PR #47408 完整报告

vllm-project/vllm

[Kernel] Applies routed_scaling_factor internally

合并时间：2026-07-07 17:00

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47408

执行摘要

- 一句话：将 routed_scaling_factor 移入路由 kernel，消除额外乘法
- 推荐动作：建议阅读以了解如何将模型后处理逻辑下沉至 kernel 以减少 kernel 启动开销。设计模式值得借鉴：通过新增参数 apply_routed_scale_to_output 兼容不同后端（ROCm aiter vs 原生 CUDA）。值得关注后续 PR：finalize/add/ar/norm 融合。

功能与动机

PR body 说明：将 M3 和 DeepSeek V3 的 routed_scaling_factor 移入 routing kernel。好处：

1) 当 torch.compile 未启用时，消除由 routed_scaling_factor 引入的额外 mul kernel； 2) 更容易支持 finalize + add + ar + norm 融合。

实现拆解

1. 在 CUDA kernel（topk_softmax_kernels.cu）中修改 topk_sigmoid 实现，使其接受 routed_scaling_factor 参数并在 kernel 内部对 topk_weights 进行缩放。
2. 更新 kernel 声明（moe_ops.h）和 Stable Torch 绑定（torch_bindings.cpp），在 topk_sigmoid 签名中新增 double routed_scaling_factor。
3. 在 Python 层自定义操作（_custom_ops.py）中同步更新 ops.topk_sigmoid 签名。
4. 修改路由器前端（fused_topk_bias_router.py）：vllm_topk_sigmoid 函数新增 routed_scaling_factor 参数并传递至下方；同时在 fused_topk_bias 的 sigmoid 分支中移除后乘代码（if routed_scaling_factor != 1.0: topk_weights *= routed_scaling_factor）。
5. 调整模型构建：DeepseekV2MoE 类新增 apply_routed_scale_to_output 参数，由外部决定是否在 MoE 输出处应用缩放，替换原来的硬编码逻辑；DeepSeek V32 和 MiniMax M3 模型传入 apply_routed_scale_to_output=False，交由 kernel 处理。

关键文件：

- csrc/libtorch_stable/moe/topk_softmax_kernels.cu（模块 路由核；类别 source；类型 core-logic）：核心 CUDA kernel 实现，将 routed_scaling_factor 应用逻辑移入 kernel，消除一次额外乘法。
- vllm/model_executor/layers/fused_moe/router/fused_topk_bias_router.py（模块 路由器；类别 source；类型 data-contract；符号 vllm_topk_sigmoid, fused_topk_bias）：路由

器前端函数，新增参数传递并移除 Python 层后乘，将缩放下沉到 kernel。

- vllm/model_executor/models/deepseek_v2.py（模块 模型类；类别 source；类型 data-contract；符号 DeepseekV2MoE）：DeepseekV2MoE 类新增 apply_routed_scale_to_output 参数，控制是否在 MoE 输出处应用缩放。
- vllm/models/deepseek_v32/nvidia/model.py（模块 DeepSeek V32；类别 source；类型 data-contract）：DeepSeek V32 模型禁用外部缩放，完全依赖 kernel 内部缩放。
- vllm/models/minimax_m3/nvidia/model.py（模块 MiniMax M3；类别 source；类型 data-contract）：MiniMax M3 模型移除外部缩放参数，与 kernel 内部缩放一致。

