PR #35933 完整报告

sgl-project/sglang

[HiCache] Clamp tombstoned SWA locs in UnifiedSWAKVPool translation

合并时间：2026-08-22 14:08

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35933

执行摘要

- 一句话：SWA 池 loc 翻译钳制墓碑负值，防 cuda-graph 非法访问
- 推荐动作：值得快速精读（约 15 分钟）。这是一个典型的 KV 池边界防御 bugfix，展示了 tombstone sentinel 的标准处理模式：收敛到保留的 padding sink 槽位而非让负值逃逸。值得关注三个点：① 为什么 slot 0 是安全的"垃圾场"；② 为什么负 loc 在已捕获的 cuda-graph 重放下是非法访问而非普通越界；③ 缺少单测的后续补强方向（可为 translate_loc_from_full_to_swa 的墓碑输入构造最小回归测试）。

功能与动机

PR body 明确指出：translate_loc_from_full_to_swa 返回了裸的虚拟到物理查找结果，墓碑化 v2p 条目（-1）会以负 slot id 逃逸到调用方；当 page_size > 1 时页计算会将其变为 [-ps, -1] 范围内的值。无论哪种情况，调用方都会用负 loc 索引 SWA KV buffer，在已捕获的 cuda-graph 重放下属于非法访问。修复目标是在 int32 转换前将翻译后的 loc 钳制到 0，匹配 MultiEndedAllocator.translate_kv_loc 中已有的墓碑安全钳制。

实现拆解

1. 定位缺陷：在 python/sglang/srt/mem_cache/unified_memory_pool.py 的 UnifiedSWAKVPool.translate_loc_from_full_to_swa 中，原实现直接把 v2p 查找结果返回；当 v2p 条目为墓碑值 -1 时，ps == 1 路径直接返回 -1，ps > 1 路径下 -1 * ps + offsets 落在 [-ps, -1]，调用方用负值索引 SWA KV buffer，在已捕获的 cuda-graph 重放下构成非法访问。
2. 重构翻译路径：将 ps == 1 与 ps > 1 两个分支的计算结果统一存入局部变量 swa_locs，在 to(torch.int32) 之前统一执行 swa_locs.clamp(min=0)，把墓碑化条目收敛到槽位 0。
3. 对齐既有约定：slot 0 是 MultiEndedAllocator 中保留的 padding sink（与 MultiEndedAllocator.translate_kv_loc 的既有钳制保持一致），墓碑化读写会无害地落到该槽位；正常条目的物理 loc 本就是非负数，clamp(min=0) 对它们完全是无操作。
4. 配套与验证：无签名与调用方改动，无新增测试文件；改动来自内部 commit 的验证（hierarchical-cache SWA serving 环境），本地未重跑 GPU/cuda-graph 复现，依赖 CI 门禁兜底。

关键文件：

- python/sglang/srt/mem_cache/unified_memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 UnifiedSWAKVPool, translate_loc_from_full_to_swa）：唯一变更文件。UnifiedSWAKVPool.translate_loc_from_full_to_swa 在 int32 转换前统一对翻译结果执行 clamp(min=0)，消除墓碑化 v2p 条目（-1）作为负 loc 逃逸到调用方的问题，与 MultiEndedAllocator.translate_kv_loc 的既有钳制对齐。

