PR #35719 完整报告

sgl-project/sglang

[AMD] Fix Qwen3.5 MTP dropping fused shared-expert weights

合并时间：2026-08-25 04:18

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35719

执行摘要

- 一句话：修复 Qwen3.5 MTP 丢弃融合共享专家权重的问题
- 推荐动作：值得精读，该 PR 展示了如何处理共享专家融合带来的权重布局变化，以及如何保持与主模型逻辑的一致性。关注其对 qwen3_5.py 的镜像逻辑和未来可能需要的扩展。

功能与动机

自 #33889 起，Qwen3.5 MXFP4 模型在 MI355X 上使用 MTP 投机解码时吞吐量下降约 12%。原因是 MTP 草稿模型在加载时未包含共享专家权重。由于 #33889 将共享专家融合的决策从全局覆盖改为按 runner 独立决策，草稿模型开始首次触发融合，但 MTP 加载器未适配，导致共享专家权重被静默跳过。本 PR 旨在修复此问题，使草稿模型能够正确加载融合后的共享专家权重。

实现拆解

1. 引入平台与开关变量：在 qwen3_5_mtp.py 顶部新增 _is_hip = is_hip() 和 _use_aiter = get_bool_env_var("SGLANG_USE_AITER") and _is_hip，用于确定是否启用共享专家融合逻辑。
2. 计算融合专家数量：在 load_weights 方法中，遍历模型的模块，查找 num_fused_shared_experts 属性，若存在则将其赋值给局部变量 num_fused_shared_experts，该数值表示融合的共享专家数量，这些专家被放置在路由专家槽位 num_experts 之后。
3. 调整专家映射大小：将 expert_params_mapping 的 num_experts 参数改为 num_experts + num_fused_shared_experts，使得映射能够包含融合专家槽位。
4. 重写权重名称：在 load_fused_expert_weights 中，当 _use_aiter 且 num_fused_shared_experts > 0 时，将 mlp.shared_expert. 前缀的重命名为 mlp.experts.{num_experts}.，以便权重加载到正确的槽位。该逻辑仅在启用融合时生效，非融合配置下保持原行为。

关键文件：

- python/sglang/srt/models/qwen3_5_mtp.py（模块 模型加载；类别 source；类型 data-contract）：核心修复文件，修改了 MTP 加载器以识别融合共享专家并正确加载权重。

关键符号：load_weights, load_fused_expert_weights

关键源码片段

python/sglang/srt/models/qwen3_5_mtp.py

核心修复文件，修改了 MTP 加载器以识别融合共享专家并正确加载权重。

```
# python/sglang/srt/models/qwen3_5_mtp.py
# 判断是否使用 aiter，仅 AMD 平台启用共享专家融合逻辑
_is_hip = is_hip()
_use_aiter = get_bool_env_var("SGLANG_USE_AITER") and _is_hip

def load_weights(self, weights, is_mtp=False):
    # ...
    num_experts = getattr(self.config, "num_experts", None)
    # 融合的共享专家会被放入额外的路由槽位 num_experts
    num_fused_shared_experts = 0
    if _use_aiter:
        for module in self.modules():
            fused = getattr(module, "num_fused_shared_experts", 0)
            if fused:
                num_fused_shared_experts = fused
                break
    if num_experts is not None:
        # 将映射大小扩展到包含融合专家，否则共享专家权重会找不到匹配
        expert_params_mapping = FusedMoE.make_expert_params_mapping(
            ckpt_gate_proj_name="gate_proj",
            ckpt_down_proj_name="down_proj",
            ckpt_up_proj_name="up_proj",
            num_experts=num_experts + num_fused_shared_experts,
        )
    # ...

def load_fused_expert_weights(self, name, loaded_weight):
    # ...
    if _use_aiter and num_fused_shared_experts > 0 and "mlp.shared_expert." in name:
        # 将共享专家权重重命名到对应的融合槽位，否则会被跳过
        name = name.replace("mlp.shared_expert.", f"mlp.experts.{num_experts}.")
    # ...
```

评论区精华

- 触发条件对齐：审查者 1am9trash 指出 qwen3_5.py 中共享专家融合的判断基于 _use_aiter，而本 PR 最初基于 _is_hip，建议对齐。作者随后修改为使用 _use_aiter，与 qwen3_5.py 保持一致。
- 触发条件对齐 (design): 作者已修改为使用 _use_aiter。

风险与影响

- 风险：

    - 影响范围受限：改动仅在 _use_aiter 为真时生效，即 AMD 平台且设置 SGLANG_USE_AITER，其他平台无影响。
    - 潜在的未覆盖场景：PR 明确说明不支持融合检查点变体（单个 experts.gate_up_proj 张量），若未来出现此类检查点，可能导致加载失败。
    - 测试覆盖缺失：本 PR 未添加自动化测试，且 CI 中可能没有覆盖相关场景，存在回归风险。
- 影响：
    - 用户影响：修复了 AMD 平台上 Qwen3.5 MTP 推理吞吐量下降的问题，恢复至预期性能。
    - 系统影响：仅影响 MTP 草稿模型的加载逻辑，不影响正常模型推理。
    - 团队影响：改动较小，降低了维护成本，但需注意未来对融合检查点的支持。
    - 风险标记：缺少测试覆盖，特定平台相关

关联脉络

- PR #33889 Make shared-experts-fusion per-runner: 该 PR 引入了按 runner 的共享专家融合开关，导致 MTP 草稿模型开始融合共享专家，是本 PR 需要修复的根因。