PR #43930 完整报告

vllm-project/vllm

[XPU][Bugfix] Fix per_token_group_fp8_quant missing dummy args on XPU

合并时间：2026-06-02 11:09

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43930

执行摘要

- 一句话：修复 XPU 上 FP8 量化少传 2 个参数的问题
- 推荐动作：此 PR 为必要的 bugfix，改动小而精，值得合入。建议在合入后验证 XPU 上 FP8 量化功能正常。

功能与动机

XPU 平台的 per_token_group_fp8_quant 算子注册签名要求 10 个参数，但两个调用点只传了 8 个参数，导致运行时错误。PR 描述明确指出需要补齐缺失的 column_major_scales 和 tma_aligned_scales 两个参数。

实现拆解

1. 统一 XPU 和 CUDA 的条件分支：在 vllm/model_executor/layers/quantization/utils/fp8_utils.py 的 per_token_group_quant_fp8 函数中，将原先独立的 if current_platform.is_xpu() 分支删除，改为将 XPU 合并到 if current_platform.is_cuda() or current_platform.is_xpu() 条件中，这样 XPU 调用也能传入完整的 10 个参数。
2. 补齐 MXFP8 路径的 dummy 参数：在 vllm/_xpu_ops.py 的 _xpu_mxfp8_quantize_impl 函数中，torch.ops._C.per_token_group_fp8_quant 调用由 8 个参数扩展为 10 个参数，新增 False 作为 column_major_scales 和 tma_aligned_scales 的 dummy 值。
3. 无参数值行为变更：对于 XPU 路径，新增的两个参数均设为 False，不改变原有量化逻辑。

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 per_token_group_quant_fp8）：核心量化函数，合并 XPU 条件分支，删除独立调用路径，确保参数完整传递。
- vllm/_xpu_ops.py（模块 XPU 算子；类别 source；类型 core-logic；符号 _xpu_mxfp8_quantize_impl）：XPU 专用 MXFP8 量化函数，补齐 dummy 参数。

关键符号：per_token_group_quant_fp8, _xpu_mxfp8_quantize_impl

关键源码片段

vllm/model_executor/layers/quantization/utils/fp8_utils.py

核心量化函数，合并 XPU 条件分支，删除独立调用路径，确保参数完整传递。

```
# 变更前：XPU 有独立分支，只传 8 个参数
if current_platform.is_cuda() and x.is_contiguous():
    torch.ops._C.per_token_group_fp8_quant(
        x, x_q, x_s, group_size, eps, fp8_min, fp8_max,
        use_ue8m0, column_major_scales, tma_aligned_scales,
    )
    return x_q, x_s

# 变更前：XPU 分支，只传 8 个参数（缺少 column_major_scales 和 tma_aligned_scales）
if current_platform.is_xpu() and x.is_contiguous():
    torch.ops._C.per_token_group_fp8_quant(
        x, x_q, x_s, group_size, eps, fp8_min, fp8_max, use_ue8m0
    )
    return x_q, x_s

# 变更后：统一 CUDA 和 XPU 分支，均传入 10 个参数
def per_token_group_quant_fp8(x, ...):
    # ... 前面分配 scale 的逻辑不变 ...
    # 统一条件：无论是 CUDA 还是 XPU，都走同一个分支
    if (current_platform.is_cuda() or current_platform.is_xpu()) and x.is_contiguous():
        torch.ops._C.per_token_group_fp8_quant(
            x,
            x_q,
            x_s,
            group_size,
            eps,
            fp8_min,
            fp8_max,
            use_ue8m0,
            column_major_scales, # 始终传入
            tma_aligned_scales, # 始终传入
        )
        return x_q, x_s
    # ... 后续 Triton fallback 不变 ...
```

vllm/_xpu_ops.py

XPU 专用 MXFP8 量化函数，补齐 dummy 参数。

```
def _xpu_mxfp8_quantize_impl(
    x: torch.Tensor, dtype: torch.dtype | None = None
) -> tuple[torch.Tensor, torch.Tensor]:
    MXFP8_BLOCK_SIZE = 32
    # ... 前面的初始化逻辑不变 ...

    x_q = torch.empty_like(x, device=x.device, dtype=dtype)
    shape = x.shape[:-1] + (x.shape[-1] // MXFP8_BLOCK_SIZE,)
    x_s = torch.empty(shape, device=x.device, dtype=torch.float32)

    # 变更前：只传 8 个参数
```

```
    # torch.ops._C.per_token_group_fp8_quant(
    # x, x_q, x_s, MXFP8_BLOCK_SIZE, eps, fp8_min, fp8_max, True
    # )

    # 变更后：补齐两个 dummy 参数，保持与算子签名一致
    torch.ops._C.per_token_group_fp8_quant(
        x,
        x_q,
        x_s,
        MXFP8_BLOCK_SIZE,
        eps,
        fp8_min,
        fp8_max,
        True,
        False, # dummy (column_major_scales)
        False, # dummy (tma_aligned_scales)
    )

    x_s = x_s.to(torch.float8_e8m0fnu)
    return x_q, x_s
```

评论区精华

Review 讨论较少，主要依赖关系确认：

- @majian4work 确认变更依赖 vllm-xpu-kernels 的修改，但该 PR 本身是安全的。
- @jikunshang 和 @xwu-intel 认为应优先合并此 bugfix，再 rebase 某个 feature PR。
- 依赖关系确认 (other): 此 bugfix 是独立的，但需要确保 XPU 内核已更新支持 10 参数。

风险与影响

- 风险：风险很低：
  - 修复仅补齐参数，不改变逻辑，且新增参数均为 False，不会影响 CUDA 或其他平台。
  - 但需确保 XPU 内核端已支持 10 参数签名，否则仍会报错。
  - 影响：直接影响 XPU 平台的 FP8 量化功能，确保其与最新的算子签名一致。不影响其他平台（CUDA、ROCm 等）。
  - 风险标记：外部依赖同步

关联脉络

- PR #39968 [Feature] XPU FP8 quantization support (推测): Issue 评论中提及此 feature PR 依赖于本 bugfix，本 bugfix 合入后需要 rebase 该 PR。