PR #42810 完整报告

vllm-project/vllm

[ROCm] [Bugfix] Fix DeepSeek V4 Functionality and Accuracy

合并时间：2026-05-18 00:18

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42810

执行摘要

- 一句话：修复 ROCm 上 DeepSeek V4 功能与高并发精度问题
- 推荐动作：值得所有 ROCm + DeepSeek V4 用户关注。设计决策（AITER 回退、topk 统一入口）对类似平台适配有参考价值。建议阅读 rocm_aiter_mla_sparse.py 中的重构细节。

功能与动机

PR Body 指出：PR#41263 导致功能崩溃，且高并发时精度退化。AITER MHC kernel 在大 token 数下存在精度问题，prefill topk 索引缓冲区被错误逻辑破坏，因此需要回退 torch 实现并修复 topk 逻辑。

实现拆解

分为四个关键步骤：

1. 回退 AITER MHC 实现（mhc.py）：由于 AITER MHC kernel 在 token 数较大时存在精度问题，将 forward_hip 中条件分支（hidden_size % 256 == 0 走 AITER）永久注释，统一走 torch 实现，并添加 TODO 等待 AITER 修复后重新启用。
2. 修复 topk 索引逻辑（rocm_aiter_mla_sparse.py）：将 _topk_indices_torch 扩展支持 row_starts 参数，对齐 CUDA top_k_per_row_prefill 的语义（索引为行内偏移而非全局列偏移）。删除仅支持 topk_tokens=2048 的 _topk_indices_prefill 和 _topk_indices_decode 包装函数，直接调用 torch.ops._C.top_k_per_row_prefill（支持任意 topk 值）。同时清理 rocm_aiter_sparse_attn_indexer_native 函数接口，移除冗余包装器。
3. 简化 sparse attention indexer（sparse_attn_indexer.py）：移除之前通过 rocm_aiter_sparse_attn_indexer_native 的 fallback 路径，仅保留 AITER 算子（rocm_aiter_sparse_attn_indexer）。如果 AITER 未启用则直接抛出 RuntimeError，要求显式设置 VLLM_ROCM_USE_AITER=1。
4. 移除冗余 ffn_norm（deepseek_v4.py）：在 _forward_rocm 中删除 x = self.ffn_norm(x) 调用。FFN 的归一化已折叠到 self.ffn.norm_gate 中，ffn() 直接接收预归一化激活，此变更对齐了与 CUDA 路径的逻辑。

测试配套：PR 通过 lm-eval 在 GSM8K 上验证了 DeepSeek-V4-Pro 和 Flash，准确率达 95.5%/95.0%，置信度为回归无忧。

关键文件：

- vllm/model_executor/layers/mhc.py（模块 MHC 层；类别 source；类型 core-logic；符号 MHCPreOp.forward_hip, MHCPostOp.forward_hip）：核心修复：禁用有精度问题的 AITER MHC kernel，统一使用 torch 实现，影响 MHC pre/post 操作正确性。
- vllm/v1/attention/ops/rocm_aiter_mla_sparse.py（模块 稀疏索引；类别 infra；类型 refactor；符号 _topk_indices_torch, _topk_indices_prefill, _topk_indices_decode, rocm_aiter_sparse_attn_indexer_native）：重构热点：修复 topk 索引对齐问题，删除包装函数，简化接口，影响 prefill/decode 稀疏索引正确性。
- vllm/model_executor/layers/sparse_attn_indexer.py（模块 检索器；类别 source；类型 core-logic；符号 SparseAttnIndexer.forward_hip）：简化控制流：移除 native fallback，强制 AITER 路径，错误信息更明确。
- vllm/model_executor/models/deepseek_v4.py（模块 模型定义；类别 source；类型 core-logic；符号 DeepseekV4DecoderLayer._forward_rocm）：修复功能崩溃：移除冗余 ffn_norm 调用，对齐 CUDA 路径。

关键符号：_topk_indices_torch, _topk_indices_prefill, _topk_indices_decode, rocm_aiter_sparse_attn_indexer_native, MHCPreOp.forward_hip, MHCPostOp.forward_hip, SparseAttnIndexer.forward_hip, DeepseekV4DecoderLayer._forward_rocm

关键源码片段

vllm/model_executor/layers/mhc.py

核心修复：禁用有精度问题的 AITER MHC kernel，统一使用 torch 实现，影响 MHC pre/post 操作正确性。

```
# vllm/model_executor/layers/mhc.py 中的 MHCPostOp 类（MHC 后处理操作）
# 修改后的 forward_hip：始终使用 torch kernel，不再根据 hidden_size 分流到 AITER
class MHCPostOp(CustomOp):
    def forward_hip(
        self,
        x: torch.Tensor,
        residual: torch.Tensor,
        post_layer_mix: torch.Tensor,
        comb_res_mix: torch.Tensor,
    ) -> torch.Tensor:
        # TODO: Reenable aiter after we are at the aiter
        # version that has this bugfix
        # https://github.com/ROCm/aiter/commit/b639cb63bcac4672dce33a731fad042a65cb3649
        # It has accuracy problem at large number of tokens.
        # 原条件分支（hidden_size % 256 == 0）被注释，直接走 torch 路径
        return mhc_kernels.mhc_post_torch(
            x,
            residual,
            post_layer_mix,
            comb_res_mix,
        )
```

