PR #26941 完整报告

sgl-project/sglang

Plug mamba_extra_buffer ping-pong slot leaks

合并时间：2026-06-04 21:46

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26941

执行摘要

- 一句话：修复 Mamba 额外缓冲区 ping-pong 槽位泄漏
- 推荐动作：值得精读，特别是流式会话和内存管理的逻辑。save_from_req 和 free_mamba_cache 中的所有权转移与引用清零模式值得在其他资源释放路径中参考。建议添加针对 extra_buffer + overlap_schedule 的集成测试。

功能与动机

在启用 mamba_scheduler_strategy=extra_buffer 和 enable_overlap_schedule 的流式会话基准测试中，关闭第一个会话后调度器的空闲检查会触发泄漏检测并杀死调度器进程。作者诊断出每个请求分配 3 个槽位，但释放不完整导致精确的 Mamba 池差额。参考 PR 正文中的错误堆栈：ValueError: pool memory leak detected! [mamba] total=1889, available=1886，缺失 3 个槽位。

实现拆解

1. python/sglang/srt/session/streaming_session.py - SessionSlot.save_from_req：将 Mamba 所有权转移到槽位后，除了现有的 req.req_pool_idx = None 和 req.mamba_pool_idx = None，还新增清除 req.mamba_ping_pong_track_buffer、req.mamba_next_track_idx、req.mamba_last_track_seqlen、req.mamba_branching_seqlen 为 None。防止后续分配路径（如检查 req.mamba_ping_pong_track_buffer is None）看到过期引用而跳过分配，导致槽位张量被复用而泄漏。
2. python/sglang/srt/mem_cache/memory_pool.py - HybridReqToTokenPool.free_mamba_cache：在池侧释放 ping-pong 缓冲后，新增清除 req.mamba_ping_pong_track_buffer = None 和 req.mamba_next_track_idx = None，匹配现有的 req.mamba_pool_idx = None 模式。避免下一次 alloc() 看到过期引用而跳过分配，导致新池槽位泄漏。
3. python/sglang/srt/session/streaming_session.py - StreamingSession.try_cache_finished_req：对于 FINISH_ABORT 的临时槽位构造，新增携带 mamba_pool_idx 和 mamba_ping_pong_track_buffer 到槽位中；并在槽位创建后清除请求上的这些引用，防止释放时双重释放。此前临时槽位缺少 Mamba 字段，导致第一个请求的 abort 孤立 3 个槽位。

关键文件：

- python/sglang/srt/session/streaming_session.py（模块 流式会话；类别 source；类型 core-logic；符号 save_from_req, try_cache_finished_req）：核心修复点：save_from_req 中增加清除所有 Mamba 引用；try_cache_finished_req 中临时槽位增加 Mamba 字段并移除请求引用。
- python/sglang/srt/mem_cache/memory_pool.py（模块 内存池；类别 source；类型 core-logic；符号 free_mamba_cache）：free_mamba_cache 中增加清除请求上的 ping-pong 缓冲引用，匹配池侧释放后的引用归零。

关键符号：SessionSlot.save_from_req, HybridReqToTokenPool.free_mamba_cache, StreamingSession.try_cache_finished_req

关键源码片段

python/sglang/srt/session/streaming_session.py

核心修复点：save_from_req 中增加清除所有 Mamba 引用；try_cache_finished_req 中临时槽位增加 Mamba 字段并移除请求引用。

```
def save_from_req(self, req: Req, is_first: bool):
    """Save KV state from a finishing request into this slot."""
    self.req_pool_idx = req.req_pool_idx
    # ... 其他 KV 字段 ...
    self.mamba_pool_idx = req.mamba_pool_idx
    self.mamba_ping_pong_track_buffer = req.mamba_ping_pong_track_buffer
    self.mamba_next_track_idx = req.mamba_next_track_idx
    self.mamba_last_track_seqlen = req.mamba_last_track_seqlen
    self.mamba_branching_seqlen = req.mamba_branching_seqlen

    # Ownership has transferred to the slot. Null *all* of the req's
    # references so any later alloc()/free path that inspects the req
    # (e.g. the alloc-skip check on `req.mamba_ping_pong_track_buffer
    # is None` , or the retract cleanup) sees no dangling pointers
    # into slot-owned tensors. Without this the alloc path can decide
    # the req still has a ping-pong buffer and skip alloc, causing
    # the slot's tensor to be reused by a new req and leaked when
    # the slot is later freed.
    req.req_pool_idx = None
    req.mamba_pool_idx = None
    req.mamba_ping_pong_track_buffer = None
    req.mamba_next_track_idx = None
    req.mamba_last_track_seqlen = None
    req.mamba_branching_seqlen = None
```

python/sglang/srt/mem_cache/memory_pool.py

free_mamba_cache 中增加清除请求上的 ping-pong 缓冲引用，匹配池侧释放后的引用归零。

```
def free_mamba_cache(self, req: Req):
    # ... 原有的释放逻辑 ...
```

```
        self.mamba_pool.free(mamba_ping_pong_track_buffer_to_free)

        # Match the req.mamba_pool_idx=None clear above so the next
        # alloc() doesn't see a stale ping-pong reference on the req
        # and skip allocation (which would silently reuse a freed
        # tensor on the req side while the new pool slot leaks).
        req.mamba_ping_pong_track_buffer = None
        req.mamba_next_track_idx = None
```

评论区精华

审核人 ispobock 批准了 PR，并要求修复 lint（codespell 警告 re-use -> reuse），已在提交 dcae9c3 中修复。无其他设计争议或未解决疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 回归风险：低。修复仅添加了引用清除逻辑，不影响正常分配释放流程。
  - 性能风险：无。清除引用是 O(1) 操作。
  - 安全风险：无。
  - 兼容性：无。仅影响 mamba_extra_buffer + overlap_schedule 路径。
  - 测试覆盖：缺少直接测试。该修复针对一个特定配置组合，应添加回归测试以覆盖泄漏场景。
- 影响：
  - 用户：修复了使用混合 SSM 模型（如 Mamba 架构）且启用额外缓冲和重叠调度时，因内存泄漏导致调度器崩溃的问题。影响面窄，仅限于该配置组合。
  - 系统：消除了空闲时的 Mamba 池泄漏，使断言通过，避免进程崩溃。
  - 团队：提升了 Mamba + 流式会话的稳定性。
  - 风险标记：缺少测试覆盖，特定配置路径

关联脉络

- PR #25000 Reduce mamba prefill allocation overhead: 该 PR 涉及 Mamba 内存分配的优化，与本 PR 的 Mamba 池逻辑相关，属于同一功能线。