PR #44804 完整报告

vllm-project/vllm

[ROCm][gpt-oss] Hybrid CDNA4 swizzle gate for A8W4 MoE

合并时间：2026-06-10 14:59

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44804

执行摘要

- 一句话：CDNA4 swizzle 只在 TP<=2 时启用
- 推荐动作：值得合入，设计干净。对于关注 ROCm 性能或 MoE 量化的工程师，建议精读 should_use_cdna4_mx_scale_swizzle 的集中式门控模式——这是一个处理硬件特性与并行度交叉限制的轻量但清晰的策略。

功能与动机

在 gfx950 上，CDNA4 swizzle 要求 BLOCK_K%256==0，但 TP>=4 时每个 rank 的中间尺寸变小，A8W4 调度会选用 BK<256 的 tile，导致 swizzle 失效。硬编码 "CDNA4_SCALE" 在所有 TP 下开启，会在 TP>=4 时引发性能下降甚至可能静默数据损坏。PR body 明确说明需要一个统一门控来保证权重重排布局与内核参数一致。

实现拆解

1. 抽取门控函数：在 vllm/model_executor/layers/quantization/utils/mxfp4_utils.py 中新增 should_use_cdna4_mx_scale_swizzle()，返回 on_gfx950() and get_tensor_model_parallel_world_size() <= 2。该函数被设计为单一真相来源。
2. 改造 swizzle 决策：将 _swizzle_mxfp4() 中原有的 on_gfx950() 条件替换为调用 should_use_cdna4_mx_scale_swizzle()；当条件不满足时 fallback 到 StridedLayout。
3. 改造内核参数：在 vllm/model_executor/layers/fused_moe/experts/aiter_mxfp4_w4a8_moe.py 的 triton_kernel_fused_mxfp4_w4a8_experts 中，用 _swizzle_mx_scale 变量替代硬编码的 "CDNA4_SCALE"，该变量值为 "CDNA4_SCALE" 或 None。
4. 向后兼容：AITER 0.1.13.post1 的 moe_gemm_a8w4 默认 swizzle_mx_scale=None，因此 None 参数是安全的无操作。

关键文件：

- vllm/model_executor/layers/quantization/utils/mxfp4_utils.py（模块 量化工具；类别 source；类型 data-contract；符号 should_use_cdna4_mx_scale_swizzle, _swizzle_mxfp4）：核心变更所在。新增 should_use_cdna4_mx_scale_swizzle() 作为统一门控，并修改 _swizzle_mxfp4 中的 swizzle 决策逻辑。
- vllm/model_executor/layers/fused_moe/experts/aiter_mxfp4_w4a8_moe.py（模块 MoE 专家；类别 source；类型 data-contract；符号 triton_kernel_fused_mxfp4_w4a8_experts）：修改 MoE 内核调用点，将硬编码的 swizzle_mx_scale="CDNA4_SCALE" 改为根据

should_use_cdna4_mx_scale_swizzle() 动态选择。

关键符号：should_use_cdna4_mx_scale_swizzle, _swizzle_mxfp4, triton_kernel_fused_mxfp4_w4a8_experts