vllm/v1/attention/ops/rocm_aiter_mla_sparse.py

重构热点：修复 topk 索引对齐问题，删除包装函数，简化接口，影响 prefill/decode 稀疏索引正确性。

```
# vllm/v1/attention/ops/rocm_aiter_mla_sparse.py
# 修改后的 _topk_indices_torch：支持 row_starts 参数，输出行内本地索引

def _topk_indices_torch(
    logits: torch.Tensor,
    topk_tokens: int,
    row_starts: torch.Tensor | None = None,
) -> torch.Tensor:
    """Compute top-k indices using torch.topk, with optional row offset correction.

    When `row_starts` is provided, indices are local within each row's valid
    range (matching CUDA `top_k_per_row_prefill` contract).
    """
    k = min(topk_tokens, logits.shape[-1])
    values, indices = torch.topk(logits, k=k, dim=-1)
    indices = indices.to(torch.int32)
    if k < topk_tokens:
        # 填充无效列索引为 -1
        padding = torch.full_like(indices, -1, dtype=torch.int32)
        indices = torch.cat([indices, padding], dim=-1)
    if row_starts is not None:
        starts = row_starts.to(dtype=torch.int32).view(-1, 1)
        # 将绝对列索引转换为行内本地索引
        indices = torch.where(indices < 0, indices, indices - starts)
    if k == topk_tokens:
        return indices
    # 如果实际 topk 小于请求的 topk_tokens，则填充到指定大小
    padded = torch.full(
        (logits.shape[0], topk_tokens), -1, dtype=torch.int32, device=logits.device
    )
    padded[:, :k] = indices
    return padded

# 原有的 _topk_indices_prefill 和 _topk_indices_decode 函数已被移除，
# 因为它们仅针对 topk_tokens=2048 做了专用路径优化，而新版 C++ kernel 支持任意值。
```

评论区精华

1. Critical：topk buffer 切片大小不匹配（@gemini-code-assist[bot]） - 问题：解码路径用 num_decode_tokens 而非 num_padded_tokens 切片 topk_indices_buffer，当 padding 开启时导致 OOB 写入和后续 reshape 失败。 - 结论：作者在后续 commit（"num_padded_tokens"）中修复了此问题，切片改为 num_padded_tokens。
2. 禁用 AITER MHC 的性能考量（@tjtanaa） - 作者说明：虽然回退到 torch 实现，但在 CUDA graph 和 torch.compile 模式下，该操作已被优化，性能优于常规 torch，并非瓶颈。

3. 清理包装函数的原因（@tjtanaa） - 自评：不再需要另一层包装，直接调用 C++ kernel 更简洁且支持所有 topk_tokens 值。

- topk buffer 切片大小不匹配 (correctness): 作者在后续 commit "num_padded_tokens" 中修复了该问题，最终代码使用 num_padded_tokens 切片。
- 禁用 AITER MHC 的性能考量 (performance): 团队接受该变通方案，等待 AITER 修复后重新启用。
- 包装函数清理 (design): 同意清理，减少间接层。

风险与影响

- 风险：
  - AITER MHC 回退风险：禁用 AITER 后，MHC pre/post 操作使用纯 torch 实现。尽管作者称 torch.compile 已优化，但在某些场景（如极端长度）仍可能比专用 AITER kernel 慢。需后续跟踪性能回归。
  - Sparse attn indexer 强制依赖 AITER：若用户未设置 VLLM_ROCM_USE_AITER=1，模型会直接抛 RuntimeError。环境依赖增强，但明确性提升。
  - Topk buffer 越界风险（已修复）：gemini 指出的 decode 路径切片问题在后续 commit 中已解决，审查时应确认最终代码使用 num_padded_tokens。
  - 无新增测试：本次改动未添加对应单元测试，回归覆盖依赖 manual lmeval。
- 影响：
  - 用户影响：DeepSeek V4/V4 Flash 在 ROCm 上功能和精度恢复，但需确保 AITER 已安装并设置环境变量。
  - 系统影响：删除 ~177 行代码（净减 89 行），降低维护成本；sparse_attn_indexer 路径简化，错误信息更明确。
  - 团队影响：与近期 PR#41710（清理 norm）方向一致，体现持续简化 DeepSeek V4 代码的风格。
  - 风险标记：AITER MHC 精度风险，Sparse attn indexer 强制依赖 AITER, Topk buffer size 潜在越界（已修复）

关联脉络

- PR #41263 Unavailable (PR#41263): 此 PR 引入了功能崩溃，本 PR 的第一项修复就是回退其影响。
- PR #41710 fix: remove unused norm for dpskv4: 同样涉及 DeepSeek V4 的 norm 清理，体现了团队持续简化代码的方向。