

PR #56526 完整报告

vllm-project/vllm

[ROCm][Kimi-K3] Fix non-contiguous state_indices crash and GPU-sync assert in fused KDA/MLA prefill

合并时间: 2026-09-12 09:17

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56526>

执行摘要

- 一句话: 修复 ROCm 上 Kimi-K3 预填充崩溃与 GPU-sync 断言
- 推荐动作: 值得快速精读 (约 15 分钟), 两个设计决策有复用价值: 一是 `.to(dtype)` 在 `dtype` 已匹配时是 no-op 的陷阱, 检查张量连续性不能依赖 `dtype` 转换, 应在内核调用边界统一收敛, 避免把约束散落到各调用方; 二是 `gpu_sync_allowed()` 的“同步点声明”模式, 让 GPU -> CPU 的预期同步在调试检查环境下显式豁免、默认路径零开销。局限是未补充单元测试, 后续在 `kimi_k3` 或 AITER MLA 路径上改动时, 建议顺手补一个针对非连续索引的回归用例。

功能与动机

PR body 明确指出修复目标是 `evals/gsm8k/test_gsm8k_correctness.py::test_gsm8k_correctness[Kimi-K3-pruned75-DSpark-AITER-TP4]` 在 ROCm 上的两阶段失败:

1) Warm-up 崩溃: `fused_kda_chunk` 断言 `state_indices` 必须为连续 int32 tensor, 而调用方传入的是 block table 的非连续列切片, `.to(torch.int32)` 因源 `dtype` 已是 int32 成为 no-op, 未修复连续性; 2) 预热后 100% 请求无效: `rocm_aiter_mla.py` 用 `.item()` 读取 GPU tensor 值但缺少 `gpu_sync_allowed()` 守卫, 触发仓库 post-warmup GPU-sync 调试检查。两处修复均限定在 ROCm-only 文件, 目标是让该评估用例稳定通过 (作者报告 `accuracy ≈ 0.33`, 达到 `0.33 ± 0.08` 阈值)。

实现拆解

1. 修复 warm-up 崩溃 (`vllm/models/kimi_k3/amd/ops/kda_chunk.py`): 在 `fused_kda_chunk` 组装底层 flashKDA 内核参数时, 将 `checkpoint_state_indices` 与 `state_indices` 从 `.to(torch.int32)` 改为 `.to(torch.int32).contiguous()`。原因是这两个张量来自 block table 的列切片, 是非连续视图, 且原 `dtype` 已是 int32, `torch.Tensor.to` 在 `dtype` 相同时是 no-op, 内存布局未被修复。修复放在内核调用边界而非上游调用方, 可覆盖所有调用路径, 避免同一问题在其他入口复现。
2. 修复 GPU-sync 断言 (`vllm/v1/attention/backends/mla/rocm_aiter_mla.py`): 新增 `from vllm.utils.gpu_sync_debug import gpu_sync_allowed` 导入, 并在 `_build_fp8_prefill_ps_metadata` 中用 `with gpu_sync_allowed()`: 包裹 `int(self.fp8_ps_reduce_indptr[-1].item())`。该 `.item()` 从 GPU tensor 取回 partial tile 数量, 是一次 GPU -> CPU 同步; 仓库在 `VLLM_GPU_SYNC_CHECK` 调试模式下会对预热后

的同步点做检查，未声明即触发断言。gpu_sync_allowed() 在默认生产路径是 no-op，仅用于显式标记预期同步，与代码库其他位置用法一致。

3. 测试与验证：本次未新增或修改测试文件，回归保障依赖 GSM8K 端到端评估用例；作者报告修复后测试稳定通过 (accuracy ≈ 0.33 ，达到 0.33 ± 0.08 阈值)，维护者 AndreasKaratzas 触发 Buildkite CI (#88429) 验证。