关键源码片段

vllm/model_executor/layers/quantization/utils/mxfp4_utils.py

核心变更所在。新增 should_use_cdna4_mx_scale_swizzle() 作为统一门控，并修改 _swizzle_mxfp4 中的 swizzle 决策逻辑。

```
# vllm/model_executor/layers/quantization/utils/mxfp4_utils.py

def should_use_cdna4_mx_scale_swizzle() -> bool:
    """Whether to use the CDNA4 swizzled scale layout for mxfp4 on gfx950.

    CDNA4 swizzle requires BLOCK_K%256==0; at TP>=4 the A8W4 dispatch
    picks BK<256 tiles for the smaller per-rank shapes, so swizzle must
    be off. Used by both the weight-load swizzle in `_swizzle_mxfp4` and
    the kernel-argument gate in `aiter_mxfp4_w4a8_moe`; they must agree.
    """
    from vllm.distributed import get_tensor_model_parallel_world_size
    from vllm.platforms.rocm import on_gfx950

    # 仅当硬件为 gfx950 且 TP 世界大小 <= 2 时启用 CDNA4 swizzle
    return on_gfx950() and get_tensor_model_parallel_world_size() <= 2


def _swizzle_mxfp4(quant_tensor, scale, num_warps=8):
    # ... 前置代码不变 ...
    elif current_platform.is_rocm():
        value_layout = StridedLayout
        # 用统一门控替换之前的 `if on_gfx950()`
        if should_use_cdna4_mx_scale_swizzle():
            try:
                from triton_kernels.tensor_details.layout import GFX950MXScaleLayout
                scale_layout = GFX950MXScaleLayout
            except ImportError:
                from triton_kernels.tensor_details.layout import CDNA4MXScaleLayout
                scale_layout = CDNA4MXScaleLayout
        else:
            scale_layout = StridedLayout
```

vllm/model_executor/layers/fused_moe/experts/aiter_mxfp4_w4a8_moe.py

修改 MoE 内核调用点，将硬编码的 swizzle_mx_scale="CDNA4_SCALE" 改为根据 should_use_cdna4_mx_scale_swizzle() 动态选择。

```
# vllm/model_executor/layers/fused_moe/experts/aiter_mxfp4_w4a8_moe.py

def triton_kernel_fused_mxfp4_w4a8_experts(
```

```
    # ... 参数列表不变 ...
) -> torch.Tensor:
    # ... 前置检查不变 ...
    from vllm.model_executor.layers.quantization.utils.mxfp4_utils import (
        should_use_cdna4_mx_scale_swizzle,
    )

    # 统一门控：TP<=2 时用 CDNA4_SCALE，否则为 None（回退到 strided）
    _swizzle_mx_scale = "CDNA4_SCALE" if should_use_cdna4_mx_scale_swizzle() else None

    # ... 继续之前逻辑 ...
    intermediate_cache1 = moe_gemm_a8w4(
        # ... 参数不变 ...
        swizzle_mx_scale=_swizzle_mx_scale, # 之前硬编码为 "CDNA4_SCALE"
        # ...
    )
    intermediate_cache3 = moe_gemm_a8w4(
        # ... 参数不变 ...
        swizzle_mx_scale=_swizzle_mx_scale, # 之前硬编码为 "CDNA4_SCALE"
        # ...
    )
    return intermediate_cache3
```

评论区精华

Rohan138 在评论中提供了 MI355X 上的基准数据：TP=1 时 swizzle 路径达到 221 tok/s，TP=8 时 strided 路径达到 316 tok/s，确认两个路径都正常工作。同时指出 swizzle_mx_scale=None 是 AITER 默认值，PR 可安全合并。AndreasKaratzas 请求了 force merge。

- 性能验证与基准数据 (performance): 验证通过，TP>=4 时 strided 性能更优。
- 向后兼容性 (other): 确认 PR 不依赖特定 AITER 版本，可安全合并。

风险与影响

- 风险：低风险。变更范围仅限两个文件，逻辑清晰。主要风险是门控条件 TP<=2 的硬编码边界——若未来引入其他 tile 调度策略（如动态 BK 选择），此硬限制可能过时。但当前已知所有 TP>=4 场景均受益于 strided 布局，门控是正确的。回退路径 None 是 AITER 默认值，无需依赖特定版本。
- 影响：影响范围局限于 ROCm gfx950 上使用 A8W4 MoE 的模型（如 AMD 的 GPT-OSS 120B）。TP=1/2 用户无变化；TP>=4 用户自动获得性能收益（最高 +19%），无需配置更改。代码可维护性提升：门控函数集中一处，避免两个位置未来出现分歧。
- 风险标记：核心路径变更

关联脉络

- PR #44945 [ROCm][Perf] Use fused softplus-sqrt-topk router under AITER fused-MoE: 同为 ROCm + AITER MoE 性能优化，修改同一模块（fused_moe）的不同部分（router vs experts）。