# PR #34447 完整报告

sgl-project/sglang

[Fix][Qwen]: fused shared-expert detection PP-safe protection

合并时间：2026-08-12 09:50

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34447

## 执行摘要

- 一句话：修复 Qwen PP 下融合共享专家检测崩溃
- 推荐动作：值得精读。这是一份小而完整的 PP 兼容修复样例，展示了 PPMissingLayer 占位符在模型初始化阶段的典型陷阱，以及如何用“本地区间扫描 + getattr 防御”规避。关注 _get_num_fused_shared_experts 的防御风格与测试桩设计即可，无需深入其它模块。

## 功能与动机

Qwen MoE 的 fused shared-expert 检测总是读取 self.model.layers[0].mlp，但 make_layers() 会将其它 pipeline stage 拥有的 decoder 层保留为 PPMissingLayer 占位符，导致非首个 PP stage 在模型初始化时崩溃：AttributeError: 'PPMissingLayer' object has no attribute 'mlp'。该崩溃发生在服务启动阶段，任何 PP 多卡部署都无法启动模型。

## 实现拆解

1. 定位根因：Qwen3_5MoeForConditionalGeneration.__init__ 在 _use_aiter 且未禁用共享专家融合时调用 _get_num_fused_shared_experts；旧实现直接检查 self.model.layers[0].mlp.num_fused_shared_experts，但 PP 非首 stage 的 decoder 层经 make_layers() 替换为 PPMissingLayer 占位符，该占位符没有 mlp 属性，因此在初始化阶段抛出 AttributeError。
2. 修复 _get_num_fused_shared_experts：先判断 self.model 是否有 layers 属性；然后仅遍历 [self.model.start_layer, self.model.end_layer) 区间内的本地层，用 getattr 安全获取 mlp，返回第一个暴露 num_fused_shared_experts 的本地 MLP 的值；若本地层都未启用融合则返回 0。该函数只在模型初始化时执行一次，不涉及请求或 token 前向路径。
3. 测试配套：新增 test/registered/unit/models/test_qwen3_5_pipeline_parallel.py，注册为 CPU CI 用例（suite=base-a-test-cpu）；三个测试分别覆盖无 decoder 层、本地 PP 区间非零融合数、本地无融合三种场景，其中前两个用例在修复前会复现 AttributeError。
4. 验证：PR 作者在 2 节点 16x MI300X（TP8/PP2，AITER + FP8 E4M3 KV cache）环境验证两个 PP stage 均能就绪，GSM8K strict-match 0.958、flexible-extract 0.958，无 4xx/5xx 响应；标准 CI（含 extra run）通过。

关键文件：

- python/sglang/srt/models/qwen3_5.py（模块 模型实现；类别 source；类型 core-logic；符号 _get_num_fused_shared_experts）：修复入口：将 fused shared-expert 检测从固定

读 layers[0] 改为扫描本地 PP 区间，避免非首 stage 的 PPMissingLayer 占位符导致初始化崩溃。

- test/registered/unit/models/test_qwen3_5_pipeline_parallel.py（模块 测试套件；类别 test；类型 test-coverage；符号 TestQwen3_5PipelineParallel, test_get_num_fused_shared_experts_returns_zero_without_layers, test_get_num_fused_shared_experts_uses_local_pp_layers, test_get_num_fused_shared_experts_returns_zero_without_local_fusion）：新增 CPU 回归测试，覆盖无 decoder 层、本地 PP 区间、本地无融合三种场景，直接验证修复后的行为。

关键符号：_get_num_fused_shared_experts

## 关键源码片段

### python/sglang/srt/models/qwen3_5.py

修复入口：将 fused shared-expert 检测从固定读 layers[0] 改为扫描本地 PP 区间，避免非首 stage 的 PPMissingLayer 占位符导致初始化崩溃。

```
def _get_num_fused_shared_experts(self):
    # PP 非首 stage 的 decoder 层会被替换成 PPMissingLayer 占位符，
    # 旧实现直接读 layers[0].mlp 会在模型初始化时抛
    # AttributeError: 'PPMissingLayer' object has no attribute 'mlp'。
    # 因此这里先确认 model 是否真正持有 layers 容器。
    if not hasattr(self.model, "layers"):
        return 0
    # 只在当前 PP rank 负责的本地区间 [start_layer, end_layer) 内扫描，
    # 跳过其它 stage 的占位符层；用 getattr 兜底避免层实现差异。
    for layer_id in range(self.model.start_layer, self.model.end_layer):
        mlp = getattr(self.model.layers[layer_id], "mlp", None)
        # 只要某个本地 MLP 暴露 num_fused_shared_experts，
        # 即认为该配置启用了共享专家融合，并采用它的数值。
        if hasattr(mlp, "num_fused_shared_experts"):
            return mlp.num_fused_shared_experts
    # 本地层均未启用融合（或无 decoder 层）时返回 0。
    return 0
```

