracking`, `child` 是否始终保持在正确状态需要持续关注。
4. 测试覆盖不足: 测试仅适配 2 行, 没有新增直接断言 `duplicate reclaim` 行为、`digest` 计算、DMA 保护等核心逻辑的针对性用例, 回归保障有限。
 - 影响: 影响范围: 启用 HiCache 且写回策略为 `write_back` 的多卡 (TP/PP) 部署, 在 Host 内存压力下的缓存效率与命中率。对用户而言, 优先回收冗余副本能提升有效缓存容量, 减少因 Host 内存不足导致的全量换出, 理论上改善长上下文复用性能。对团队而言, 新增跨 TP 一致性强校验机制为后续 Rust 端口保持等价行为提供契约; 接口层新增抽象方法要求所有 `TreeCore` 实现同步实现, 带来一定的维护成本。
 - 风险标记: `host` 内存回收核心路径变更, 跨 TP 一致性强校验, 缺少新增针对性测试, 惰性跟踪依赖实时复查

关联脉络

- PR #33468 Remove the HiMambaRadixTree that is no longer in use: 同属 `UnifiedRadixCache / mem_cache` 重构演进, 移除旧 Mamba 专用缓存树, 统一由 `UnifiedRadixCache` 接管, 与本 PR 改动的 `unified_tree_core.py`、`unified_radix_cache.py` 在同一子系统。
- PR #33925 config: route DCP topology reads through `get_parallel()`: 围绕配置种子与运行时一致性问题, 强调跨 rank 一致性; 本 PR 的 `digest all_reduce` 校验是同一主题在 Host 缓存回收路径上的延伸。
- PR #32556 Autotune flashinfer extend buckets at warmup: 同为 HiCache 相关的运行时优化, 涉及 `flashinfer_autotune` 与 `base_runner` 的预热路径, 与本次 `write_back` 回收策略共同影响缓存容量与性能。