PR #25285 完整报告

sgl-project/sglang

Fix EPLB mapping for TopK paths

合并时间：2026-05-18 14:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25285

执行摘要

- 一句话：修复 EPLB TopK 路径中逻辑 / 物理专家 ID 映射错误
- 推荐动作：值得其他开发者了解 EPLB 中逻辑 / 物理 ID 映射的设计模式，尤其是如何避免重复映射。新增测试可作为类似场景的参考。代码简洁，可作快速阅读。

功能与动机

EPLB（Expert Parallel Load Balancing）在 TopK 路径中存在两处专家 ID 映射错误：

1) biased_topk_impl 和 biased_topk_jit_kernel_impl 内部已经调用了 topk_ids_logical_to_physical，而后续 _post_process_topk_ids() 又会再次映射，导致双重重映射； 2) 当 DeepEP 共享专家融合与 EPLB 一起运行时，物理专家数需要从 ExpertLocationDispatchInfo 获取，而非 router_logits.shape[1]，避免使用错误的专家数量。PR body 中提到 "keep biased_topk_impl and biased_topk_jit_kernel_impl returning logical expert ids, then let _post_process_topk_ids() apply EPLB mapping exactly once"。

实现拆解

变更涉及两个文件，核心改动是在 topk.py 中去掉内部映射，并在 test_topk.py 中增加验证。具体步骤如下：

1. 移除内部映射：在 python/sglang/srt/layers/moe/topk.py 的 biased_topk_impl（第 862 行附近）和 biased_topk_jit_kernel_impl（第 850 行附近）中，删除 topk_ids_logical_to_physical(topk_ids, expert_location_dispatch_info) 和 _mask_topk_ids_padded_region(topk_ids, num_token_non_padded) 两行调用。从当前 head_excerpt 可以看到这两个调用已被移除，函数直接 return topk_weights, topk_ids。这样 biased_topk_impl 和 biased_topk_jit_kernel_impl 返回的是逻辑专家 ID，后续由 _post_process_topk_ids() 统一进行物理映射，确保映射只执行一次。
2. 新增测试类：在 test/registered/cpu/test_topk.py 中新增 TestBiasedTopK 测试类，包含 test_biased_topk_returns_logical_ids_with_eplb_info 方法。该方法构造一个简单的输入（1 个 token，4 个 expert），使用 ExpertLocationDispatchInfo 提供静态 EPLB 映射（逻辑 expert 0→物理 2，逻辑 1→物理 3，逻辑 2→物理 0，逻辑 3→物理 1），调用 biased_topk_impl（即 native_biased_topk）并断言返回的 topk_ids 是逻辑 ID [0, 1]。测试验证了修改的正确性。
3. 导入调整：在新测试中引入了 ExpertLocationDispatchInfo 和 biased_topk_impl 的导入。

PR body 还提及第二个修复涉及 ExpertLocationDispatchInfo.num_physical_experts 用于 DeepEP 共享专家 remap，但该改动可能在其他文件中，当前变更文件未直接体现。

关键文件：

- python/sglang/srt/layers/moe/topk.py（模块 TopK；类别 source；类型 core-logic；符号 biased_topk_impl, biased_topk_jit_kernel_impl）：核心源码修改：移除 biased_topk_impl 和 biased_topk_jit_kernel_impl 内部的逻辑到物理 ID 映射及 padding 掩码，使其返回逻辑 ID。
- test/registered/cpu/test_topk.py（模块 TopK；类别 test；类型 test-coverage；符号 TestBiasedTopK, test_biased_topk_returns_logical_ids_with_eplb_info）：新增测试用例，验证 biased_topk 在 EPLB 场景下返回逻辑 ID，确保修复正确性。

