PR #43277 完整报告

vllm-project/vllm

[XPU] add scale transpose to prepare_fp8_moe_layer_for_xpu and bump up kernels

合并时间：2026-05-29 11:22

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43277

执行摘要

- 一句话：XPU FP8 MoE 层支持 scale 转置
- 推荐动作：该 PR 修复了一个明确的 bug，改动量小且正确性经过 review 确认。值得合并，但建议后续补充针对不同 scale 维度的单元测试。

功能与动机

Block FP8 量化时，权重和 scale 张量都需要进行转置才能适配 XPU 内核的输入布局要求。原实现仅对权重 w13 和 w2 进行转置，遗漏了 scale 张量，导致 block FP8 模式无法正确工作。

实现拆解

1. 修改 prepare_fp8_moe_layer_for_xpu 函数签名（xpu_moe.py）：新增 w13_scale 和 w2_scale 参数，返回值扩展为 4 元组 (w13, w13_scale, w2, w2_scale)。
2. 添加条件转置逻辑：仅当 scale 不为 None 且维度为 3（即 block FP8 格式）时，执行 .transpose(-1, -2).contiguous()；否则保持原样。
3. 更新调用点（oracle/fp8.py）：在 Fp8MoeBackend.XPU 分支中，传递 w13_scale 和 w2_scale 给新签名的函数，并正确解包返回值。
4. 升级 XPU 内核依赖（requirements/xpu.txt）：将 vllm_xpu_kernels 从 v0.1.8 升级到 v0.1.9，以匹配新 scale 处理逻辑。

关键文件：

- vllm/model_executor/layers/fused_moe/experts/xpu_moe.py（模块 MoE 专家层；类别 source；类型 data-contract；符号 prepare_fp8_moe_layer_for_xpu）：核心变更文件：修改了 prepare_fp8_moe_layer_for_xpu 函数，支持 scale 转置。
- vllm/model_executor/layers/fused_moe/oracle/fp8.py（模块 MoE 格式化器；类别 source；类型 data-contract）：调用侧适配：更新 XPU 后端分支，传递 scale 参数并正确解包返回值。
- requirements/xpu.txt（模块 依赖配置；类别 docs；类型 documentation）：依赖升级：vllm_xpu_kernels 从 v0.1.8 升至 v0.1.9，以匹配 scale 处理需求。

关键符号：prepare_fp8_moe_layer_for_xpu

关键源码片段

vllm/model_executor/layers/fused_moe/experts/xpu_moe.py

核心变更文件：修改了 prepare_fp8_moe_layer_for_xpu 函数，支持 scale 转置。

```
# SPDX-License-Identifier: Apache-2.0
# SPDX-FileCopyrightText: Copyright contributors to the vLLM project
import torch

from vllm.platforms import current_platform

if current_platform.is_xpu():
    from vllm_xpu_kernels.fused_moe_interface import XpuFusedMoe


def prepare_fp8_moe_layer_for_xpu(
    w13: torch.Tensor,
    w13_scale: torch.Tensor, # New: scale for w13, can be None
    w2: torch.Tensor,
    w2_scale: torch.Tensor, # New: scale for w2, can be None
) -> tuple[torch.Tensor, torch.Tensor, torch.Tensor, torch.Tensor]:
    # 仅对 3D block FP8 scale 进行转置，避免对 per-tensor (0D/1D) 或 per-channel (2D) 误操作
    if w13_scale is not None and w13_scale.ndim == 3:
        w13_scale = w13_scale.transpose(-1, -2).contiguous()
    if w2_scale is not None and w2_scale.ndim == 3:
        w2_scale = w2_scale.transpose(-1, -2).contiguous()
    return (
        w13.transpose(-1, -2).contiguous(), # 权重始终转置
        w13_scale,
        w2.transpose(-1, -2).contiguous(),
        w2_scale,
    )
```

评论区精华

gemini-code-assist[bot] 指出现有实现无条件调用 .transpose(-1, -2) 存在风险：对于 per-tensor（0D/1D）或 per-channel（2D）scale 会引发 RuntimeError 或产生错误结果。建议添加 ndim == 3 条件限制，并更新类型提示为 torch.Tensor | None。作者已在后续提交中修复（添加了 if w13_scale is not None and w13_scale.ndim == 3 检查）。

- 无条件 transpose 可能导致 RuntimeError (correctness): 作者添加了 ndim == 3 条件限制，修复了潜在错误。

风险与影响

- 风险：低风险。变更局限于 XPU 平台的 FP8 MoE 层，且已通过维度检查避免对非 3D scale 的误操作。但未引入新测试，存在回归覆盖不足的风险。依赖版本升级可能引入其他变更。
- 影响：影响范围：仅 Intel XPU 平台。影响模块：FP8 MoE 层（block FP8 模式）。用户需升级 vllm_xpu_kernels 至 v0.1.9 才能使用此修复。

- 风险标记：缺少测试覆盖，依赖版本升级

关联脉络

- PR #43905 [DSv4] Move mHC tilelang kernels & Don't use CustomOP in dsv4/nvidia: 均为 MoE 相关内核重构，但影响不同平台和量化方案。
- PR #43660 [Attention][AMD] Standardize kv layout to blocks first for AMD: 类似平台特定的布局标准化变更，体现跨平台适配模式。