PR #45707 完整报告

vllm-project/vllm

[Bugfix][MoE] Restore routed output unpadding before shared expert add

合并时间：2026-06-16 21:06

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45707

执行摘要

- 一句话：修复 MoE 路由输出在共享专家加前未反填充的回归
- 推荐动作：该 PR 值得精读，尤其是在使用 MoE 模型和 flashinfer_trtllm 后端的场景。其设计决策体现了对张量内存布局和量化后填充的细致考虑。建议后续关注类似的条件判断谨慎避免 AND/OR 混淆。

功能与动机

PR body 明确指出，该修复针对 Nemotron-3 Nano 系列模型在 FLASHINFER_TRTLLM MoE 后端上的回归问题。回归由 PR #41184 引入，而本应捕获该问题的测试在 #39510 中被跳过且未在 TRTLLM NVFP4 启用时重新启用。提交者提供了详细的错误日志：RuntimeError('The size of tensor a (2688) must match the size of tensor b (2816) at non-singleton dimension 1')。

实现拆解

修复涉及两个文件：

1. vllm/model_executor/layers/fused_moe/runner/moe_runner.py中的 _maybe_pad_hidden_states 方法：
   - 关键改动是将条件从 and（老代码中的 and）恢复为 or：当存在 routed_output_transform 或存在共享专家时，都设置 pre_xform_trunc_size。同时，post_xform_trunc_size 的逻辑被拆分为两步：先设为 transformed_hidden_dim，若同时有 transform 和共享专家，则改为 shared_experts_hidden_dim。这确保了路由输出在加共享专家前被正确截断到 unpadded size（2688），而最终输出则根据是否有 transform 决定使用 shared_experts_hidden_dim。
2. tests/quantization/test_blackwell_moe.py：
   - 移除了 test_nemotron_fp4_moe_flashinfer_throughput 和 test_nemotron_fp4_moe_flashinfer_latency 两个测试上的 @pytest.mark.skip 装饰器，因为 TRTLLM NVFP4 后端现在已支持该部署配置。
   - 将测试函数重命名：test_nemotron_fp4_moe_flashinfer_throughput -> test_nemotron_fp4_moe_flashinfer_cutlass，test_nemotron_fp4_moe_flashinfer_latency -> test_nemotron_fp4_moe_flashinfer_trtllm，使名称更准确反映测试后端。

关键文件：

- vllm/model_executor/layers/fused_moe/runner/moe_runner.py（模块 MoE 运行器；类别 source；类型 data-contract；符号 _maybe_pad_hidden_states）：核心修复文件，修改 _maybe_pad_hidden_states 方法中的截断条件逻辑。
- tests/quantization/test_blackwell_moe.py（模块 Blackwell MoE；类别 test；类型 test-coverage；符号 test_nemotron_fp4_moe_flashinfer_throughput, test_nemotron_fp4_moe_flashinfer_cutlass, test_nemotron_fp4_moe_flashinfer_latency, test_nemotron_fp4_moe_flashinfer_trtllm）：测试配套：移除跳过标记并重命名测试，确保回归测试覆盖。

