PR #44490 完整报告

vllm-project/vllm

[Bugfix][Core] Fix host memory leak from undrained new_block_ids

合并时间：2026-07-07 15:32

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44490

执行摘要

- 一句话：修复 new_block_ids 未清空导致的主机内存泄漏
- 推荐动作：值得精读，特别是对于理解 V1 调度器与缓存管理器的交互契约以及条件记录 / 清空的设计权衡。展示了如何通过最小化 flag 传递来避免回归且保持正确性。

功能与动机

用户报告在长时间运行 max_tokens=1 负载时主机 RSS 线性增长（#44175）。根本原因是 #35219 添加的 new_block_ids 记录仅在全注意力 /MLA 块分配时追加，但清空操作仅在有 Mamba 层时执行，导致非 Mamba 模型的列表无限增长。gc.freeze 隐藏了此泄漏，使其表现为内存碎片。修复是必需的，否则长时间服务会导致 OOM。

实现拆解

1. 在 SingleTypeKVCacheManager.__init__中添加 needs_kv_cache_zeroing 参数，并根据 spec 类型设置 _record_new_block_ids 标志，控制是否记录新块 ID。
2. 将 allocate_new_blocks 和 allocate_external_computed_blocks 中的记录条件由类型检查替换为 _record_new_block_ids，避免非 Mamba 模型的不必要记录。
3. 在 Scheduler.schedule 方法中，无条件调用 kv_cache_manager.take_new_block_ids() 清空列表，然后根据 needs_kv_cache_zeroing 决定是否传递新块 ID 给 SchedulerOutput。
4. 在 KVCacheCoordinator 中创建管理器时传递 needs_kv_cache_zeroing 参数，确保记录 / 清空行为与模型类型一致。

关键文件：

- vllm/v1/core/single_type_kv_cache_manager.py（模块 缓存管理；类别 source；类型 core-logic；符号 SingleTypeKVCacheManager.init, allocate_new_blocks, allocate_external_computed_blocks）：核心修复文件：添加 needs_kv_cache_zeroing 构造函数参数和 _record_new_block_ids 标志，控制 new_block_ids 的记录，避免非 Mamba 模型的不必要开销。
- vllm/v1/core/sched/scheduler.py（模块 调度器；类别 source；类型 core-logic；符号 schedule）：修复调度器侧：无条件清空 new_block_ids，但仅当需要零化时传递给 SchedulerOutput。
- vllm/v1/core/kv_cache_coordinator.py（模块 协调器；类别 source；类型 configuration ；符号 KVCacheCoordinator.init）：传递 needs_kv_cache_zeroing 参数给单个管理器，

使其能够正确判断是否需要记录 ID。

关键符号：SingleTypeKVCacheManager.init, SingleTypeKVCacheManager.allocate_new_blocks, SingleTypeKVCacheManager.allocate_external_computed_blocks, Scheduler.schedule, KVCacheCoordinator.init

关键源码片段

vllm/v1/core/single_type_kv_cache_manager.py

核心修复文件：添加 needs_kv_cache_zeroing 构造函数参数和 _record_new_block_ids 标志，控制 new_block_ids 的记录，避免非 Mamba 模型的不必要开销。

```
# 构造函数新增 needs_kv_cache_zeroing 参数，用于控制是否记录新分配的块 ID
def __init__(
    self,
    kv_cache_spec: KVCacheSpec,
    block_pool: BlockPool,
    enable_caching: bool,
    kv_cache_group_id: int,
    scheduler_block_size: int,
    dcp_world_size: int = 1,
    pcp_world_size: int = 1,
    needs_kv_cache_zeroing: bool = False, # 新增参数：是否需要 KV cache 零化
    max_admission_blocks_per_request: int | None = None,
) -> None:
    # ...
    # 仅当需要零化且 spec 类型属于需要零化的类型时，才记录新块 ID
    self._record_new_block_ids = needs_kv_cache_zeroing and type(kv_cache_spec) in (
        FullAttentionSpec,
        TQFullAttentionSpec,
        MLAAttentionSpec,
        HiddenStateCacheSpec,
    )
    self.new_block_ids: list[int] = []
    # ...

# allocate_new_blocks 方法中，之前使用重复的类型检查，现在使用 _record_new_block_ids 标志
def allocate_new_blocks(self, request_id: str, num_new_blocks: int) -> list[KVCacheBlock]:
    # ...
    if self._record_new_block_ids: # 只有需要记录时才追加
        self.new_block_ids.extend(b.block_id for b in new_blocks)
    return new_blocks

def allocate_external_computed_blocks(self, request_id, ...):
    # ...
    if self._record_new_block_ids:
        self.new_block_ids.extend(b.block_id for b in allocated_blocks)
```

vllm/v1/core/sched/scheduler.py

修复调度器侧：无条件清空 new_block_ids，但仅当需要零化时传递给 SchedulerOutput。

```
# 在 schedule 方法末尾，无条件清空 new_block_ids 列表
# 只有需要零化时才将块 ID 传递给 SchedulerOutput
# 这确保了非 Mamba 模型不会积累未清空的 ID
new_attn_block_ids = self.kv_cache_manager.take_new_block_ids()
new_block_ids_to_zero = (
    (new_attn_block_ids or None) if self.needs_kv_cache_zeroing else None
)

scheduler_output = SchedulerOutput(
    # ... 其他字段
    new_block_ids_to_zero=new_block_ids_to_zero,
    # ...
)
```

评论区精华

主要讨论围绕设计权衡：作者最初避免添加构造函数标志，担心默认 False 会静默禁用零化路径；但 reviewer（njhill）采纳了 chaunceyjiang 的建议，添加 _record_new_block_ids 属性以减少非 Mamba 模型的不必要列表操作，确保不会引入性能开销。最终接受 flag 方案，并确认在协调器中正确传递参数。

- 无条件清空 new_block_ids 的安全性 (correctness): 接受无条件清空方案，对 Mamba 模型无影响。
- 添加 _record_new_block_ids 属性避免性能开销 (design): 接受 flag，确保在 KVCacheCoordinator 中正确传递参数。

风险与影响

- 风险：变更涉及调度器和缓存管理器的核心路径，但逻辑简单审慎。主要风险是 needs_kv_cache_zeroing 参数是否在所有管理器构造路径中正确传递；由于只有 KVCacheCoordinator 一处创建管理器，风险可控。另外，无条件清空 new_block_ids 的额外开销极低（仅 pop 列表），且 _record_new_block_ids 确保非 Mamba 模型追加操作为空。
- 影响：对所有 V1 标准注意力模型（如 Llama、Qwen、Gemma、DeepSeek-MLA）修复了长时间运行的内存泄漏，服务稳定性显著提升。Mamba 或混合 Mamba 模型无行为变化。影响范围覆盖大部分主流模型，但不涉及 API、前端或数据路径。
- 风险标记：调度器核心路径变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR