PR #36515 完整报告

sgl-project/sglang

[AMD] fix: do not emit a shared-expert marker twice on the per-rank slot path

合并时间：2026-08-30 09:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/36515

执行摘要

- 一句话：修复 per-rank MoE 共享专家标记重复发射，杜绝 top-k 缩水与 id 越界
- 推荐动作：值得精读。这是一个典型的 " 静默缺陷因果链分析 " 范例：PR body 把两个无报错缺陷的完整因果链（槽位被占 -> top-k 缩水；id 越界 -> remap 后出界）讲得非常清晰，并诚实评估了端到端 benchmark 的统计分辨力。对于维护 MoE 路由代码的工程师，建议关注 select_experts 中多调用点参数一致性的设计方式，以及后续补充一个构造性单元测试来锁定 top-k 语义与 id 边界。

功能与动机

PR body 开宗明义：在 per-rank fused shared-slot 路径上，共享专家由 fused_append_remap_shared_experts_deepep 在 gate 之后追加，但 select_experts 也把 num_fused_shared_experts 传给 gate，于是 both of them emit a shared marker，产生两个都不报错的缺陷：gate 被要求 K_routed = top_k - num_fused_shared_experts 个槽位后自己又占一个写标记，top-6 路由静默变成 top-5；标记写在 id num_experts，被 DeepEP per-rank remap 后移到专家空间末尾之外（EP8 下 384 -> 392，合法 id 为 0..391）。这两个缺陷无声无息，端到端精度测试无法区分，因此需要从路由语义本身修复。

实现拆解

1. 根因定位：python/sglang/srt/layers/moe/topk.py 的 select_experts 中，num_routed_topk = top_k - num_fused_shared_experts 已从 gate 的请求槽位中扣除共享专家数，但 _biased_topk 与 fused_topk 两个 gate 调用仍把 num_fused_shared_experts 原值传入。这意味着 gate 输出中会自带一个共享标记（占用 1 个槽位 + 1 个 id），而 per-rank 路径上的 fused_append_remap_shared_experts_deepep 随后又会追加一次共享专家，同一槽位出现两个生产者。
2. 新增修正值：在 select_experts 中引入局部变量 num_fused_shared_experts_for_gate，用 has_per_rank_fused_shared_slots(num_fused_shared_experts) 判定当前是否处于 per-rank shared-slot 路径；成立时对 gate 传 0，让 gate 只负责 routed 选择，共享槽位由 append 步骤唯一生产；否则保持原值，对 grouped_topk / biased_grouped_topk / fused_topk_native / custom_routing_function 等其它路径完全无影响。
3. 替换调用点：把 _biased_topk（JIT sqrtsoftplus/sigmoid 路径）与 fused_topk（默认路径）两处的 num_fused_shared_experts 参数替换为 num_fused_shared_experts_for_gate。grouped_topk / biased_grouped_topk 分支未改

动，因为这两个内核内部自带共享专家处理逻辑，不属于本次双重发射范围。

4. 验证与配套：作者在 MI355X (gfx950) 上以 DeepSeek-V4-Pro 运行 TP8 + DP8 attention + EP8/MoRI + --enforce-shared-experts-fusion，GSM8K 1319 题双 arm 对比（每个 arm 测量前先断言自身处于修复哪一侧），结果 0.945 vs 0.948。作者明确表示 4 题差距在噪声范围内，GSM8K 无法裁决该 bug，报告目的只是证明不回归。PR 未附带单元测试；CI 方面 PR Test (Base) 通过，PR Test (Extra) 与 AMD ROCm 7.2 网格曾显示失败，HaiShaw 用 /tag-and-rerun-ci 触发过重跑后合入。

关键文件：

- python/sglang/srt/layers/moe/topk.py（模块 MoE 路由；类别 source；类型 core-logic；符号 select_experts）：唯一变更文件。select_experts 是 MoE 路由核心入口，本次引入 num_fused_shared_experts_for_gate 修正 gate 收到的共享专家数，修复 per-rank fused shared-slot 路径的共享标记双重发射，同时避免路由槽位浪费与专家 id 越界。

关键符号：select_experts

关键源码片段

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

唯一变更文件。select_experts 是 MoE 路由核心入口，本次引入 num_fused_shared_experts_for_gate 修正 gate 收到的共享专家数，修复 per-rank fused shared-slot 路径的共享标记双重发射，同时避免路由槽位浪费与专家 id 越界。

```
# select_experts 是 MoE 路由的统一入口。对 DeepSeek 系列模型，top_k 需先扣除
# fused shared expert 的数量，得到真正的 routed topk 规模。
num_routed_topk = top_k - num_fused_shared_experts

# 修复点：在 per-rank fused shared-slot 路径上，共享专家的槽位由
# fused_append_remap_shared_experts_deepep 在 gate 之后统一追加。
# 若此时还把 num_fused_shared_experts 传给 gate，会出现两个静默错误：
# 1. gate 被要求输出 K_routed 个槽位，却用其中 1 个写自己的共享标记，
# 导致 top-6 路由实际只保留 5 个 routed expert；
# 2. 该标记被写在 id = num_experts 处，经 DeepEP per-rank remap 后
# 越过专家空间末尾——384 个 routed experts 在 EP8 下变成 392，
# 而合法 id 范围是 0..391。
# 因此这里将传给 gate 的共享专家数清零，让 append 成为共享槽位的
# 唯一生产者；非 per-rank 路径保持原值，行为完全不变。
num_fused_shared_experts_for_gate = (
    0
    if has_per_rank_fused_shared_slots(num_fused_shared_experts)
    else num_fused_shared_experts
)

# 所有 gate 调用统一改用修正值，例如默认的 fused_topk 路径：
topk_weights, topk_ids = fused_topk(
    hidden_states=hidden_states,
    gating_output=router_logits,
    topk=num_routed_topk if _use_aiter else top_k,
```

```
        renormalize=renormalize,
        correction_bias=correction_bias,
        scoring_func=scoring_func,
        num_fused_shared_experts=num_fused_shared_experts_for_gate,
        routed_scaling_factor=routed_scaling_factor,
        apply_routed_scaling_factor_on_output=apply_routed_scaling_factor_on_output,
    )
```

评论区精华

本 PR 没有实质性的 review 讨论线程：HaiShaw 直接 APPROVED，仅通过 issue 评论 /tag-and-rerun-ci 触发 CI 重跑。最有价值的观点来自作者在 PR body 中的自我评估：GSM8K 的 0.945 vs 0.948 差距在 run-to-run 噪声内，不能作为修复有效的证据，只能证明不回归；真正的论据是路由语义本身的两个静默缺陷——top-6 实际跑成 top-5、专家 id 写到 id 空间末尾之外。这种对 benchmark 统计分辨力的清醒认识值得借鉴。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 覆盖不完整：修复只覆盖 _biased_topk 与 fused_topk 两个调用点，grouped_topk / biased_grouped_topk 分支仍直接传 num_fused_shared_experts 原值。若未来 per-rank shared-slot 模式与 grouped_topk 组合，仍可能复现双重发射，需要确认这两个内核是否各自处理共享标记。
  2. 缺少测试覆盖：top-k 语义（恰好保留 K_routed 个 routed expert）与 id 边界（最终 id 落在 0..391）都可以用构造性小规模用例直接断言，比 GSM8K 端到端精度更能锁定回归，但 PR 未附带任何单元测试。
  3. 判定一致性依赖：修复正确性依赖 has_per_rank_fused_shared_slots 与 fused_append_remap_shared_experts_deepep 实际执行路径的一致性；若判定失真，会出现 gate 少选（判 True 但未 append）或多选（判 False 但实际 append）专家。
  4. CI 信号：PR Test (Extra) 与 AMD ROCm 7.2 网格曾失败，仓库未留下失败原因，合并前 HaiShaw 重跑过 CI，需留意是否掩盖了潜在的平台回归。 - 影响：影响面严格限定在 AMD/DeepEP 的 --enforce-shared-experts-fusion per-rank 配置下（DeepSeek-V4 系列），非 per-rank 路径是纯 no-op。收益有两个：一是路由质量恢复，top-6 不再静默缩水成 top-5；二是消除越界专家 id（EP8 下 392）对后续 gather / remap 阶段的潜在内存安全风险。对团队而言这是单文件 14 行的小改动，但揭示了一个设计要点：fused shared expert 存在两条生产路径（gate 内嵌标记 vs 外部 append 追加），调用方必须保证两者互斥，否则会出现静默语义错误。 - 风险标记：核心路由逻辑变更，缺少测试覆盖，特定硬件路径（AMD/DeepEP），gate 与 append 职责依赖判定

关联脉络

- 暂无明显关联 PR