PR #39391 完整报告

vllm-project/vllm

fix: clamp NaN/Inf in topk_softmax to prevent duplicate expert IDs

合并时间：2026-04-21 19:04

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39391

执行摘要

- 一句话：修复 MoE topk_softmax 中 NaN/Inf 处理，防止 CUDA 图下生成重复专家 ID 导致的非法内存访问。
- 推荐动作：建议精读此 PR，了解如何处理数值异常情况，以及对 MoE 路由和 CUDA 图集成的设计权衡。

功能与动机

修复 issue #39244 中描述的 CUDA 非法内存访问崩溃。根本原因是 CUDA 图 replay 中填充 token 的隐藏状态退化产生 NaN 门控 logits，导致 topk_softmax 生成重复专家 ID，进而触发 FlashInfer MoE 排序 kernel 的 bug。

实现拆解

1. 核心 kernel 修复：在 csrc/moe/topk_softmax_kernels.cu 的 topkGatingSoftmax warp kernel 中添加 NaN/Inf clamp，将 NaN/Inf 值置为 0，防止 argmax 循环始终选择专家 0。
2. 扩展至备用路径：同样修改 moeSoftmax 和 moeSigmoid kernel，覆盖专家数量非标准（非 2 的幂或 64 的倍数）的情况，确保全面修复。
3. 添加回归测试：在 tests/kernels/moe/test_fused_topk.py 中新增 test_fused_topk_nan_inf_clamp 测试，参数化覆盖多种数据类型、评分函数和坏值，验证 clamp 效果和专家 ID 唯一性。
4. 性能验证：通过微基准测试和端到端测试确认修复无性能开销，并解决高并发下的崩溃问题。

关键文件：

- csrc/moe/topk_softmax_kernels.cu（模块 MoE 内核；类别 source；类型 core-logic；符号 topkGatingSoftmax, moeSoftmax, moeSigmoid）：核心 kernel 文件，添加 NaN/Inf clamp 逻辑，防止重复专家 ID 生成。
- tests/kernels/moe/test_fused_topk.py（模块 融合算子测试；类别 test；类型 test-coverage；符号 test_fused_topk_nan_inf_clamp）：新增回归测试，验证 NaN/Inf clamp 在不同参数组合下的正确性。

关键符号：topkGatingSoftmax, moeSoftmax, moeSigmoid, test_fused_topk_nan_inf_clamp

关键源码片段

tests/kernels/moe/test_fused_topk.py

新增回归测试，验证 NaN/Inf clamp 在不同参数组合下的正确性。

```
# 回归测试：验证 NaN/Inf clamp 在 topk_softmax kernel 中的效果
def test_fused_topk_nan_inf_clamp(
    num_experts: int,
    topk: int,
    scoring_func: str,
    bad_value: float, # 坏值可以是 NaN 或 Inf
    dtype: torch.dtype,
):
    """
    模拟填充token产生的NaN/Inf门控输出，验证clamp后专家ID唯一且权重有限。
    """
    # 创建部分包含坏值的 gating_output
    gating_output = torch.randn((num_tokens, num_experts), dtype=dtype, device="cuda")
    gating_output[1:, :] = bad_value # 第 2 行及之后设为 NaN 或 Inf

    # 调用修复后的 fused_topk kernel
    topk_weights, topk_ids, _ = fused_topk(
        hidden_states=hidden_states,
        gating_output=gating_output,
        topk=topk,
        renormalize=False,
        scoring_func=scoring_func,
    )

    # 验证：正常行与参考一致，坏值行专家 ID 必须唯一
    for row in range(1, num_tokens):
        row_ids = topk_ids[row]
        assert row_ids.unique().numel() == topk, f"Row {row} has duplicate expert IDs"
        assert torch.isfinite(topk_weights[row]).all(), f"Row {row} has non-finite weights"
```

评论区精华

review 中，gemini-code-assist[bot] 指出备用路径也需修复，作者确认已包含；ZJY0516 询问 clamp 的必要性，作者解释专家数量非标准时使用备用路径；tlrmchlsmth 建议使用 torch.nan_to_num，但 PR 选择了直接 clamp 以保持低开销。

- 备用路径修复 (design): 作者确认已包含备用路径的修复，确保全面性。
- clamp 实现选择 (design): PR 选择直接 clamp 以保持低性能开销，且 kernel 级修复更直接。

风险与影响

- 风险：风险低：clamp 逻辑只在输入为 NaN/Inf 时生效，正常输入无影响；性能开销可忽略。但需确保所有 kernel 路径都已覆盖，防止遗漏导致类似问题。
- 影响：影响使用 MoE 模型（如 Qwen3.5-397B）和 CUDA 图的用户，特别是在高并发场景。修复后能避免 CUDA 非法内存访问崩溃，提高服务稳定性和可靠性。

- 风险标记：核心路径变更，数值稳定性

关联脉络

- 暂无明显关联 PR