

PR #34823 完整报告

sgl-project/sglang

Skip oow slot freeing under eagle

合并时间: 2026-08-14 17:13

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34823>

执行摘要

- 一句话: EAGLE 下跳过 OOW slot 释放, 修复 SWA 混合模型崩溃
- 推荐动作: 值得精读。该 PR 展示了一个由默认开关翻转引发的跨模块单位不匹配问题, 修复手法简洁 (条件跳过), 但根因分析透彻, 适合作为“默认开关风险”的案例。关注点: `free_out_of_window_slots` 与 `tree_core.is_eagle` 的交互、EAGLE bigram 索引与 raw token 的换算关系, 以及后续是否会补充真正的单位转换修复。

功能与动机

PR body 指出 #34653 翻转 `SGLANG_OPT_UNIFIED_CACHE_FREE_OUT_OF_WINDOW_SLOT` 为 `True` 后, 在混合 SWA 模型 + EAGLE 推测解码下会触发 pool 不变式检查并杀死每个 TP rank: `pool memory leak detected!`。根因是 `split_pos` 单位混用 (raw token vs bigram key), 且影响范围超出开关本身, 因为 `registry.py` 无条件将混合 SWA 模型路由到 `UnifiedRadixCache`。

实现拆解

1. 变更入口: `python/sglang/srt/mem_cache/unified_radix_cache.py` 中的 `UnifiedRadixCache.cache_unfinished_req`。
2. 核心逻辑: 在调用 `comp.free_out_of_window_slots(...)` 前增加 `and not self.tree_core.is_eagle` 条件, 使 EAGLE 推测解码路径跳过该优化, 避免 `swa_evicted_seqlen` 与 `prefix_len` 单位不匹配导致的分裂位置为负、槽位被释放但树仍记账为可驱逐的问题。
3. 影响范围: 由于 `tree_core.is_eagle` 等于 `params.is_eagle` and MAMBA not in components, 因此 DSpark、DFlash、非推测解码和纯 SSM 场景仍保留 OOW 优化, 仅约束 EAGLE + 混合 SWA 场景。
4. 配套改动: 本次仅修改源码, 没有新增测试文件, 依赖既有 `test_gpt_oss_120b.py` 等回归测试验证。

关键文件:

- `python/sglang/srt/mem_cache/unified_radix_cache.py` (模块 缓存层; 类别 source; 类型 core-logic; 符号 `cache_unfinished_req`): 核心修复文件, 修改了 `cache_unfinished_req` 中 OOW slot 释放的条件, 是本次变更的唯一文件。

关键符号: `cache_unfinished_req`, `free_out_of_window_slots`

关键源码片段

python/sglang/srt/mem_cache/unified_radix_cache.py

核心修复文件，修改了 `cache_unfinished_req` 中 OOW slot 释放的条件，是本次变更的唯一文件。

```
# python/sglang/srt/mem_cache/unified_radix_cache.py
# 在 cache_unfinished_req 中，计算 effective_cache_len 后调用 free_out_of_window_slots 前
# swa_evicted_seqlen 是 raw-token 长度，但 EAGLE 下 insert key 使用 bigram 索引，
# 直接混用会导致 split_pos 为负，槽位已释放但树仍记账，触发 pool 不变式校验失败
if (
    envs.SGLANG_OPT_UNIFIED_CACHE_FREE_OUT_OF_WINDOW_SLOTS.get()
    and not self.tree_core.is_eagle
):
    # 仅非 EAGLE 场景执行 OOW 释放，避免单位不匹配
    # tree_core.is_eagle 为 params.is_eagle and MAMBA not in components,
    # 因此 DSpark / DFlash / 非推测 / 纯 SSM 路径仍保留该优化
    for comp in self._components_tuple:
        comp.free_out_of_window_slots(
            req, effective_cache_len - 1, insert_params
        )
```

评论区精华

PR 没有实质性的 review 评论，仅有作者发出的 `/rerun-test registered/8-gpu-models/test_gpt_oss_120b.py` 指令和 CI 自动回复。第一次 rerun 在 H200/B200 均通过，第二次 rerun H200 失败但 B200 通过，可能为间歇性问题，未在 PR 内深入讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 修复范围受限：仅对 EAGLE 场景关闭 OOW 释放，但根因（单位不匹配）并未修复，未来若在其他路径触发同样问题仍可能崩溃。
 2. 缺少单元测试：没有直接针对该 bug 的测试，回归依赖 8-GPU 大模型测试，本地难以快速验证，且第二次 rerun 出现 H200 失败，存在偶发风险。
 3. 行为退化：EAGLE + 混合 SWA 模型将暂时失去 OOW 槽位释放的显存优化，可能导致显存占用略有上升。
 4. 条件隐式耦合：`tree_core.is_eagle` 的语义与组件类型（MAMBA）耦合，后续若调整组件组合可能导致条件失效。
 - 影响：直接影响使用 EAGLE 推测解码 + 混合 SWA 模型（如 gpt-oss-120b）的用户，消除了启动即崩溃的阻断性问题。对 DSpark、DFlash、非推测解码用户无影响，仍保留优化。团队需注意 root cause 尚未修复，后续应跟进单位统一方案。影响范围较小但严重度高。
 - 风险标记：核心路径变更，缺少测试覆盖，修复范围受限

关联脉络

- PR #34653 Flip SGLANG_OPT_UNIFIED_CACHE_FREE_OUT_OF_WINDOW_SLOTS to True: PR body 明确指出该 PR 翻转了 OOW slot 释放的默认开关，是本次崩溃的直接触发源。
- PR #34729 Retain SWA down to the last state checkpoint: 修改了同一缓存层（unified_radix_cache.py 及相关组件），涉及 SWA 状态管理，与本 PR 的 OOW 逻辑相关。
- PR #34808 Fix mamba checkpoint depth under dcp: 同样修改了 unified_radix_cache.py 和 mamba 相关组件，属于同一条功能线的持续 bugfix。