PR #42767 完整报告

vllm-project/vllm

[Refactor] Remove dead cuda kernels

合并时间：2026-05-19 02:14

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42767

执行摘要

- 一句话：移除三个未使用的 CUDA 内核及其 Python 绑定
- 推荐动作：该 PR 是常规的代码清理，逻辑清晰无设计争议，值得快速合并。可供参考的是守卫条件的隐式依赖处理方式，在跨平台多后端项目中应习惯使用具体算子存在判断而非顶层模块存在判断。

功能与动机

PR 标题和正文明确指出目的为『Remove cuda dead kernels』，包括 marlin_gemm_moe、convert_vertical_slash_indexes 和 convert_vertical_slash_indexes_mergehead，这些内核在代码库中已无调用方，属于死代码清理，旨在降低编译负担和代码维护成本。

实现拆解

1. Python 包装层：在 vllm/_custom_ops.py 中移除了 convert_vertical_slash_indexes 和 convert_vertical_slash_indexes_mergehead 两个函数以及 marlin_gemm_moe_fake 的虚假注册，并收紧 moe_wna16_marlin_gemm_fake 的守卫条件。
2. C++ 头文件：在 csrc/ops.h 中删除了 #ifndef USE_ROCM 内的两个函数声明。
3. CUDA 内核源文件：完整删除 csrc/attention/vertical_slash_index.cu（401 行）。
4. Torch 绑定注册：在 csrc/torch_bindings.cpp 中移除两个 convert_vertical_slash_indexes 的绑定，在 csrc/moe/torch_bindings.cpp 中移除 marlin_gemm_moe 的绑定。
5. 构建系统：在 CMakeLists.txt 的源文件列表中移除 vertical_slash_index.cu。

关键文件：

- vllm/_custom_ops.py（模块 Python 绑定；类别 source；类型 core-logic；符号 convert_vertical_slash_indexes, convert_vertical_slash_indexes_mergehead, marlin_gemm_moe_fake）：删除两个 Python 包装函数和 fake kernel 注册，是变更的核心入口。
- csrc/ops.h（模块 C++ 核心；类别 source；类型 core-logic）：删除两个 C++ 函数声明，与 Python 包装对应。
- csrc/attention/vertical_slash_index.cu（模块 注意力 CUDA；类别 other；类型 deletion）：完整的 CUDA 内核源文件被删除，减少约 400 行编译量。

- csrc/torch_bindings.cpp（模块 Torch 绑定；类别 source；类型 core-logic）：移除两个 CUDA 算子的 torch 绑定注册。
- csrc/moe/torch_bindings.cpp（模块 MoE CUDA；类别 source；类型 core-logic）：移除 marlin_gemm_moe 的 torch 绑定注册。
- CMakeLists.txt（模块 构建配置；类别 docs；类型 documentation）：构建系统同步删除 vertical_slash_index.cu 的编译条目。

关键符号：convert_vertical_slash_indexes, convert_vertical_slash_indexes_mergehead, marlin_gemm_moe_fake

关键源码片段

vllm/_custom_ops.py

删除两个 Python 包装函数和 fake kernel 注册，是变更的核心入口。

```
# merge attn states ops
def merge_attn_states(
    output: torch.Tensor,
    prefix_output: torch.Tensor,
    prefix_lse: torch.Tensor,
    suffix_output: torch.Tensor,
    suffix_lse: torch.Tensor,
    output_lse: torch.Tensor | None = None,
    prefill_tokens_with_context: int | None = None,
    output_scale: torch.Tensor | None = None,
) -> None:
    torch.ops._C.merge_attn_states(
        output,
        output_lse,
        prefix_output,
        prefix_lse,
        suffix_output,
        suffix_lse,
        prefill_tokens_with_context,
        output_scale,
    )

# 移除的死 CUDA 内核：convert_vertical_slash_indexes 和 convert_vertical_slash_indexes_
mergehead
# 此前位于 merge_attn_states 与 pos encoding ops 之间。

# pos encoding ops
def rotary_embedding(
    positions: torch.Tensor,
    query: torch.Tensor,
    key: torch.Tensor | None,
    head_size: int,
    cos_sin_cache: torch.Tensor,
```

```
    is_neox: bool,
    rope_dim_offset: int = 0,
    inverse: bool = False,
) -> None:
    if rope_dim_offset == 0 and not inverse:
        torch.ops._C.rotary_embedding(
            positions, query, key, head_size, cos_sin_cache, is_neox
        )
    else:
        torch.ops._C.rotary_embedding(
            positions,
            query,
            key,
            head_size,
            cos_sin_cache,
            is_neox,
            rope_dim_offset,
            inverse,
        )
```

csrc/ops.h

删除两个 C++ 函数声明，与 Python 包装对应。

```
void merge_attn_states(
    torch::Tensor& output, std::optional<torch::Tensor> output_lse,
    const torch::Tensor& prefix_output, const torch::Tensor& prefix_lse,
    const torch::Tensor& suffix_output, const torch::Tensor& suffix_lse,
    const std::optional<int64_t> prefill_tokens_with_context,
    const std::optional<torch::Tensor>& output_scale = std::nullopt);

// 此前 #ifndef USE_ROCM 块内包含 convert_vertical_slash_indexes 和
// convert_vertical_slash_indexes_mergehead 两个声明，现已移除。

void rms_norm(torch::Tensor& out, torch::Tensor& input, torch::Tensor& weight,
              double epsilon);
```

评论区精华

两条 review 评论均已解决：

- gemini-code-assist[bot]指出初始的守卫条件 if hasattr(torch.ops, "_moe_C") 在 ROCm 上可能因 moe_wna16_marlin_gemm 不存在而导致初始化崩溃，建议改为检查具体算子。PR 采纳意见，最终改为 if hasattr(torch.ops, "_moe_C") and hasattr(torch.ops._moe_C, "moe_wna16_marlin_gemm"):。
- mgoin提醒需要同步更新 CMakeLists.txt 移除源文件引用，后续 commit 已补充。
- 守卫范围过宽影响 ROCm (correctness): 作者已修改守卫为需同时检查 _moe_C 和 moe_wna16_marlin_gemm 存在。
- CMakeLists 缺少同步更新 (other): 后续 commit 已补充 CMakeLists 修改。

风险与影响

- 风险：风险极低。待删除的内核在代码库中已无任何调用点（由提交者确认），移除后不会影响现有功能。唯一的间接风险是 moe_wna16_marlin_gemm_fake 守卫的调整：若守卫太宽可能导致 ROCm 初始化失败，但最终版本已加入更具体的算子存在检查，风险已消除。建议在 CI 中确认 ROCm 和 CUDA 均通过编译。
- 影响：对用户无可见影响；对系统可减少约 400 行编译量，略微提升构建速度；对团队维护负担降低，代码库更干净。没有公共 API 变更或运行时行为变化。
- 风险标记：无回归风险，构建加速

关联脉络

- PR #42483 Refactor AWQ Marlin MoE onto modular WNA16 oracle: 重构 AWQ Marlin MoE 后旧的 marlin_gemm_moe 内核不再被引用，本 PR 将其清理。