### test/registered/unit/models/test_qwen3_5_pipeline_parallel.py

新增 CPU 回归测试，覆盖无 decoder 层、本地 PP 区间、本地无融合三种场景，直接验证修复后的行为。

```
class TestQwen3_5PipelineParallel(CustomTestCase):
    # 用 SimpleNamespace 构造轻量模型桩，
    # 避免在 CPU 单测中拉起真实模型权重。
    @staticmethod
    def _get_num_fused_shared_experts(layers, start_layer, end_layer):
        model = SimpleNamespace(
            model=SimpleNamespace(
                layers=layers,
```

```
                start_layer=start_layer,
                end_layer=end_layer,
            )
        )
        return Qwen3_5MoeForConditionalGeneration._get_num_fused_shared_experts(model)

    def test_get_num_fused_shared_experts_uses_local_pp_layers(self):
        # 模拟 PP = 2 的第二个 stage：前两层是 PPMissingLayer 占位符，
        # 本地区间 [2, 4) 内的两个 MLP 都带有 num_fused_shared_experts = 1。
        layers = [
            PPMissingLayer(),
            PPMissingLayer(),
            SimpleNamespace(mlp=SimpleNamespace(num_fused_shared_experts=1)),
            SimpleNamespace(mlp=SimpleNamespace(num_fused_shared_experts=1)),
        ]
        self.assertEqual(
            self._get_num_fused_shared_experts(layers, start_layer=2, end_layer=4),
            1,
        )
```

## 评论区精华

Review 过程没有展开的设计争论，三位 reviewer 均直接 approve：

- yichiche：确认该防护可修复 AttributeError: 'PPMissingLayer' object has no attribute 'mlp'。
- HaiShaw：改动既覆盖了非 PP 场景的前置条件，也扩展覆盖 PP 下模型分片的情况。
- 1am9trash：空评论通过。 整体属于明确的小范围 bugfix，无未解决疑虑。
- PP 下 fused shared-expert 检测保护 (correctness): 三位 reviewer 均 approve，无未解决疑虑。

## 风险与影响

- 风险：
  - 依赖 start_layer / end_layer 属性：修复后扫描依赖这两个属性存在；若某配置下 layers 存在但未设置区间属性，会抛 AttributeError。当前 make_layers() 总会设置，风险较低，但测试未覆盖该组合。
  - 语义假设：num_fused_shared_experts 在层间应一致，实现取第一个有该属性的本地 MLP 值；若未来支持层间差异化融合，需重新审视该逻辑。
  - 行为兼容：对非 PP 单 stage，start_layer 通常为 0、end_layer 为总层数，行为与原来一致；无 decoder 层时返回 0，也保持旧逻辑的容错。
  - 回归面：改动局限于模型初始化路径，不进入前向、调度或 kernel 执行路径，且新增 CPU 回归测试，回归风险低。
- 影响：

- 用户：修复 Qwen3.5 MoE 在 PP 多 stage 下无法启动的问题，使 AITER 共享专家融合与 PP 可叠加使用；非 PP 用户无感知。
- 系统：仅初始化期多一次 O( 本地层数 ) 的扫描，无运行时开销，不影响吞吐和延迟。
- 团队：新增 CPU 回归用例可防止后续对 PP 占位符或融合检测的改动重新引入该崩溃，降低 PP 相关维护成本。
- 风险标记：PP 非首 stage 模型初始化路径，依赖 start_layer/end_layer 属性存在，测试基于桩对象而非真实 PP 环境

## 关联脉络

- PR #34357 Update Qwen3.5 B200 NVFP4 MTP config: 同属 Qwen3.5 模型家族，涉及 PP 与部署配置；本 PR 与其无直接代码依赖，但同功能线可串联阅读。