关键符号：translate_loc_from_full_to_swa

关键源码片段

python/sglang/srt/mem_cache/unified_memory_pool.py

唯一变更文件。UnifiedSWAKVPool.translate_loc_from_full_to_swa 在 int32 转换前统一对翻译结果执行 clamp(min=0)，消除墓碑化 v2p 条目（-1）作为负 loc 逃逸到调用方的问题，与 MultiEndedAllocator.translate_kv_loc 的既有钳制对齐。

```
def translate_loc_from_full_to_swa(self, kv_indices: torch.Tensor):
    """Virtual token ids -> swa-physical token ids (int32)."""
    assert self._swa_allocator is not None, (
        "UnifiedSWAKVPool.translate_loc_from_full_to_swa called before "
        "attach_allocators"
    )
    ps = self._swa_allocator.page_size

    # 墓碑安全钳制，与 MultiEndedAllocator.translate_kv_loc 保持一致：
    # v2p 表中被墓碑化的条目值为 -1，若直接返回会成为负 slot id，
    # 在已捕获的 cuda-graph 重放下会非法访问 SWA KV buffer 前端。
    if ps == 1:
        swa_locs = self._swa_allocator.virtual_to_physical[kv_indices]
    else:
        virt_pages = kv_indices // ps
        offsets = kv_indices % ps
        swa_phys_pages = self._swa_allocator.virtual_to_physical[virt_pages]
        # 墓碑化页面：-1 * ps + offset 会落在 [-ps, -1] 区间。
        swa_locs = swa_phys_pages * ps + offsets

    # 钳到 0：slot 0 是保留的 padding sink，墓碑化读写会无害地落在这里；
    # 正常条目的物理 loc 必然非负，clamp 对它们是完全的 no-op。
    return swa_locs.clamp(min=0).to(torch.int32)
```

评论区精华

该 PR 没有外部 reviewer 参与，唯一的讨论出现在 issue 评论区，由作者 @xiezhq-hermann 自己发起：

- 关于 base-a-test-cpu (9) 的失败：作者指出 test/registered/unit/test_legacy_global_ratchet.py 报错 get_global_server_args call-sites grew: 2 > baseline 1，这是 main 上 #35794 引入 python/sglang/srt/models/granite.py 调用点导致的 pre-existing 失败，与本 PR 无关；本 PR 只触碰 UnifiedSWAKVPool.translate_loc_from_full_to_swa，没有新

增 get_global_server_args 调用点。

- 关于早期 pr-gate 失败：作者解释是标签竞争——gate job 在 /tag-and-rerun-ci 应用 run-ci 标签之前就读了 labels，重新触发后已通过。
- CI ratchet 失败与 pre-existing 问题归属 (testing): 失败非本 PR 引入，需在 main 上迁移调用点到 runtime_context.get_server_args() 或上调基线后转绿；本 PR 重新触发 CI 后已通过。

风险与影响

- 风险：
  1. 缺少直接测试覆盖：本次改动没有新增针对墓碑化翻译路径的单元测试，回归验证完全依赖 CI，tombstone 分支的正确性缺乏本地断言。
  2. cuda-graph 复现未本地执行：PR body 说明 GPU/cuda-graph 复现来自内部 commit，本 checkout 没有合适环境重跑，非法访问路径是否真正消除仍需 CI 或后续手工验证。
  3. clamp 的语义边界：墓碑条目钳到 slot 0 后，读取到的是 padding sink 里的数据（安全但不正确）；如果上层逻辑将来依赖“墓碑可见性”（例如通过负值识别无效条目），需要注意该约定已被屏蔽。当前调用方场景下这是可接受的 trade-off。
  4. 核心路径变更：该文件属于 KV cache 内存池主路径，虽然逻辑上对非墓碑条目是 no-op，但 clamp 引入了一次额外的张量操作，对极端性能敏感路径需留意（预期可忽略）。 - 影响：影响范围集中在 HiCache 层级缓存下 SWA 池的 loc 翻译路径，仅涉及 python/sglang/srt/mem_cache/unified_memory_pool.py 中一个方法。对用户而言，修复了 hierarchical-cache SWA serving 在 cuda-graph 重放场景下因负 loc 触发的非法访问（潜在 CUDA 错误或内存损坏）。对系统而言，行为向后兼容：非墓碑路径输出与之前完全一致，墓碑路径从“非法访问”变为“安全落在 padding sink”。对团队而言，该改动确立了一种与 MultiEndedAllocator.translate_kv_loc 一致的墓碑处理模式，可作为后续内存池防御性编码的参考范式。 - 风险标记：核心 KV 内存路径变更，缺少直接测试覆盖，cuda-graph 复现依赖 CI

关联脉络

- PR #35906 config: project the config bags from the resolution result: 同属 HiCache 功能线，改动涉及 hybrid_pool_assembler.py，与本次 hybrid-pool 内存管理主题相关。
- PR #36005 [Mamba] fix mamba index h unexpected assertion for dcp: 同为 serving 运行时对越界 / 非法状态索引的防御性修复，体现对 KV 状态管理边界的系统性加固。