

PR #34006 完整报告

sgl-project/sglang

Fix Qwen3-MoE producing garbage with the mori a2a backend

合并时间: 2026-08-09 06:44

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34006>

执行摘要

- 一句话: 一行修复 mori 后端下 Qwen3-MoE 双重归约导致的垃圾输出
- 推荐动作: 值得精读的关键点不在于这一行, 而在于后端门控模式的演进: 新增后端时需要所有 MoE 模型的 forward 分支中同步补全枚举判断。建议后续补充一个针对 mori 后端的回归测试, 或将后端能力判断从各模型手写条件收敛为统一封装, 避免再次遗漏。

功能与动机

PR body 明确指出: `MoeA2ABackend.is_deepest()` 是精确枚举匹配, mori 运行会落入 `forward_normal`, 而 `forward_normal` 会在 mori 已经合并的专家输出之上再做一次 expert-parallel all-reduce。 `deepseek_v2`、`deepseek_v4`、`glm4_moe`、`glm4_moe_lite` 都在 `is_deepest()` 旁同时判断 `is_mori()`, 唯独 `qwen3_moe` 被遗漏, 导致生成退化为重复标点。作者还排除了量化因素: bf16 与 FP8 失败表现一致。

实现拆解

1. 变更入口: `python/sglang/srt/models/qwen3_moe.py` 中 `Qwen3MoeSparseMoeBlock.forward` 的路径选择条件。
2. 核心逻辑: 原条件 `not is_deepest() and not is_ascend_fuseep()` 只覆盖了 `deepest` 与 `ascend_fuseep` 两个后端; mori 作为精确枚举值无法被 `is_deepest()` 命中, 因此落入 `forward_normal`。修复后条件变为 `not is_deepest() and not is_mori() and not is_ascend_fuseep()`, 使 mori 与其余两个专家并行后端一样进入 `forward_deepest`, 避免对 mori 已合并的专家输出再做一次全归约。
3. 配套说明: 本 PR 未新增测试或配置改动, 属于纯源码单行修复; 作者在 8x MI355X (gfx950) 上完成了 Qwen3-30B-A3B 的 RL 验证, Megatron 重算与 SGLang rollout log prob 差异约 0.005。

关键文件:

- `python/sglang/srt/models/qwen3_moe.py` (模块 模型层; 类别 source; 类型 core-logic ; 符号 `Qwen3MoeSparseMoeBlock.forward`, `forward_normal`, `forward_deepest`) : 唯一变更文件, 在 `Qwen3MoeSparseMoeBlock.forward` 的路径选择条件中加入 `is_mori()` 判断, 使 mori 后端走 `forward_deepest`, 避免 `forward_normal` 中的专家并行 all-reduce 双重归约。

关键符号: `Qwen3MoeSparseMoeBlock.forward`, `forward_normal`, `forward_deepest`

关键源码片段

python/sglang/srt/models/qwen3_moe.py

唯一变更文件，在 `Qwen3MoeSparseMoeBlock.forward` 的路径选择条件中加入 `is_mori()` 判断，使 `mori` 后端走 `forward_deepest`，避免 `forward_normal` 中的专家并行 all-reduce 双重归约。

```
def forward(
    self,
    hidden_states: torch.Tensor,
    forward_batch: Optional[ForwardBatch] = None,
) -> torch.Tensor:

    # 关键路径门控: is_deepest() 是精确枚举匹配, mori 不会被命中;
    # 因此必须显式补充 is_mori(), 否则会落入 forward_normal,
    # 对 mori 已做专家并行合并的输出再次 all-reduce, 造成双重归约。
    if (
        not get_moe_a2a_backend().is_deepest()
        and not get_moe_a2a_backend().is_mori()
        and not get_moe_a2a_backend().is_ascend_fuseep()
    ):
        return self.forward_normal(hidden_states)
    else:
        return self.forward_deepest(hidden_states, forward_batch)
```

评论区精华

评审过程非常简洁，仅有合并者 HaiShaw 在 Issue 评论中确认 `gate applied to is_mori`，表明门控已正确落到 `is_mori()`。该评论与 PR body 的自述互相印证，是一次明确的遗漏修复，不涉及接口或语义争议，无未解决疑虑。

- `mori` 门控补充确认 (design): 修复方式被接受：在 `qwen3_moe` 门控中与 `deepest`、`ascend_fuseep` 并列加入 `is_mori()`。

风险与影响

- 风险：主要风险是 `qwen3_moe` 在 `mori` 后端从 `forward_normal` 切换到 `forward_deepest` 后，需要确保该路径对 Qwen3-MoE 的 `ep` 冗余专家、`top-k` 等配置完整可用；不过 `deepseek_v2` 等模型已采用同一门控模式，风险较低。改动仅影响 `mori` 后端下的路径选择，默认 `deepest`、`ascend_fuseep` 及其余后端行为不变。工程风险是缺少针对 `mori` 后端的单元测试，后端枚举继续扩展时可能再次出现同类遗漏。
- 影响：影响范围为使用 `mori a2a` 后端运行 Qwen3-30B-A3B 等 Qwen3-MoE 模型的用户，修复前输出为重复标点垃圾，修复后与无 `a2a` 基线接近。对默认 `deepest` 或其他后端无行为变化，影响面小但正确性收益显著。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- 暂无明显关联 PR