PR #32016 完整报告

sgl-project/sglang

[Fix] Evict only the KV shortfall in evict_from_tree_cache

合并时间：2026-07-23 03:41

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/32016

执行摘要

- 一句话：修复 Standard allocator 过度驱逐问题
- 推荐动作：值得快速合并。建议后续补充针对 evict_from_tree_cache Standard allocator 分支的单元测试，验证驱逐量正确性。

功能与动机

Standard allocator 分支在 evict_from_tree_cache 中驱逐了完整的 num_tokens 而非仅驱逐短缺量 num_tokens - available_size，而 SWA 分支已经正确计算了短缺量。每次内存压力事件会额外破坏最多 available_size 个缓存前缀，在 MambaRadixCache 上损失更大（不可增量恢复）。见 PR 描述。

实现拆解

1. python/sglang/srt/mem_cache/common.py - evict_from_tree_cache 函数：在 else 分支（Standard allocator）中，将 tree_cache.evict(EvictParams(num_tokens=num_tokens)) 改为 tree_cache.evict(EvictParams(num_tokens=num_tokens - available_size))。先读取 available_size，若不足 num_tokens，则只驱逐短缺量。
2. python/sglang/srt/managers/schedule_batch.py - check_decode_mem 方法：添加了 docstring 说明仅驱逐 shortfall，无逻辑变更。

关键文件：

- python/sglang/srt/mem_cache/common.py（模块 缓存层；类别 source；类型 core-logic；符号 evict_from_tree_cache）：核心 bugfix 位置：修正 evict_from_tree_cache Standard allocator 分支的驱逐量计算。
- python/sglang/srt/managers/schedule_batch.py（模块 调度器；类别 source；类型 documentation；符号 check_decode_mem）：调用 evict_from_tree_cache 的函数，添加了 docstring 说明。

关键符号：evict_from_tree_cache, check_decode_mem

关键源码片段

python/sglang/srt/mem_cache/common.py

核心 bugfix 位置：修正 evict_from_tree_cache Standard allocator 分支的驱逐量计算。

```
# python/sglang/srt/mem_cache/common.py
def evict_from_tree_cache(tree_cache: BasePrefixCache | None, num_tokens: int):
    if tree_cache is None:
        return
    if tree_cache.is_chunk_cache():
        return

    allocator = tree_cache.token_to_kv_pool_allocator

    if isinstance(allocator, SWATokenToKVPoolAllocator):
        # Hybrid allocator: evict only the shortfall for each pool
        full_available_size = allocator.full_available_size()
        swa_available_size = allocator.swa_available_size()
        if full_available_size < num_tokens or swa_available_size < num_tokens:
            full_num_tokens = max(0, num_tokens - full_available_size)
            swa_num_tokens = max(0, num_tokens - swa_available_size)
            tree_cache.evict(
                EvictParams(num_tokens=full_num_tokens, swa_num_tokens=swa_num_tokens)
            )
    else:
        # Standard allocator: evict only the shortfall (mirrors the SWA arm)
        available_size = allocator.available_size()
        if available_size < num_tokens:
            # 修正前：驱逐完整 num_tokens 导致过度驱逐
            tree_cache.evict(EvictParams(num_tokens=num_tokens - available_size))
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：核心变更仅 1 行逻辑修正，将驱逐量从 num_tokens 改为 num_tokens - available_size，与 SWA 分支保持一致。未引入新代码路径或配置变更。但测试覆盖不足：未包含专门针对 Standard allocator 驱逐量的回归测试。
- 影响：影响范围限于 Standard allocator 路径下的内存驱逐行为。减少不必要的缓存前缀驱逐，提高 KV cache 命中率和 decode 性能，尤其对 MambaRadixCache 和长序列场景有正面影响。不影响 SWA allocator、chunk cache 或无 tree cache 场景。
- 风险标记：缺少测试覆盖

关联脉络

- PR #32029 [Fix] Unify pinned host pool release on graceful shutdown: 也在 scheduler 和 memory pool 目录下修复内存管理 bug，同一维护者 hnyls2002。