PR #22470 完整报告

sgl-project/sglang

Fix SWA eviction boundary and page-align chunked prefill

合并时间：2026-04-10 13:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22470

执行摘要

本 PR 修复了 SWA（推测解码中的滑动窗口注意力）驱逐边界计算错误，当页面大小大于滑动窗口大小时，防止因过度驱逐导致的 radix 树插入故障、负使用和潜在双重释放。通过预防性调整驱逐公式和防御性插入检查，增强缓存系统的稳定性与多轮重用能力。

功能与动机

该修复源于一个特定场景的 bug：在 page_size > sliding_window_size 时，_evict_swa 函数可能将驱逐前沿精确推到 page_floor(seq_len)，使得所有待插入令牌完全被驱逐（case 3）。如 PR body 所述，这导致 swa_evictable_size_ 膨胀、负使用和后续树驱逐时的潜在双重释放，威胁系统可靠性。

实现拆解

主要改动集中在三个文件：

- python/sglang/srt/managers/schedule_batch.py：修改 _evict_swa 函数，在计算 new_swa_evicted_seqlen 时减去额外 page_size，确保驱逐前沿永不触及插入边界。注释说明此举保留至少一页非驱逐 KV 用于缓存重用。
- python/sglang/srt/mem_cache/swa_radix_cache.py：在 _insert_helper 函数中添加对 case 3（swa_evicted_seqlen == total_length）的早期返回逻辑，释放令牌值而不创建节点，作为防御性保护。注释关联了与 _evict_swa 的协同关系。
- test/registered/unit/mem_cache/test_swa_eviction_boundary.py：新增单元测试，模拟真实树、分配器和池，覆盖页面大小与滑动窗口的各种组合，验证修复正确性。

评论区精华

Review 中无具体讨论，但 commit 历史揭示了演进过程：

> 例如，commit 'clarify defensive relationship in comments' 强调了预防性和防御性修复的协同作用，避免仅依赖单一防护。

测试多次重写以完善覆盖，从初始边界测试扩展至全场景验证，确保逻辑健壮性。

风险与影响

- 技术风险：修改了核心缓存驱逐和插入路径，但变更范围小且专注；新增测试降低了回归风险。潜在影响仅限于 *page_size* > sliding_window_size 的 SWA 场景。
- 影响分析：修复避免系统崩溃，提升多轮对话下的缓存效率；对终端用户透明，但可能改善使用 SWA 的推理任务性能。

关联脉络

从仓库近期历史 PR 看，本 PR 与以下相关：

- PR #22458：修复 NCCL AllGather hanging issue，同样涉及推测解码组件，共享 'speculative-decoding' 标签。
- PR #22241：修复 MultiLayerEagleWorkerV2 返回 logprobs 问题，可能交互 SWA 缓存逻辑。 这些 PR 反映了项目在推测解码和调度模块持续优化稳定性，本 PR 是这一趋势中的具体 bugfix。