PR #22753 完整报告

sgl-project/sglang

Fix streaming session busy-check double-counting via active_pool_idxs

合并时间：2026-04-15 04:11

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22753

执行摘要

- 一句话：修复流式会话内存统计双计数问题，改用运行时计算活动池索引。
- 推荐动作：该 PR 值得精读，尤其是对于关注流式会话内存管理和调度器设计的工程师。重点关注从状态标志到运行时计算的架构转变，这种“单一事实来源”的设计模式在分布式或并发系统中常被用于避免状态不一致。同时，注意作者如何通过提交历史逐步重构，体现了良好的代码演进习惯。

功能与动机

根据 PR body 描述，这是流式会话内存记账修复系列（#22651）的一部分。问题在于流式会话的借用期间（restore_to_req -> forward -> save_from_req），session_held_tokens() 和 _get_total_uncached_sizes() 都会对相同的 KV 页面进行计数，导致 total_accounted > total，在 SGLANG_ENABLE_STRICT_MEM_CHECK_DURING_BUSY=2 下触发误报断言。PR #22213 引入的 SessionSlot.is_active 布尔标志在 retract/abort/speculative-v2-overlap 路径中生命周期配对会被破坏，导致标志卡在错误状态。

实现拆解

1. 移除易出错的标志：删除 SessionSlot 类中的 is_active 字段及其在 save_from_req 和 restore_to_req 方法中的设置，消除生命周期管理依赖。
2. 引入运行时活动池索引计算：在 SchedulerRuntimeCheckerMixin 中新增 _active_pool_idxs 方法，通过遍历 last_batch 和 running_batch 中的请求，收集所有 req_pool_idx 不为 None 的索引，形成当前被请求拥有的池索引集合。
3. 修改会话持有令牌计算方法：将 SessionAwareCache 中的 session_held_tokens、session_held_full_tokens 和 session_held_swa_tokens 方法改为接受 active_pool_idxs 参数，在计算时排除那些池索引在活动集合中的槽位，避免双重计数。
4. 更新调用方：调整 SchedulerRuntimeCheckerMixin 中的 _session_held_tokens、_session_held_full_tokens 和 _session_held_swa_tokens 方法，调用时传入 _active_pool_idxs() 的结果。
5. 清理与文档：删除无用的 is_active 字段和相关代码，更新方法文档以反映新的所有权逻辑。

关键文件：

- python/sglang/srt/mem_cache/session_aware_cache.py（模块 内存缓存；类别 source；类型 core-logic；符号 session_held_tokens, session_held_full_tokens,

session_held_swa_tokens, session_held_req_count）：核心变更文件，修改了会话持有令牌的计算逻辑，移除 is_active 标志，引入 active_pool_idxs 参数以避免双重计数。

- python/sglang/srt/managers/scheduler_runtime_checker_mixin.py（模块 调度管理；类别 source；类型 core-logic；符号 _active_pool_idxs）：新增 _active_pool_idxs 方法，提供活动池索引的运行时计算，并更新调用 session_held_tokens 等方法的接口。

关键符号：session_held_tokens, _active_pool_idxs, save_from_req, restore_to_req

关键源码片段

python/sglang/srt/mem_cache/session_aware_cache.py

核心变更文件，修改了会话持有令牌的计算逻辑，移除 is_active 标志，引入 active_pool_idxs 参数以避免双重计数。

```
def session_held_tokens(self, active_pool_idxs: Optional[set] = None) -> int:
    """Total KV tokens held by session slots, not tracked by the tree.

    Excludes slots whose KV is currently owned by an owning request —
    those tokens are counted via uncached_size in the busy mem check.
    A slot's pool_idx being in active_pool_idxs indicates a req owns it.
    """
    total = 0
    for slot in self.slots.values():
        # 判断槽位的池索引是否在当前活动请求的集合中
        in_batch = (
            active_pool_idxs is not None and slot.req_pool_idx in active_pool_idxs
        )
        # 仅统计持有KV且不被活动请求拥有的槽位
        if slot.is_holding_kv and not in_batch:
            allocated = ceil_align(slot.kv_allocated_len, self.page_size)
            total += allocated - slot.cache_protected_len
    return total
```

