PR #43961 完整报告

vllm-project/vllm

[Bugfix] Corrupted MLA + linear attention

合并时间：2026-05-29 20:00

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43961

执行摘要

- 一句话：修复 MLA 注意力 KV 缓存腐败
- 推荐动作：值得合并，修复明确，风险极低。建议 reviewer 额外关注是否还有其他 attention spec 被遗漏（如未来的新类型），可考虑 vadiklyutiy 建议的简化方案——无条件清零所有新分配 block。另外，建议在开发者文档中记录哪些 attention kernel 需要清零 KV cache block。

功能与动机

Kimi Linear 模型（使用 MLA 注意力）在 KV 缓存填满后输出变为垃圾。根本原因是 PR #35219 仅覆盖了 FullAttentionSpec / TQFullAttentionSpec，而 MLAAttentionSpec 未被包含，导致 FlashMLA 和 FlashInfer MLA 后端无法避免 NaN 传播（它们依赖 * 0 而非 masking）。原始问题在 #35219 中已有讨论，但当时社区要求最小化变更，社区只添加了 FullAttentionSpec，后续又有 TQFullAttentionSpec，而 MLA 被遗漏了。

实现拆解

1. 定位问题：通过调试发现 KV 缓存腐败仅在 MLA + 线性注意力模型（Kimi Linear）中出现，且 KV 缓存填满后出现。
2. 确定修复位置：vllm/v1/core/single_type_kv_cache_manager.py 中有两处用于判断哪些 attention spec 的新分配 block 需要清零：allocate_new_computed_blocks() 和 allocate_new_blocks()。
3. 修改条件判断：在两处条件中均将 MLAAttentionSpec 加入 type(self.kv_cache_spec) in (...) 的元组中，使得 MLA 注意力模型的新分配 KV cache block ID 被记录到 self.new_block_ids，随后在 worker 侧被清零。
4. 验证：通过 GSM8K 多轮评估验证修复有效性。在 main 分支上，经过 epoch 1 后分数从 0.9075 跌至 0.7096，epoch 2 后跌至 0.0136；而修复后每个 epoch 分数稳定在 0.89 左右。
5. 配套措施：无额外测试或配置变更；仅源码修改，前置清零 kernel 在 #35219 中已实现。

关键文件：

- vllm/v1/core/single_type_kv_cache_manager.py（模块 KV 缓存管理；类别 source；类型 core-logic；符号 allocate_new_computed_blocks, allocate_new_blocks）：核心变更文件，修改了两处条件判断，将 MLAAttentionSpec 加入需要清零新分配 block 的 attention

spec 列表。

关键符号：allocate_new_computed_blocks, allocate_new_blocks

关键源码片段

vllm/v1/core/single_type_kv_cache_manager.py

核心变更文件，修改了两处条件判断，将 MLAAttentionSpec 加入需要清零新分配 block 的 attention spec 列表。

```
# vllm/v1/core/single_type_kv_cache_manager.py

# 在 allocate_new_computed_blocks 方法中，原条件只包含 FullAttentionSpec 和
TQFullAttentionSpec。
# 现在加入 MLAAttentionSpec，确保 MLA 注意力的新分配 block 也会被清零。
if type(self.kv_cache_spec) in (
    FullAttentionSpec,
    TQFullAttentionSpec,
    MLAAttentionSpec,
):
    self.new_block_ids.extend(b.block_id for b in allocated_blocks)

# 在 allocate_new_blocks 方法中做相同扩展
if type(self.kv_cache_spec) in (
    FullAttentionSpec,
    TQFullAttentionSpec,
    MLAAttentionSpec,
):
    self.new_block_ids.extend(b.block_id for b in new_blocks)
```

评论区精华

Review 中 vadiklyutiy（同时也是 #35219 的作者）确认：

- 最初尝试覆盖更通用的情况，但社区要求最小化变更，只限于 FullAttentionSpec。
- 后来有人添加了 TQFullAttentionSpec，但 MLA 再次被遗漏。
- 他建议是否应该重新考虑最初的决定，直接清零所有新分配的 block，避免后续再遗漏。
- gau-nernst 补充说，应该审查所有现有 attention kernel 中哪些容易受 NaN/Inf KV 缓存影响，并记录在文档中。
- 是否应该清零所有新分配 block (design): 未达成一致决定，但当前 PR 保持最小化修改，仅扩展覆盖范围。长期可能考虑统一清零所有新 block。

风险与影响

- 风险：风险极低。变更仅在条件判断中增加一种 attention spec 类型，不会影响其他注意力类型的逻辑。清零操作的 kernel 已在 #35219 中实现并上线，本 PR 只是扩展现有修复覆盖范围。可能的风险是漏掉其他未覆盖的 attention spec（如未来的新类型），但当前所有已知的易受影响类型都已包含。

- 影响：影响范围局限于使用 MLA 注意力且依赖 FlashMLA / FlashInfer MLA 后端的模型（如 Kimi Linear）。对这些模型，修复后 KV 缓存填满后模型输出不再腐败，评估分数稳定（例如 GSM8K 从 epoch 1 的 0.0136 恢复到 0.89）。对其他模型无影响。团队维护成本低：仅增加了两行代码。
- 风险标记：核心路径变更（KV 缓存管理）

关联脉络

- PR #35219 [BUGFIX][Mamba][Qwen3.5] Zero freed SSM cache blocks on GPU: 本 PR 继承并扩展了 #35219 的修复模式，从 FullAttentionSpec / TQFullAttentionSpec 扩展到 MLAAttentionSpec。