PR #22948 完整报告

sgl-project/sglang

[AMD] Qwen3.5 MXFP4 breaks after shared expert fusion is enabled

合并时间：2026-04-17 06:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22948

执行摘要

- 一句话：修复 Qwen3.5 MXFP4 模型在启用共享专家融合后的崩溃问题。
- 推荐动作：该 PR 值得精读，因为它揭示了量化模型在专家融合时的数据类型不匹配问题，并展示了通过配置检查来优雅降级的设计决策。关注 can_fuse_shared_expert 函数中新增的排除层逻辑，以及未来可能的重构方向（如将逻辑移至 QuantConfig）。

功能与动机

根据 PR body，在 #20736 为 Qwen3.5 模型启用共享专家融合后，MXFP4 模型遇到问题：检查点中的共享专家基于BF16，但当前权重加载只能将其视为MXFP4（路由专家的数据类型）。在在线量化就绪或共享专家预量化为 MXFP4 之前，必须为 MXFP4 模型跳过共享专家融合功能。

实现拆解

1. 修改 can_fuse_shared_expert 函数签名和逻辑：
   - 文件：python/sglang/srt/models/qwen2_moe.py
   - 关键符号：can_fuse_shared_expert
   - 变更：在函数参数中添加 quant_config: Optional[QuantizationConfig]，并新增逻辑检查 quant_config.exclude_layers 是否包含共享专家层（排除 shared_expert_gate 和以 mtp. 开头的层）。如果包含，则返回 False 禁用融合。
   - 原因：防止将 FP32/BF16 共享专家错误地融合到量化 MoE 权重张量中，这需要当前不支持的在线量化。
   - 影响：影响所有使用 can_fuse_shared_expert 的调用点，特别是 Qwen2MoeMLP.__init__ 中的融合决策。
2. 更新 Qwen2MoeMLP.__init__ 中的调用：
   - 文件：python/sglang/srt/models/qwen2_moe.py
   - 关键符号：Qwen2MoeMLP.__init__
   - 变更：将 can_fuse_shared_expert(config) 调用改为 can_fuse_shared_expert(config, quant_config)，传递量化配置以支持新逻辑。
   - 原因：确保融合决策考虑量化配置中的排除层信息。
   - 影响：直接影响 Qwen3.5 MoE 模型的初始化行为，特别是 MXFP4 量化模型。

3. 测试与配置配套：
   - 本次改动未包含直接测试文件变更，但 Issue 评论中讨论了添加 4-GPU e2e 测试到 AMD Nightly Test 的计划。

关键文件：

- python/sglang/srt/models/qwen2_moe.py（模块 模型层；类别 source；类型 core-logic；符号 can_fuse_shared_expert, Qwen2MoeMLP.init）：核心变更文件，修改了共享专家融合决策逻辑，直接影响 Qwen3.5 MoE 模型的初始化和量化行为。

关键符号：can_fuse_shared_expert, Qwen2MoeMLP.init

关键源码片段

python/sglang/srt/models/qwen2_moe.py

核心变更文件，修改了共享专家融合决策逻辑，直接影响 Qwen3.5 MoE 模型的初始化和量化行为。

```
def can_fuse_shared_expert(
    config: PretrainedConfig,
    quant_config: Optional[QuantizationConfig], # 新增参数，用于检查量化配置
) -> bool:
    """Whether the shared expert may be fused as an extra MoE expert (Qwen3.5 + Aiter).

    Caller must still gate on ``support_shared_expert_fusion`` and ``_use_aiter``.
    """
    if (
        get_global_server_args().disable_shared_experts_fusion is True
        or getattr(config, "shared_expert_intermediate_size", 0) <= 0
        or config.shared_expert_intermediate_size != config.moe_intermediate_size
        or get_moe_a2a_backend().is_deepep()
    ):
        return False

    # 如果共享专家被排除在量化之外（在检查点中存储为 FP32），
    # 将其融合到量化 MoE 权重张量需要在线量化，当前不支持。在此情况下禁用融合。
    if quant_config is not None:
        exclude_layers = getattr(quant_config, "exclude_layers", [])
        if any(
            "shared_expert" in layer
            and "shared_expert_gate" not in layer # 排除 gate 层，仅检查专家权重
            and not layer.startswith("mtp.") # 排除 MTP 相关层
            for layer in exclude_layers
        ):
            return False # 检测到共享专家被排除，禁用融合

    return True
```

评论区精华

- 类型提示修正：gemini-code-assist[bot] 指出 quant_config 的类型提示应为 Optional[QuantizationConfig] 而非 None，以保持代码一致性。此建议被采纳（从 diff 可见修正）。
- 设计重构建议：BowenBao 建议将 can_fuse_shared_expert 逻辑移至 QuantConfig 类中，以保持量化相关逻辑内聚，并允许更精确地检查共享专家是否与 MoE 层共享相同的量化规格。此建议未被立即采纳，但 HaiShaw 在批准时提到“look for refactor is feasible”，表明未来可能考虑重构。
  - quant_config 类型提示修正 (correctness): 建议被采纳，类型提示已修正。
  - 将融合逻辑重构到 QuantConfig 中 (design): 未被立即采纳，但 HaiShaw 在批准时提到未来可能考虑重构。

风险与影响

- 风险：- 回归风险：低。变更仅影响 MXFP4 量化模型的共享专家融合行为，通过条件检查避免破坏现有功能。但需确保 exclude_layers 属性在所有量化配置中正确设置。
- 性能影响：无显著性能风险。禁用融合可能略微增加计算开销，但避免了崩溃，是必要的权衡。
- 兼容性：与现有量化配置兼容，只要 quant_config 为 None 或包含 exclude_layers 属性即可。
- 安全风险：无。
- 影响：- 用户影响：MXFP4 量化模型的 Qwen3.5 用户将不再遇到共享专家融合导致的崩溃，模型可正常加载和运行。
- 系统影响：仅影响使用 qwen2_moe.py 中共享专家融合逻辑的 MoE 模型初始化路径，特别是 AMD 平台上的 MXFP4 量化场景。
- 团队影响：修复了已知问题，减少了支持负担；但未添加单元测试，依赖现有 CI 和计划中的 e2e 测试来验证。
- 风险标记：配置依赖风险，缺少单元测试

关联脉络

- PR #20736 （未提供，但从 PR body 引用）：PR body 提到该 PR 为 Qwen3.5 模型启用了共享专家融合功能，导致当前问题，是本 PR 的直接前置变更。