python/sglang/srt/managers/scheduler_runtime_checker_mixin.py

新增 _active_pool_idxs 方法，提供活动池索引的运行时计算，并更新调用 session_held_tokens 等方法的接口。

```
def _active_pool_idxs(self: Scheduler) -> set:
    """Pool idxs currently owned by reqs in last_batch / running_batch.

    Used to decide which session slots' KV is owned by batch reqs
    (and thus counted via uncached_size, not session_held).
    """
    idxs = set()
    # 遍历最近的两个批次（last_batch和running_batch）
    for batch in [self.last_batch, self.running_batch]:
        if batch is None or batch.is_empty():
            continue # 跳过空批次
        for req in batch.reqs:
```

```
            if req.req_pool_idx is not None:
                idxs.add(req.req_pool_idx) # 收集所有非None的池索引
        return idxs

    def _session_held_tokens(self: Scheduler) -> int:
        if isinstance(self.tree_cache, SessionAwareCache):
            # 调用时传入实时计算的活动池索引集合
            return self.tree_cache.session_held_tokens(self._active_pool_idxs())
        return 0
```

评论区精华

由于 review 评论为空，没有公开的讨论记录。从提交历史看，作者在三个提交中逐步完成了重构：首先引入 active_pool_idxs 并移除对 is_active 的依赖，然后删除未使用的 is_active 字段，最后统一注释术语为“owning request”。

- 暂无高价值评论线程

风险与影响

- 风险：技术风险：
- 回归风险：修改了核心的内存统计逻辑，如果 _active_pool_idxs 计算不准确（例如遗漏某些请求状态），可能导致统计错误，影响调度决策或内存检查。
- 性能影响：每次调用 session_held_tokens 时都需要遍历批次请求计算集合，可能增加 CPU 开销，但鉴于批次大小有限，影响应可控。
- 兼容性：SessionAwareCache 的接口变更（新增参数）可能影响直接调用这些方法的第三方代码，但根据上下文，这些方法主要在内部使用。 具体文件风险：
- python/sglang/srt/mem_cache/session_aware_cache.py：session_held_tokens 等方法的逻辑变更，需确保 in_batch 判断正确覆盖所有活动槽位。
- python/sglang/srt/managers/scheduler_runtime_checker_mixin.py：新增 _active_pool_idxs 方法需正确处理 batch 为 None 或空的情况。
- 影响：影响范围：
- 用户影响：对终端用户透明，但修复了可能导致内存断言失败的问题，提升系统稳定性。
- 系统影响：确保流式会话下的内存统计准确性，支持 retract、abort、speculative 等高级功能的严格内存检查。
- 团队影响：简化了流式会话内存管理逻辑，移除易出错的标志，降低了未来开发中的认知负担和 bug 风险。 影响程度：中等，主要影响流式会话的内存记账核心路径，但通过测试验证（所有流式会话测试在严格检查下通过）。
- 风险标记：核心路径变更，状态管理重构，接口参数变更

关联脉络

- PR #22651 enable streaming session retract tests: 该 PR 是流式会话内存记账修复系列的一部分，关联 Issue #22651 旨在启用流式会话 retract 测试，本修复确保测试在严格内

存检查下通过。

- PR #22213 未提供，但从 PR body 提及：PR body 提到 PR #22213 引入了 SessionSlot.is_active 标志，本 PR 移除了该标志并解决了其生命周期问题，属于对该设计的改进。
- PR #22755 Rename _alive_streaming_session_count; use _is_streaming helper: 同为流式会话相关的重构 PR，涉及调度器运行时检查的代码调整，可能共享类似的设计上下文。