关键符号：vllm_topk_sigmoid, fused_topk_bias, DeepseekV2MoE.init

关键源码片段

vllm/model_executor/layers/fused_moe/router/fused_topk_bias_router.py

路由器前端函数，新增参数传递并移除 Python 层后乘，将缩放下沉到 kernel。

```
'''
vllm/model_executor/layers/fused_moe/router/fused_topk_bias_router.py

关键变更：将 routed_scaling_factor 传入 ops.topk_sigmoid，并移除 Python 层后乘。
'''


def vllm_topk_sigmoid(
    topk_weights: torch.Tensor,
    topk_indices: torch.Tensor,
    token_expert_indices: torch.Tensor,
    gating_output: torch.Tensor,
    renormalize: bool = False,
    e_score_correction_bias: torch.Tensor | None = None,
    # NEW: scaling factor 直接传递给 CUDA kernel，由 kernel 内部应用
    routed_scaling_factor: float = 1.0,
) -> tuple[torch.Tensor, ...]:
    ops.topk_sigmoid(
        topk_weights,
        topk_indices,
        token_expert_indices,
        gating_output,
        renormalize,
        e_score_correction_bias,
        routed_scaling_factor, # 新参数，kernel 内部完成缩放
    )
    return topk_weights, topk_indices

def fused_topk_bias(...):
    ...
    if scoring_func == 'sigmoid':
        topk_weights, topk_ids = vllm_topk_sigmoid(
            topk_weights,
```

```
            topk_ids,
            token_expert_indices,
            gating_output,
            renormalize,
            e_score_correction_bias,
            routed_scaling_factor, # 新参数
        )
        # 原后乘代码已删除:
        # if routed_scaling_factor != 1.0:
        # topk_weights *= routed_scaling_factor
        return topk_weights, topk_ids
    ...
```

vllm/model_executor/models/deepseek_v2.py

DeepseekV2MoE 类新增 apply_routed_scale_to_output 参数，控制是否在 MoE 输出处应用缩放。

```
'''
vllm/model_executor/models/deepseek_v2.py

关键变更：DeepseekV2MoE 类新增 apply_routed_scale_to_output 参数，
控制是否在 MoE 输出处应用 routed_scaling_factor。
'''


class DeepseekV2MoE(nn.Module):
    def __init__(
        self,
        config: DeepseekV2Config | DeepseekV3Config,
        parallel_config: ParallelConfig,
        quant_config: QuantizationConfig | None = None,
        prefix: str = '',
        apply_routed_scale_to_output: bool = False, # NEW: 由外部控制
    ):
        ...
        self.routed_scaling_factor = getattr(config, 'routed_scaling_factor', 1.0)
        ...
        self.experts = FusedMoE(
            ...
            routed_scaling_factor=self.routed_scaling_factor,
            # 使用入参替换原来的硬编码 not self.is_rocm_aiter_moe_enabled
            apply_routed_scale_to_output=apply_routed_scale_to_output,
            ...
        )

# 在模型构建时（DeepseekV2DecoderLayer 处）：
self.mlp = DeepseekV2MoE(
    config=config,
    parallel_config=parallel_config,
    quant_config=quant_config,
```

```
        prefix=f'{prefix}.mlp',
        # aiter 内部已应用 scaling，此处关闭外部应用
        apply_routed_scale_to_output=not rocm_aiter_ops.is_fused_moe_enabled(),
    )
```

评论区精华

代码审查由 zyongye 批准，未出现设计分歧或重大争议。自动评论仅来自 claude bot 提示手动审查，无实质讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险包括： 1) 数值一致性：kernel 内部缩放与原有后乘逻辑可能因浮点运算顺序差异导致结果微小偏差，需通过端到端测试验证。 2) 后端兼容性：ROCm aiter 路径已有独立实现（内部缩放），本改动不影响；但非 aiter 路径的 sigmoid 分支现全部由 kernel 处理，可能影响 ROCm 非 aiter 用户。 3) 缺少专项测试：本次改动未附带新测试用例，仅依赖现有集成测试。 4) 模型覆盖：改动涉及 DeepSeek V2/V3 和 MiniMax M3，需确认 DeepSeek V2（原基础模型）的缩放行为完全一致。
  - 影响：影响范围：所有使用 fused_topk_bias 路由器且 scoring_func 为 sigmoid 的模型，主要是 DeepSeek V3 和 MiniMax M3（DeepSeek V2 也受影响但其缩放行为取决于 apply_routed_scale_to_output 参数）。对用户透明，预期带来微小性能提升。对团队而言，该 PR 为后续 MoE 融合优化铺平道路。
  - 风险标记：缺少测试覆盖，核心路径变更，数值一致性风险

关联脉络

- 暂无明显关联 PR