PR #43769 完整报告

vllm-project/vllm

[Bugfix] Pass `routed_scaling_factor` to FlashInfer TRTLLM BF16 MoE

合并时间：2026-05-28 12:29

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43769

执行摘要

- 一句话：修复 BF16 MoE 缺失 routed_scaling_factor 传递
- 推荐动作：此 PR 改动虽小但影响极大（正确性 bug），值得作为故障案例学习：在多层参数传递中，每个后端 / 调用点都需确保所有参数透传。相关开发者应检查其他 MoE 后端是否也存在类似遗漏。

功能与动机

routed_scaling_factor 是 MoE 层路由权重缩放的关键配置，但 FlashInfer TRTLLM BF16 MoE 后端在调用底层 kernel 时遗漏了该参数的传递，导致所有使用该后端且配置了 routed_scaling_factor 的模型（如 Kimi-Linear-48B-A3B-Instruct）推理结果严重错误。PR body 中指出该 bug 影响了 apply_routed_scale_to_output=False 的 Blackwell 环境。

实现拆解

1. 修复 kernel 调用参数传递：在 vllm/model_executor/layers/fused_moe/experts/trtllm_bf16_moe.py 的 apply 方法中，向 flashinfer.fused_moe.trtllm_bf16_moe 新增 routed_scaling_factor=routed_scaling_factor 参数，确保 scaling 值被正确传入底层 kernel。
2. 更新测试验证：在 tests/kernels/moe/test_moe.py 的 test_unquantized_bf16_flashinfer_trtllm_backend 测试中，将 layer.routed_scaling_factor 从 None 改为 2.446（与配置一致），并在 baseline 输出中乘以该 scaling factor，使测试能够正确验证修复后的行为。
3. GSM8K 基准验证：在 Kimi-Linear-48B-A3B-Instruct 和 Trinity-Mini 两个模型上对比修复前后准确率，修复前 TRTLLM 后端准确率仅 0.378/0.151，修复后提升至 0.886/0.885，与 Triton 后端基线一致。

关键文件：

- vllm/model_executor/layers/fused_moe/experts/trtllm_bf16_moe.py（模块 MoE 算子；类别 source；类型 data-contract；符号 apply）：核心修复文件：在 kernel 调用调用中添加缺失的 routed_scaling_factor 参数传递，一行变更直接修复正确性 bug。
- tests/kernels/moe/test_moe.py（模块 MoE 测试；类别 test；类型 test-coverage；符号 test_unquantized_bf16_flashinfer_trtllm_backend）：测试验证文件：更新 unquantized BF16 TRTLLM 测试以设置非 None 的 routed_scaling_factor，并在 baseline 计算中乘以

该因子，确保测试能 catch 类似遗漏。

关键符号：apply

关键源码片段

vllm/model_executor/layers/fused_moe/experts/trtllm_bf16_moe.py

核心修复文件：在 kernel 调用调用中添加缺失的 routed_scaling_factor 参数传递，一行变更直接修复正确性 bug。

```
# vllm/model_executor/layers/fused_moe/experts/trtllm_bf16_moe.py
# 在 apply 方法的 kernel 调用中添加 routed_scaling_factor 参数
return flashinfer.fused_moe.trtllm_bf16_moe(
    routing_logits=router_logits,
    routing_bias=e_score_correction_bias,
    hidden_states=hidden_states,
    gemm1_weights=w1,
    gemm2_weights=w2,
    num_experts=global_num_experts,
    top_k=self.topk,
    n_group=num_expert_group,
    topk_group=topk_group,
    intermediate_size=self.intermediate_size_per_partition,
    local_expert_offset=self.ep_rank * self.local_num_experts,
    local_num_experts=self.local_num_experts,
    routed_scaling_factor=routed_scaling_factor, # 新增：将 scaling factor 传入 kernel
    routing_method_type=self.routing_method_type,
)
```

tests/kernels/moe/test_moe.py

测试验证文件：更新 unquantized BF16 TRTLLM 测试以设置非 None 的 routed_scaling_factor，并在 baseline 计算中乘以该因子，确保测试能 catch 类似遗漏。

```
# tests/kernels/moe/test_moe.py 中 test_unquantized_bf16_flashinfer_trtllm_backend 的修改
# 将 routed_scaling_factor 设为非 None 的配置值
layer.routed_scaling_factor = 2.446 # 之前为 None，导致测试无法暴露缺少 scaling 的 bug

# 在计算 torch baseline 时，必须乘以 scaling factor 以匹配 kernel 行为
baseline_output = (
    torch_moe(a, w1_original, w2_original, router_logits, topk)
    * layer.routed_scaling_factor # 新增：确保 baseline 包含 scaling
)
```

评论区精华

讨论较少，仅有一位 reviewer 直接 approve。PR 本身改动清晰明确，未出现大的设计争议。

- 暂无高价值评论线程

风险与影响

- 风险：修复仅补传了一个已在函数签名中存在的参数，回归风险极低。唯一潜在风险是 TRTLLM BF16 kernel 内部对 routed_scaling_factor 的实现是否符合预期，但 GSM8K 验证结果已基本确认其正确性。
- 影响：直接影响所有使用 FlashInfer TRTLLM BF16 MoE 后端的模型，尤其是配置了 routed_scaling_factor 且 apply_routed_scale_to_output=False 的场景。修复后推理正确性得到保障，准确率从接近随机提升至正常水平，对用户影响显著。
- 风险标记：暂无

关联脉络

- PR #43599 [Bugfix][Kernel] TRTLLM NVFP4 MoE chunking: 同样是修复 TRTLLM MoE kernel 的 bug，涉及相同的后端文件集合和测试文件，体现持续优化 TRTLLM MoE 后端的趋势。