PR #31280 完整报告

sgl-project/sglang

[NPU] Acc fix for afmoe model introduced by topk refactor.

合并时间：2026-07-27 15:40

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/31280

执行摘要

- 一句话：修复 NPU 上 AfmoE topk 重构导致的精度回退
- 推荐动作：建议合并。该 PR 修复了由重构引起的精度退化，改动量小且聚焦于 NPU 特判逻辑，讨论中已充分验证。但建议后续增加针对 NPU 下 AfmoE topk 参数传递的单元测试，防止类似回归。

功能与动机

PR #29909 引入了精度问题。在 NPU 上，旧版 topk 判断逻辑为 Trinity 模型设置了 renormalize = False 并在算子内乘以 routed_scaling_factor；而重构后，apply_routed_scaling_factor_on_output 默认初始化为 False，导致该模型不再乘以 routed_scaling_factor，造成精度下降。PR 作者在 body 中指出“Before with old version topk judgement, for this model, I set renormalize = False to multiply routed_scaling_factor in ops. After this pr's change, trinity model could not * routed_scaling_factor”。

实现拆解

1. 恢复 NPU 的 renormalize 逻辑：将 renormalize 赋值从 self.route_norm if self.score_func == "sigmoid" and not _is_npu else False 简化为 self.route_norm if self.score_func == "sigmoid" else False，移除 _is_npu 特判，使 NPU 重新进入 rerormalize=False 路径，与旧版行为一致。
2. 显式传递 apply_routed_scaling_factor_on_output：在构建 TopK 时，NPU 分支额外传递 "apply_routed_scaling_factor_on_output": True，确保 routed_scaling_factor 在输出阶段被应用，从而正确缩放路由结果。
3. 调整 NPU 关键字参数结构：将 TopK 的 **{} 字典从只传 scoring_func 改为包含 scoring_func 和 apply_routed_scaling_factor_on_output，保证 NPU 获得完整配置。
4. 保持 GPU 路径不变：GPU 分支仍使用空字典，不受影响。

关键文件：

- python/sglang/srt/models/afmoe.py（模块 MoE 路由；类别 source；类型 data-contract）：包含所有修复逻辑：恢复 renormalize 计算、为 NPU 传递 apply_routed_scaling_factor_on_output。

关键符号：未识别

关键源码片段

python/sglang/srt/models/afmoe.py

包含所有修复逻辑：恢复 renormalize 计算、为 NPU 传递 apply_routed_scaling_factor_on_output。

```
# python/sglang/srt/models/afmoe.py
# 关键片段：AfmoeMoE.__init__ 中 TopK 构建逻辑

# 第 236 行：恢复 renormalize 计算，移除 _is_npu 特判
renormalize = self.route_norm if self.score_func == "sigmoid" else False

# 第 237-254 行：TopK 初始化，NPU 分支显式传递 apply_routed_scaling_factor_on_output
self.topk = TopK(
    top_k=self.top_k,
    renormalize=renormalize,
    use_grouped_topk=self.use_grouped_topk,
    num_expert_group=self.n_group if self.use_grouped_topk else None,
    topk_group=self.topk_group if self.use_grouped_topk else None,
    custom_routing_function=custom_routing_fn,
    correction_bias=correction_bias,
    routed_scaling_factor=self.route_scale,
    # NPU 专用参数：传递 scoring_func 并强制在输出应用 routed_scaling_factor
    **(
        {
            "scoring_func": self.score_func,
            "apply_routed_scaling_factor_on_output": True,
        }
        if _is_npu
        else {}
    ),
)
```

评论区精华

Reviewer Hexq0210 提出质疑：当前修改是否恢复了 correction_bias = None？作者 McZyWu 回应称不能简单移除该逻辑，因为 NPU 和 GPU 的 MoE 计算路径不同——NPU 使用分组 matmul，GPU 使用非分组实现，非分组路径的 bias 参数设计上始终为空。讨论确认了保留 correction_bias = None if not _is_npu else self.expert_bias 是正确的。

- correction_bias 与 NPU/GPU 路径差异 (design): 保留原有 correction_bias 逻辑，NPU 使用 expert_bias，GPU 使用 None。

风险与影响

- 风险：
  - 回归风险低：仅修改 NPU 分支的特判逻辑，GPU 路径完全不变。但需验证 NPU 下的其他 MoE 模型（如 DeepSeek）是否因新增 apply_routed_scaling_factor_on_output=Tr

ue 而行为改变，理论上该参数仅影响 Trinity 模型。

  - 测试覆盖不足：没有新增单元测试来验证该修复，依赖 NPU CI 回归测试。未来应考虑补充 test_afmoe.py 测试，覆盖 NPU 上的 topk 缩放因子。
  - 兼容性：与 PR #29909 的兼容性良好，未引入新的 data-contract 变更。
- 影响：
  - 影响范围：仅影响 NPU 上使用 AfmoE 架构的模型（如 Trinity），恢复其精度至重构前水平。
  - 性能影响：无，仅调整初始化参数传递。
  - 团队影响：NPU 团队可确认修复后 CI 通过，并于合并后交付正确精度的 NPU 构建。
  - 风险标记：缺少测试覆盖

关联脉络

- PR #29909 [refactor] TopK refactor: 本 PR 修复了 #29909 引入的 NPU 精度回退问题，是该 topk 重构的跟随修复。