# PR #25805 完整报告

sgl-project/sglang

Fix SWA double-free in disagg decode with MTP speculation

合并时间：2026-05-22 15:28

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25805

## 执行摘要

- 一句话：修复 SWA double-free 在 disagg decode 中的 bug
- 推荐动作：值得精读，尤其是关注 disagg 和推测解码稳定性的团队。修复思路清晰，可作为类似残留引用问题的参考模式。

## 功能与动机

在 disagg decode + EAGLE MTP 推测解码场景下，集群运行约 2000 个请求后频繁触发断言失败 AssertionError: swa_attn_allocator.available_size() <= swa_attn_allocator.size，表明 SWA 内存分配器状态不一致，出现 double-free。经初步分析（附链接），原因是回收的 full KV 页面（来自 alloc_extend，1:1 SWA 映射）中的 full_to_swa_index_mapping 在进入 alloc_extend_swa_tail 后仍保留旧请求的映射，导致后续 eviction 或 free 时错误地释放了其他请求的 SWA 页面。

## 实现拆解

变更仅涉及 python/sglang/srt/mem_cache/swa_memory_pool.py 中的 alloc_extend_swa_tail 方法，在映射建立后增加一段清理逻辑： 1. 检测 tail 长度：判断 swa_tail_len 是否小于 extend_num_tokens，即是否存在非 tail 的 full KV 页面（1:1 SWA 映射）。 2. 清理陈旧映射：当存在非 tail 部分时，将 alloc_full_indices[:-swa_tail_len] 对应的 full_to_swa_index_mapping 置零，消除旧请求遗留的 SWA 索引映射。 该修复确保每次 alloc_extend_swa_tail 调用后，非 tail 页面的映射是干净的，不会在后续 eviction/free 时错误释放其他请求的 SWA 页面。

关键文件：

- python/sglang/srt/mem_cache/swa_memory_pool.py（模块 缓存层；类别 source；类型 core-logic；符号 alloc_extend_swa_tail）：核心修复文件，在 alloc_extend_swa_tail 中增加对非 tail 部分陈旧映射的清理逻辑，是解决 double-free 的唯一变更。

关键符号：alloc_extend_swa_tail

## 关键源码片段

### python/sglang/srt/mem_cache/swa_memory_pool.py

核心修复文件，在 alloc_extend_swa_tail 中增加对非 tail 部分陈旧映射的清理逻辑，是解决 double-free 的唯一变更。

```
# python/sglang/srt/mem_cache/swa_memory_pool.py
# 在 alloc_extend_swa_tail 方法中，分配并建立 full-to-SWA 映射后，
# 增加对非 tail 部分（1:1 SWA 映射）陈旧引用清理的逻辑。

def alloc_extend_swa_tail(self, ...):
    # ... 前面的分配逻辑保持不变 ...

    # 建立 tail 部分（SWA >1:1）的映射
    self.full_to_swa_index_mapping[alloc_full_indices[-swa_tail_len:]] = (
        alloc_swa_indices
    )

    # 修复：如果存在非 tail 的 full 页面（1:1 SWA 映射），
    # 它们可能来自之前请求的 alloc_extend，残留了旧映射。
    # 这些页面被回收后，旧映射会导致后续 eviction/free
    # 错误释放其他请求的 SWA 页面，造成 double-free。
    # 通过将映射置零来消除陈旧引用。
    if swa_tail_len < extend_num_tokens:
        self.full_to_swa_index_mapping[alloc_full_indices[:-swa_tail_len]] = 0

    return alloc_full_indices
```

## 评论区精华

讨论较少，主要关注合并速度。nvpohanh 建议尽快合并以避免 double-free 问题。ispobock 引用了相关 PR #24857、#25901、#25385，表明此问题有深层关联。

- 暂无高价值评论线程

## 风险与影响

- 风险：风险较低。变更仅为在映射赋值后增加条件置零语句，不影响正常情况下 tail 部分的映射。若 full_to_swa_index_mapping 置零后，对应页面的 SWA 映射在后续 decode 阶段需要重建，但 decode 路径（alloc_decode）会覆盖所有索引，因此不会遗漏。仅 2 行增加，逻辑清晰，且已在 10240 请求的长稳测试中通过。
- 影响：直接影响 disagg decode + MTP 推测解码场景的稳定性，消除断言失败和潜在的内存损坏。对非 disagg 或非推测解码场景无影响，因为 alloc_extend_swa_tail 仅在 tail 分配时调用。修复后集群可长期稳定运行。
- 风险标记：核心路径变更

## 关联脉络

- PR #24857 未获取到标题，但 ispobock 引用为相关：ispobock 在评论中引用，表明与 SWA 映射或内存管理相关。

- PR #25901 未获取到标题，但 ispobock 引用为相关：ispobock 在评论中引用，表明与 SWA 映射或内存管理相关。
- PR #25385 未获取到标题，但 ispobock 引用为相关：ispobock 在评论中引用，表明与 SWA 映射或内存管理相关。