

PR #28469 完整报告

sgl-project/sglang

fix(moe): MoRI EP init_mori_op missing BF16 dispatch branch

合并时间: 2026-06-17 12:02

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/28469>

执行摘要

- 一句话: 修复 MoRI EP 缺少 BF16 分发分支导致模型精度错误
- 推荐动作: 值得合并, 修复了一个关键但隐蔽的精度 bug。建议后续补充针对 BF16 分支的单元测试, 确保回归覆盖。

功能与动机

BF16 是默认的分发 dtype, 但 `init_mori_op` 缺少对应分支, 导致 BF16 分发 fallthrough 到 FP8 默认配置, 算子使用 FP8+scale 语义处理 BF16 数据, 结果被静默损坏。该问题由未量化的 BF16 专家层模型暴露。

实现拆解

1. 在 `python/sglang/srt/layers/moe/token_dispatcher/moriep.py` 的 `init_mori_op` 函数中, 于 FP8 分支之前新增 `DispatchDtype.bf16` 分支。
2. 在该分支中, 将 `data_type` 设置为 `params_dtype` (即实际的参数类型, 如 `torch.bfloat16`), 并将 `scale_dim` 设为 0 (禁用 `scale`)。
3. 后续的 FP8 和 FP4 分支保持不变, 形成完整的 if-elif 逻辑结构。

关键文件:

- `python/sglang/srt/layers/moe/token_dispatcher/moriep.py` (模块 MoE; 类别 source; 类型 core-logic): 唯一修改的文件, 在 `init_mori_op` 函数中新增 BF16 分发分支, 修复静默精度错误。

关键符号: `init_mori_op`

关键源码片段

`python/sglang/srt/layers/moe/token_dispatcher/moriep.py`

唯一修改的文件, 在 `init_mori_op` 函数中新增 BF16 分发分支, 修复静默精度错误。

```
# 位于 init_mori_op 函数内, 在 FP8 分支前插入 BF16 处理
if dispatch_dtype == DispatchDtype.bf16:
    data_type = params_dtype # 使用实际的参数类型 (如 bfloat16), 而非 FP8
    scale_dim = 0 # 禁用 scale, BF16 不需要量化 scale
elif dispatch_dtype == DispatchDtype.fp8:
```

```
scale_dim = hidden_size // FP8_BLOCK_SIZE
elif dispatch_dtype == DispatchDtype.fp4:
    # FP4 kernel 仍使用原始 hidden_size 进行内部量化
    hidden_dim = hidden_size
    scale_dim = hidden_size // MXFP4_BLOCK_SIZE
    data_type = torch.float4_e2m1fn_x2
    scale_type_size = torch.float8_e8m0fnu.itemsize
    # ... 模式相关的配置
```

评论区精华

PR 讨论较少，主要来自 author 在 commit message 和 body 中的问题描述。Reviewer HaiShaw 直接批准，未提出额外意见。

- 暂无高价值评论线程

风险与影响

- 风险：该修复改动极小（4 行新增），且显式处理了 BF16 这一关键默认路径，回归风险极低。但未附带单元测试验证 BF16 分支下的逻辑正确性，存在未来重构时被意外删除的长期风险。
- 影响：直接影响使用 MoRI EP 且 expert 层为 BF16 量化的模型，修复了静默精度错误。对其他 dtype（FP8、FP4）无影响。由于 BF16 是默认 dtype，受影响的用户范围较广。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR