PR #25419 完整报告

sgl-project/sglang

Port SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN from deepseek_v4_dev

合并时间：2026-05-16 08:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25419

执行摘要

- 一句话：修复 SWA 逐出边界 env 变量未生效 bug
- 推荐动作：值得精读的场景：关注 SWA cache 逐出策略的开发者；希望了解多分支间 env 变量移植实践的读者。推荐与 #24857 和 swa_radix_cache.py 中 _insert_helper 的 case 3 对照阅读。

功能与动机

修复 issue #24857 中报告的高并发 disagg decode 下 SWA tail-only preallocation 导致的 AssertionError: swa_attn_allocator.available_size() <= swa_attn_allocator.size。原因是 deepseek_v4_dev 中已存在的 env 变量 SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN 在 #24857 移植 SWA tail preallocation 优化到 main 时未被包含，导致 main 分支忽略了该 flag，使用了更保守的逐出阈值。

实现拆解

1. 注册新环境变量：在 python/sglang/srt/environ.py 的 Envs 类中新增 SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN = EnvBool(False)，默认关闭。
2. 修改逐出逻辑：在 python/sglang/srt/managers/schedule_batch.py 的 _evict_swa 方法中，根据环境变量值计算逐出阈值：若启用则使用 pre_len - sliding_window_size（不保留边距），否则使用 pre_len - sliding_window_size - page_size（保留一个 page 边距）。
3. 该改动仅影响 SWA 逐出逻辑，不对其他模块产生副作用。test 覆盖：本次变更未添加新的单元测试，依靠现有 CI 覆盖。

关键文件：

- python/sglang/srt/managers/schedule_batch.py（模块 调度器；类别 source；类型 core-logic；符号 _evict_swa）：核心逻辑变更：在 _evict_swa 方法中根据环境变量动态计算逐出阈值。
- python/sglang/srt/environ.py（模块 配置；类别 source；类型 configuration；符号 Envs）：新增环境变量 SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN 的定义。

关键符号：_evict_swa

关键源码片段

python/sglang/srt/managers/schedule_batch.py

核心逻辑变更：在 _evict_swa 方法中根据环境变量动态计算逐出阈值。

```
  def _evict_swa(self, req: Req, pre_len: int):
      assert self.tree_cache.supports_swa(), "prefix cache must support swa"
      sliding_window_size = self.tree_cache.sliding_window_size

      # 截断 SWA 逐出边界，避免逐出仍在滑动窗口内的 token
      assert (
          req.cache_protected_len % self.tree_cache.page_size == 0
      ), "cache_protected_len must be page aligned"
      req.swa_evicted_seqlen = max(req.swa_evicted_seqlen, req.cache_protected_len)

      # 根据环境变量决定是否保留 page_size 边距
      # 该边距用于保护 radix tree 插入点，防止叶节点被 tombstone
      # 导致多轮对话 SWA cache 泄漏
      if envs.SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN.get():
          # 激进：直接逐出到滑动窗口边界
          evict_threshold = pre_len - sliding_window_size
      else:
          # 保守：保留一个 page 的边距（默认行为）
          evict_threshold = pre_len - sliding_window_size - self.tree_cache.page_size

      new_swa_evicted_seqlen = max(
          req.swa_evicted_seqlen,
          evict_threshold,
      )

      # 确保逐出长度 page 对齐
      if self.tree_cache.page_size > 1:
          new_swa_evicted_seqlen = (
              new_swa_evicted_seqlen // self.tree_cache.page_size
          ) * self.tree_cache.page_size

      if new_swa_evicted_seqlen > req.swa_evicted_seqlen:
          free_slots = self.req_to_token_pool.req_to_token[
              req.req_pool_idx, req.swa_evicted_seqlen : new_swa_evicted_seqlen
          ]
          self.token_to_kv_pool_allocator.free_swa(free_slots)
          req.swa_evicted_seqlen = new_swa_evicted_seqlen
```

评论区精华

无 review 讨论。该 PR 只有一个 commit，获得了一次 Approved review。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。该变更仅新增一个环境变量注册和对应的一条 if-else 分支，默认行为（未设置变量时）与之前完全一致。当显式设置 SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN=1 时，逐出边界更激进，可能影响多轮对话中 SWA *cache* 的复用，但这本就是预期的优化行为。
- 影响：影响范围有限。仅影响设置了 SGLANG_OPT_SWA_EVICT_DROP_PAGE_MARGIN=1 的部署（如 InferenceX CI），修复了这些场景下的 assertion 崩溃。对默认部署无影响。
- 风险标记：默认行为不变，env 变量开关，无新增测试覆盖

关联脉络

- PR #24857 Optimize SWA memory preallocation for disaggregated decode: 该 PR 引入了 SWA tail preallocation 优化，但遗漏了此 env 变量的移植