关键文件：

- vllm/v1/attention/backends/mla/rocm_aiter_mla.py (模块 注意力后端；类别 source；类型 core-logic；符号 _build_fp8_prefill_ps_metadata)：修复了预热后 100% 请求无效的 GPU-sync 断言问题：在 _build_fp8_prefill_ps_metadata 里将 .item() 同步点包入 gpu_sync_allowed()，并新增对应导入。这是两个故障中影响面更大的一个，也是 review 讨论的焦点。
- vllm/models/kimi_k3/amd/ops/kda_chunk.py (模块 KDA 算子；类别 source；类型 core-logic；符号 fused_kda_chunk)：修复 warm-up 阶段内核断言崩溃：fused_kda_chunk 在调用 flashKDA 内核前对 checkpoint_state_indices / state_indices 显式 .contiguous()，在调用边界统一满足内核的连续 int32 约束，覆盖所有调用方。

关键符号：fused_kda_chunk, _build_fp8_prefill_ps_metadata

关键源码片段

vllm/v1/attention/backends/mla/rocm_aiter_mla.py

修复了预热后 100% 请求无效的 GPU-sync 断言问题：在 _build_fp8_prefill_ps_metadata 里将 .item() 同步点包入 gpu_sync_allowed()，并新增对应导入。这是两个故障中影响面更大的一个，也是 review 讨论的焦点。

```
# 元数据构建末期需要知道当前 batch 真实活跃的 partial tile 数，
# 它保存在 GPU 侧 fp8_ps_reduce_indptr 的尾元素里。在构建期取回 CPU，
# 避免在每层 forward 路径上同步从而破坏 CUDA Graph 捕获。
# .item() 会触发 GPU -> CPU 同步；gpu_sync_allowed() 在默认生产路径是
# no-op，仅在开启 VLLM_GPU_SYNC_CHECK 调试检查时，把该同步声明为预期行为。
with gpu_sync_allowed():
    num_partial_tiles = int(self.fp8_ps_reduce_indptr[-1].item())
```

```
# 把 partial-sum 元数据挂到 metadata 对象，供 forward_mha 在注意力前向中读取。
metadata.fp8_prefill_qo_indptr = qo_indptr
metadata.fp8_prefill_kv_indptr = kv_indptr
metadata.fp8_prefill_kv_indices = kv_indices
metadata.fp8_prefill_work_indptr = self.fp8_ps_work_indptr
metadata.fp8_prefill_work_info_set = self.fp8_ps_work_info
metadata.fp8_prefill_reduce_indptr = self.fp8_ps_reduce_indptr
metadata.fp8_prefill_reduce_final_map = self.fp8_ps_reduce_final_map
metadata.fp8_prefill_reduce_partial_map = self.fp8_ps_reduce_partial_map
metadata.fp8_prefill_max_q_len = prefill.max_query_len
metadata.fp8_prefill_num_partial_tiles = num_partial_tiles
```

vllm/models/kimi_k3/amd/ops/kda_chunk.py

修复 warm-up 阶段内核断言崩溃：fused_kda_chunk 在调用 flashKDA 内核前对 checkpoint_state_indices / state_indices 显式 .contiguous()，在调用边界统一满足内核的连续 int32 约束，覆盖所有调用方。

```
# fused_kda_chunk 的收尾段：把索引参数统一转为 int32 后送入底层
# flashKDA DCP 内核（下方调用入口名未在本次 diff 变更中给出）。
# 修复背景：checkpoint_state_indices / state_indices 通常来自 block table
# 的列切片，属于非连续视图；它们的 dtype 恰好已是 int32，导致
# .to(torch.int32) 成为 no-op，连续性没有被修复，内核侧
# “must be contiguous int32”断言随之崩溃。现在在调用边界统一补
# .contiguous()，让所有调用方传入的索引都满足内核约束。
out, final_state = <flashKDA 内核调用>(
    None if checkpoint_offsets is None else checkpoint_offsets.to(torch.int32),
    None if checkpoint_state_indices is None
    else checkpoint_state_indices.to(torch.int32).contiguous(),
    state_cache,
    None if state_indices is None else state_indices.to(torch.int32).contiguous(),
    has_initial_state,
)
return out, final_state
```