关键符号：biased_topk_impl, biased_topk_jit_kernel_impl

关键源码片段

python/sglang/srt/layers/moe/topk.py

核心源码修改：移除 biased_topk_impl 和 biased_topk_jit_kernel_impl 内部的逻辑到物理 ID 映射及 padding 掩码，使其返回逻辑 ID。

```
# python/sglang/srt/layers/moe/topk.py ( 精简后 )

def biased_topk_impl(...):
    # ... 计算 scores, topk_ids ...
    topk_weights, topk_ids = topk_weights.to(torch.float32), topk_ids.to(torch.int32)
    # 不再在此处映射到物理 ID，让调用方统一处理
    # topk_ids = topk_ids_logical_to_physical(topk_ids, expert_location_dispatch_info)
    # _mask_topk_ids_padded_region(topk_ids, num_token_non_padded)
    return topk_weights, topk_ids

def biased_topk_jit_kernel_impl(...):
    # ... 调用 moe_fused_gate ...
    topk_weights, topk_ids = topk_weights.to(torch.float32), topk_ids.to(torch.int32)
    # 同样移除映射
    return topk_weights, topk_ids
```

test/registered/cpu/test_topk.py

新增测试用例，验证 biased_topk 在 EPLB 场景下返回逻辑 ID，确保修复正确性。

```
# test/registered/cpu/test_topk.py ( 新增片段 )
class TestBiasedTopK(CustomTestCase):
    def test_biased_topk_returns_logical_ids_with_eplb_info(self):
        hidden_states = torch.ones(1, 4) # 1 个 token, 4 个 expert
        gating_output = torch.tensor([[10.0, 9.0, 1.0, 0.0]])
        correction_bias = torch.zeros(4)
        # 构造 EPLB 静态映射：逻辑 expert 0-> 物理 2, 1->3, 2->0, 3->1
        dispatch_info = ExpertLocationDispatchInfo(
            ep_dispatch_algorithm="static",
```

```
        partial_logical_to_rank_dispatch_physical_map=torch.tensor(
            [2, 3, 0, 1], dtype=torch.int64
        ),
        partial_logical_to_all_physical_map=torch.tensor(
            [[2], [3], [0], [1]], dtype=torch.int64
        ),
        partial_logical_to_all_physical_map_num_valid=torch.ones(4, dtype=torch.int64),
        num_physical_experts=4,
    )
    _, topk_ids = native_biased_topk(
        hidden_states=hidden_states,
        gating_output=gating_output,
        correction_bias=correction_bias,
        topk=2,
        renormalize=False,
        scoring_func="sqrtsoftplus",
        expert_location_dispatch_info=dispatch_info,
    )
    # 期望返回逻辑 ID [0, 1]（最高分的两个逻辑 expert），而不是物理 ID [2, 3]
    torch.testing.assert_close(topk_ids, torch.tensor([[0, 1]], dtype=torch.int32))
```

评论区精华

唯一的 review 评论来自 gemini-code-assist[bot]，指出新测试断言可能不稳定，因为 torch.topk 默认 sorted=False 时不保证输出索引的顺序，建议使用 sort(dim=-1) 后再比较。该评论被标记为 medium 优先级，但作者未采纳（未在后续提交中修改），且 PR 已被合并。未解决此问题。

- 新测试断言可能存在 flakiness (correctness): 作者未修改，但 PR 已合并。由于输入分数差异大（10 和 9 vs 1 和 0），实际不会 flaky。

风险与影响

- 风险：低风险。改动集中且明确：仅移除两个函数内部的重映射逻辑，将职责后移到统一处理点 _post_process_topk_ids()。新增测试覆盖了核心路径。潜在风险是其他调用方依赖 biased_topk_impl 直接返回物理 ID，但根据 commit message 和 PR body，EPLB 路径的设计本就是让 _post_process_topk_ids() 统一映射，因此回归可能性低。测试断言未排序可能导致偶发失败，但实际输入已保证 top-2 分数顺序明确（10 和 9 远大于 1 和 0），在 sorted=False 时依然稳定。
- 影响：影响范围限于使用 EPLB 的 TopK 路径（具体是 biased_topk_impl 和 biased_topk_jit_kernel_impl）。用户无感知，但内部逻辑正确性提升。对系统运行无负面影响。团队方面，该修复为后续 EPLB 相关改进扫清障碍。影响程度低。
- 风险标记：测试稳定性警告未解决

关联脉络

- PR #22822 [Refactor] Refactor DeepEP dispatcher: 都涉及 EPLB/DeepEP 相关映射逻辑