

PR #49621 完整报告

vllm-project/vllm

Remove triton per group quant [ROCm] [Bugfix]

合并时间: 2026-07-28 13:53

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49621>

执行摘要

- 一句话: 删除冗余 Triton group quant op, 恢复 ROCm FP8 fusion
- 推荐动作: 建议所有 ROCm FP8 用户合并此 PR。对于研究编译融合或量化实现的开发者, 值得阅读以理解 op 注册与融合 pass 交互。

功能与动机

On ROCm, dynamic FP8 per-group activation quant was routed through a `triton_per_token_group_quant_fp8` custom op. That op adds no compute, but it blocks RMSNorm + quant fusion. The fusion matchers only emit `rocm_aiter_group_fp8_quant` or `_C.per_token_group_fp8_quant`, never the triton wrapper. So any layer hitting the tuned-GEMM list (e.g. DeepSeek-R1 MLA `q_b_proj` (3072, 1536)) emitted an unmatchable op and left a standalone group-quant kernel unfused — ~9% of decode GPU time in an 8xTP DeepSeek-R1 profile. (引自 PR body)

实现拆解

1. 移除 custom op 注册: 在 `fp8_utils.py` 中删除 `_triton_per_token_group_quant_fp8_impl` 和 `_triton_per_token_group_quant_fp8_fake` 函数及 `direct_register_custom_op` 调用, 移除不再需要的导入。
2. 简化 `QuantFP8.forward_hip` 路由: 在 `input_quant_fp8.py` 的 `forward_hip` 方法中, 删除 `if self.is_group_quant and use_triton`: 早期返回分支。现在 group quant 直接通过 `aiter` 或 fallback 到 `forward_cuda`。
3. 清理测试和文档: `test_fusion_all_reduce.py` 删除 `use_triton_quant` 参数及分支; `test_silu_mul_quant_fusion.py` 移除 monkey-patch; `test_fusion.py` 移除条件分支; `allreduce_rms_fusion.py` 更新文档字符串。
4. 测试与验证: 单元测试通过 (部分已有失败无关); 精度测试 DeepSeek-R1 GSM8K 无 regression。

关键文件:

- `vllm/model_executor/layers/quantization/utils/fp8_utils.py` (模块 量化工具; 类别 source; 类型 data-contract; 符号 `_triton_per_token_group_quant_fp8_impl`, `_triton_per_token_group_quant_fp8_fake`): 删除 `triton_per_token_group_quant_fp8` custom op 的完整实现、fake 实现及注册函数, 是 PR 的核心变更, 直接移除冗余包装。

- vllm/model_executor/layers/quantization/input_quant_fp8.py (模块 输入量化; 类别 source; 类型 core-logic) : 在 forward_hip 中删除了导致 fusion 阻塞的早期返回分支, 使 group quant 直接走 aiter/C++ 路径。
- tests/compile/passes/distributed/test_fusion_all_reduce.py (模块 融合测试; 类别 test; 类型 test-coverage) : 移除 use_triton_quant 参数及相关分支, 简化测试模型, 与源代码变更同步。
- tests/compile/passes/test_silu_mul_quant_fusion.py (模块 融合测试; 类别 test; 类型 test-coverage) : 移除手动设置 use_triton=True 的 monkey-patch, 使测试不再依赖已删除的 op。
- vllm/compilation/passes/fusion/allreduce_rms_fusion.py (模块 编译融合; 类别 source; 类型 core-logic) : 更新 fusion pass 注释文档, 统一说明匹配的 op 仅为 rocm_aiter_group_fp8_quant。
- tests/compile/passes/test_fusion.py (模块 融合测试; 类别 test; 类型 test-coverage) : 移除返回 triton_per_token_group_quant_fp8 的分支, 与核心变更对齐。

关键符号: _triton_per_token_group_quant_fp8_impl,
_triton_per_token_group_quant_fp8_fake

关键源码片段

vllm/model_executor/layers/quantization/input_quant_fp8.py

在 forward_hip 中删除了导致 fusion 阻塞的早期返回分支, 使 group quant 直接走 aiter/C++ 路径。

```
def forward_hip(
    self,
    x: torch.Tensor,
    scale: torch.Tensor | None = None,
    scale_ub: torch.Tensor | None = None,
    use_triton: bool = False,
) -> tuple[torch.Tensor, torch.Tensor]:
    # 注意: 原处删除了 `if self.is_group_quant and use_triton:` 分支,
    # 该分支曾直接调用 `torch.ops.vllm.triton_per_token_group_quant_fp8`,
    # 现已统一走以下路径。

    use_aiter_quant = self.use_aiter and scale_ub is None and x.is_contiguous()
    use_aiter_per_tensor_quant = (
        use_aiter_quant and self.group_shape.is_per_tensor()
    )
    use_aiter_per_token_quant = use_aiter_quant and self.group_shape.is_per_token()
    use_aiter_per_group_quant = use_aiter_quant and self.group_shape.is_per_group()

    if use_aiter_per_group_quant:
        return rocm_aiter_ops.group_fp8_quant(x, self.group_size)
    if use_aiter_per_tensor_quant:
        return rocm_aiter_ops.per_tensor_quant(x, _FP8_DTYPE, scale)
    if use_aiter_per_token_quant:
```

```
return rocm_aiter_ops.per_token_quant(x, _FP8_DTYPE, scale)
```

```
# Fallback to CUDA implementation
```

```
return self.forward_cuda(x, scale, scale_ub)
```

评论区精华

两位 ROCm 团队开发者（Rohan138、dllehr-amd）直接批准，评论 "LGTM, thanks!" 和 "Thanks Andy!"，表明变更清晰无争议。无实质技术讨论。

- 暂无高价值评论线程

风险与影响

- 风险：
 - 回归风险：不使用 aiter 时，forward_hip fallback 到 forward_cuda，其内部使用 ops.scaled_fp8_quant，与之前通过 Triton wrapper 调用的 C++ kernel 逻辑一致，行为不变。
 - 兼容风险：删除了公开的 custom op triton_per_token_group_quant_fp8，但该 op 仅内部使用且已被 #42758 的 C++ kernel 替代，风险低。
 - 精度风险：精度测试无 regression。
 - 影响：用户影响：ROCm 平台使用 FP8 per-group dynamic quantization 的用户将获得 decode 阶段 ~9% 加速（DeepSeek-R1）。CUDA 用户无影响。系统影响：编译融合 pass 现在能正确匹配 RMSNorm+group quant 模式，减少 kernel 启动次数。团队影响：移除无用抽象层，降低维护成本。
- 风险标记：核心路径变更，兼容性风险（删除自定义 op）

关联脉络

- PR #42758 [ROCm] Enable per_token_group_fp8_quant C++ kernel on ROCm: 该 PR 使 C++ kernel 在 ROCm 上可用，使 triton wrapper 冗余，本 PR 进一步移除 wrapper。
- PR #44437 [ROCm] RMSNorm+MXFP4 fusion (actively developed): 在同一融合文件中有重叠的文档字符串行，本 PR 修改了相关行，需注意 rebase 兼容。