评论区精华

review 中唯一实质讨论围绕 rocm_aiter_mla.py:800 的性能疑虑展开：

- AndreasKaratzas（合并者）提问：this might introduce a perf drop. Did you test for perf regression?
- divakar-amd 回应：It won't affect perf because in the default/production case, gpu_sync_allowed() is a no-op. We need it here because this test env sets VLLM_GPU_SYNC_CHECK (think of it as a debug check that warns/fails on GPU <-> CPU syncs). The sync (.item()) already existed before this PR. We are just telling the checker that this particular one is expected.
- 结论：作者澄清该守卫默认零开销、同步点本来就存在，改动只是把预期同步显式声明给调试检查器；合并者认可并以 APPROVED 结束，无遗留未解决问题。
- gpu_sync_allowed() 是否引入性能回退 (performance)：作者解释被接受：
gpu_sync_allowed() 默认零开销，改动不改变生产行为，仅显式标记预期同步；合并者以 APPROVED 收尾，无遗留问题。

风险与影响

- 风险：
 - 性能风险（低）：kda_chunk.py 新增的 .contiguous() 在输入非连续时会触发一次小型显存拷贝；state_indices 为 block table 列切片、长度较小，TP4 场景下单次拷贝开销可忽略，但频繁 prefill 下仍值得留意。
 - 行为风险（低）：gpu_sync_allowed() 在默认路径按实现是 no-op（作者已在 review 中确认），不改变生产行为；但它属于“豁免声明”模式，若后续代码演进让该同步点移出元

数据构建期，可能掩盖真实 GPU 同步问题。

- 覆盖风险（中）：两个修复均无配套单元测试，仅靠 GSM8K 端到端用例回归；该用例依赖特定模型、AITER、TP4 与 ROCm 环境，CI 未覆盖时容易悄悄回归。
- 平台边界：两处改动都在 ROCm 专属路径（`kimi_k3/amd` 与 `rocm_aiter_mla.py`），对 NVIDIA 等其他平台无影响。
- 影响：
 - 用户 / 评估侧：此前 Kimi-K3 pruned75 + DSpark + AITER + TP4 在 ROCm 上的 GSM8K 评估必然失败（warm-up 崩溃或预热后全量请求无效），修复后稳定通过，准确率约 0.33，达到阈值要求。
 - 系统侧：在开启 VLLM_GPU_SYNC_CHECK 调试检查的环境中，`num_partial_tiles` 的预期同步不再误报；默认路径无行为变化。
 - 团队侧：为 ROCm 侧 KDA/MLA 融合路径补齐了索引布局与同步声明两处规范用法，后续维护同类内核调用时可复用这两个模式。
 - 影响范围：变更面很小（2 文件共 8 行），聚焦单一平台单一模型路径，波及面可控。
 - 风险标记：ROCm 专属路径，缺少配套单元测试，潜在小拷贝开销

关联脉络

- PR #56485 [KDA] Update flashKDA to support bf16 checkpoint state: 同属 flashKDA 内核能力线：一个解决内核侧的 bf16 checkpoint 状态支持，本 PR 解决调用侧状态索引的布局约束，两者共同完善 KDA 路径。
- PR #55426 [Bugfix][Kimi-K3] Fix KDA projection overlap on Hopper: 同为 Kimi-K3 KDA 内核 bugfix 系列，体现 KDA 路径在 NVIDIA / AMD 双平台的持续加固。
- PR #56153 [ROCm][Kernel][DSV4] Remove tl.constexpr to avoid cold-compile churn in indexer gather kernel: 同处 `rocm_aiter_mla.py` 所在模块，反映 ROCm AITER prefill 路径近期在做性能与稳定性收敛。
- PR #56429 Revert "[Rocm][Kimi-k3] Fix pipeline_parallel support for the kimik3 DCP mode (#53664)": 同属 ROCm Kimi-K3 注意力路径的清理与维护，说明该平台路径近期处于活跃维护期。