PR #42716 完整报告

vllm-project/vllm

Fix Weight loading for Qwen3.5-MTP and Qwen3-VL using runai_streamer

合并时间：2026-05-17 08:54

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42716

执行摘要

- 一句话：修复 Qwen3.5-MTP 与 Qwen3-VL MoE 权重加载中参数丢失
- 推荐动作：这是一个明确且低风险的 bugfix，值得合并。代码虽少，但体现了对数据契约一致性的重视。建议在类似模型（如其他 MoE 模型）中检查是否有相同的调用模式，统一修复以避免遗留。

功能与动机

PR body 指出，本修复参考了 PR #42521（未提供具体链接），但认为同样适用于 Qwen3.5-MTP 和 Qwen3-VL 模型。根本动机是：在使用 runai_streamer（TPU 环境）时，weight_loader 期望的 expert_id 和 shard_id 必须以关键字参数传递，否则会被忽略，导致 MoE 层权重加载异常。

实现拆解

1. 定位问题：在 qwen3_5_mtp.py 和 qwen3_vl_moe.py 的 load_fused_expert_weights 方法中，调用 weight_loader() 时，shard_id 和 expert_id 以位置参数传递，而不是关键字参数。
2. 修复方案：将这两个参数改为关键字参数传递：shard_id=shard_id, expert_id=expert_id，确保即使 weight_loader 的参数顺序有变化，也能正确绑定。
3. 测试配套：本次 PR 未附带测试变更。依赖已有测试覆盖。

关键文件：

- vllm/model_executor/models/qwen3_5_mtp.py（模块 模型加载；类别 source；类型 data-contract；符号 load_fused_expert_weights）：修复 MoE 权重加载中 shard_id 和 expert_id 参数传递方式，从位置参数改为关键字参数。
- vllm/model_executor/models/qwen3_vl_moe.py（模块 模型加载；类别 source；类型 data-contract；符号 load_fused_expert_weights）：与 qwen3_5_mtp.py 相同的修复，确保 MoE 权重加载参数正确传递。

关键符号：load_fused_expert_weights

关键源码片段

vllm/model_executor/models/qwen3_5_mtp.py

修复 MoE 权重加载中 shard_id 和 expert_id 参数传递方式，从位置参数改为关键字参数。

```
def load_fused_expert_weights(
    self,
    name: str,
    params_dict: dict,
    loaded_weight: torch.Tensor,
    shard_id: str,
    num_experts: int,
) -> bool:
    param = params_dict[name]
    weight_loader = typing.cast(Callable[..., bool], param.weight_loader)
    loaded_local_expert = False
    for expert_id in range(num_experts):
        curr_expert_weight = loaded_weight[expert_id]
        # 修复：使用关键字参数传递 shard_id 和 expert_id，
        # 避免位置参数顺序不匹配导致参数静默丢失（尤其在 TPU 环境下）
        success = weight_loader(
            param,
            curr_expert_weight,
            name,
            shard_id=shard_id,
            expert_id=expert_id,
            return_success=True,
        )
        if success:
            loaded_local_expert = True
    return loaded_local_expert
```

vllm/model_executor/models/qwen3_vl_moe.py

与 qwen3_5_mtp.py 相同的修复，确保 MoE 权重加载参数正确传递。

```
def load_fused_expert_weights(
    self,
    name: str,
    params_dict: dict,
    loaded_weight: torch.Tensor,
    shard_id: str,
    num_experts: int,
) -> bool:
    param = params_dict[name]
    weight_loader = typing.cast(Callable[..., bool], param.weight_loader)
    loaded_local_expert = False
    for expert_id in range(num_experts):
        curr_expert_weight = loaded_weight[expert_id]
        # 修复：使用关键字参数传递 shard_id 和 expert_id，
        # 避免位置参数顺序不匹配导致参数静默丢失（尤其在 TPU 环境下）
        success = weight_loader(
            param,
            curr_expert_weight,
```

```
                name,
                shard_id=shard_id,
                expert_id=expert_id,
                return_success=True,
            )
            if success:
                loaded_local_expert = True
    return loaded_local_expert
```

评论区精华

讨论较少。PR 作者请求两位开发者审查（ZJY0516、hks-9697-v2），仅收到 LGTM。自动化机器人（claude、gemini-code-assist）给出一般性评论，最终由核心维护者 mgoin 和 Isotr0py 批准。无实质性争议。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。仅修改了两个函数调用中的参数传递方式（从位置参数改为关键字参数），功能逻辑不变。若 weight_loader 内部依赖位置参数顺序（虽不符合常见设计），可能引入兼容性问题，但根据代码上下文，weight_loader 是 vLLM 内部的统一回调，始终支持关键字参数。未发现回归风险。
- 影响：影响范围局限于 Qwen3.5-MTP 和 Qwen3-VL 两个模型在特定环境（TPU runai_streamer）下的 MoE 权重加载。修复后，这些模型在 TPU 上应能正确加载 MoE 专家权重。对其他模型和环境无影响。
- 风险标记：未附带测试，依赖外部接口契约

关联脉络

- PR #42521 Fix MoE weight loading for runai_streamer: PR 作者指出此 PR 参考了 #42521，但将其应用于 Qwen3.5-MTP 和 Qwen3-VL 模型。