关键符号：_maybe_pad_hidden_states

关键源码片段

vllm/model_executor/layers/fused_moe/runner/moe_runner.py

核心修复文件，修改 _maybe_pad_hidden_states 方法中的截断条件逻辑。

```
def _maybe_pad_hidden_states(
    self,
    hidden_states: torch.Tensor,
    shared_experts_input: torch.Tensor | None,
) -> tuple[torch.Tensor, int | None, int | None]:
    """
    Pad hidden_states 至 moe_config.hidden_dim 并记录原始尺寸以便后续截断。
    对于 latent MoE，路由 hidden_states 可能小于 hidden_dim，填充后确保
    融合 MoE 核中张量大小一致。返回的 trunc_size 用于 _maybe_reduce_final_output
    移除填充。
    """
    shared_experts_hidden_dim = (
        shared_experts_input.shape[-1] if shared_experts_input is not None else 0
    )
    transformed_hidden_dim: int | None = hidden_states.shape[-1]
    if (
        not self._quant_method.skip_forward_padding
        and self.moe_config.hidden_dim != transformed_hidden_dim
    ):
        assert transformed_hidden_dim is not None
        hidden_states = F.pad(
            hidden_states,
            (0, self.moe_config.hidden_dim - transformed_hidden_dim),
            mode="constant",
            value=0.0,
        )

    # 截断尺寸：None 表示无需截断。
    # 前向过程中有两个截断点：
    # pre_xform：在 routed_output_transform 之前截断融合输出
    # post_xform：在 all-reduce 之后对最终结果截断
```

```
    #
    # 对于有 routed transform 或 shared experts 的 MoE（如 Nemotron-3 Nano）：
    # - 若 transform 需要 unpadded 路由输出，或 shared+routed 相加需要匹配 hidden dim，则设
    pre_xform
    # - 当 transform 和 shared experts 同时存在时，post_xform 使用 shared_experts_hidden_dim
    # 否则使用 transformed_hidden_dim
    if transformed_hidden_dim == hidden_states.shape[-1]:
        transformed_hidden_dim = None

    pre_xform_trunc_size = None
    # 关键修复：从 AND 恢复为 OR，确保只要存在 transform 或 shared experts 就进行 pre_xform
    截断
    if self.routed_output_transform is not None or shared_experts_hidden_dim > 0:
        pre_xform_trunc_size = transformed_hidden_dim
    post_xform_trunc_size = transformed_hidden_dim
    if self.routed_output_transform is not None and shared_experts_hidden_dim > 0:
        post_xform_trunc_size = shared_experts_hidden_dim

    return hidden_states, pre_xform_trunc_size, post_xform_trunc_size
```

tests/quantization/test_blackwell_moe.py

测试配套：移除跳过标记并重命名测试，确保回归测试覆盖。

```
# 原测试被跳过，现取消跳过并重命名以准确反映后端
def test_nemotron_fp4_moe_flashinfer_cutlass(monkeypatch: pytest.MonkeyPatch):
    # 测试 CUTLASS 后端初始化，不涉及 padding
    can_initialize(
        "nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-NVFP4",
        hf_overrides=HF_OVERRIDE_TEXT,
        extra_args=["--moe-backend=flashinfer_cutlass"],
    )

# 原测试因 "hidden_dim % 512 != 0" 被跳过，现在 TRTLLM 后端已修复
def test_nemotron_fp4_moe_flashinfer_trtllm(monkeypatch: pytest.MonkeyPatch):
    # 测试 TRTLLM 后端初始化，该场景触发 padding，回归前会崩溃
    can_initialize(
        "nvidia/NVIDIA-Nemotron-3-Nano-30B-A3B-NVFP4",
        hf_overrides=HF_OVERRIDE_TEXT,
        extra_args=["--moe-backend=flashinfer_trtllm"],
    )
```

评论区精华

该 PR 的审核评论较少，两位审核人（bnellnm 和 tomeras91）均直接批准，且未作详细评论。PR 提交者在 issue 评论中标记了相关审核人，解释这是 #41184 引入的回归。无实质性讨论。

- 暂无高价值评论线程

风险与影响

- 风险：修复本身改动范围小（仅修改一个条件运算符和截断逻辑），且已通过取消跳过的测试验证，风险较低。但需注意：
  - 条件从 AND 改回 OR 可能影响其他 latent MoE 模型（如 NemotronH）的行为，需确保已有测试覆盖。
  - 新的 post_xform_trunc_size 逻辑拆分为两步，可能影响没有 transform 但只有共享专家的模型，需确认。建议运行全量 MoE 测试套件。
  - 影响：直接影响 Nemotron-3 Nano 系列模型（NVFP4 量化和 FLASHINFER_TRTLLM 后端），修复了此前无法启动的 RuntimeError。对其他 MoE 模型（如 GPT-OSS、Mixtral）无影响，因为它们不涉及 shared experts + padding 组合。团队维护成本低，CI 中重新启用测试将提高该场景的回归防护能力。
  - 风险标记：核心路径变更，回归风险

关联脉络

- PR #41184 Introduced the regression by erroneously changing OR to AND in _maybe_pad_hidden_states: 该 PR 是回归引入的源头，条件从 OR 改成了 AND。
- PR #39510 Skipped the test that would have caught the regression: 该 PR 跳过了本应捕获此回归的测试，且未在 TRTLLM NVFP4 启用时恢复。