# PR #46441 完整报告

vllm-project/vllm

fix gpt_oss pp>1 with ep

合并时间：2026-06-23 16:59

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/46441

## 执行摘要

- 一句话：修复 GPT-OSS 在 PP > 1 时 EP 权重加载失败
- 推荐动作：该 PR 是一行修复，逻辑清晰，风险低，建议合入。值得关注的是评审者的发现：GPT-OSS 是唯一错误使用 get_ep_group().rank 的模型，其他模型实现均正确。建议后续在代码审查中统一推广 rank_in_group 的使用，或在框架层面封装正确的方法以避免类似 bug。

## 功能与动机

当同时使用 expert parallelism（EP）和 pipeline parallelism（PP > 1）时，GptOssModel.load_weights() 中 ep_rank = get_ep_group().rank 返回的是所有 worker 中的全局 rank，而非 EP 组内的局部 rank。对于两阶段流水线，PP stage 1 的 worker 全局 rank 偏移了 pp_rank * tp_size，导致 ep_rank_start 越界，expert 权重切片为空，所有 expert 权重加载被静默跳过。PR body 详细描述了该 bug 的触发条件和根因。

## 实现拆解

1. 定位问题：在 vllm/model_executor/models/gpt_oss.py 的 load_weights 方法中，第 1081 行使用 get_ep_group().rank 获取 EP 组 rank，该值在 PP 场景下是全局 rank（横跨多个 pipeline stage），导致后续 ep_rank_start = ep_rank * experts_per_rank 计算错误。
2. 修复措施：将 get_ep_group().rank 替换为 get_ep_group().rank_in_group，该方法返回当前 worker 在 EP 组内的 0-based rank，不受 pipeline stage 影响。
3. 影响范围：仅修改一行代码，直接修复了 GPT-OSS 模型在 EP + PP 组合配下的权重加载逻辑。其他模型若使用类似模式未受影响，reviewer 确认 GPT-OSS 是唯一使用 get_ep_group().rank 的模型。

关键文件：

- vllm/model_executor/models/gpt_oss.py（模块 模型加载；类别 source；类型 data-contract；符号 load_weights）：包含 load_weights 方法，是修复的唯一文件。将 get_ep_group().rank 改为 get_ep_group().rank_in_group，修复 PP + EP 组合下 expert 权重加载越界问题。

关键符号：load_weights

## 关键源码片段

### vllm/model_executor/models/gpt_oss.py

包含 load_weights 方法，是修复的唯一文件。将 get_ep_group().rank 改为 get_ep_group().rank_in_group，修复 PP + EP 组合下 expert 权重加载越界问题。

```
# vllm/model_executor/models/gpt_oss.py (line 1081)
# 修复前：get_ep_group().rank 返回全局 rank，在 PP>1 时出错
# 修复后：get_ep_group().rank_in_group 返回 EP 组内的局部 rank，正确计算 expert 权重切片

ep_size = get_ep_group().world_size
ep_rank = get_ep_group().rank_in_group # 关键修复：使用 rank_in_group 替代 rank
num_experts = self.config.num_local_experts
experts_per_rank = num_experts // ep_size
ep_rank_start = ep_rank * experts_per_rank
ep_rank_end = (ep_rank + 1) * experts_per_rank
```

## 评论区精华

PR 无 review 评论，但 reviewer jikunshang 在批准时附加了一条评论："LGTM. seems only gpt_oss use get_ep_group().rank instead of get_ep_group().rank_in_group"，确认该问题是 GPT-OSS 模型特有的，其他模型已正确使用 rank_in_group。

- 暂无高价值评论线程

## 风险与影响

- 风险：
  1. 回归风险低：变更仅一行，且方法名语义明确（rank vs rank_in_group），reviewer 确认仅 GPT-OSS 存在此问题。
  2. 缺少测试覆盖：PR 未附带任何测试用例，虽然变更简单明确，但添加针对 PP + EP 场景的权重加载测试有助于防止未来回归。
  3. 兼容性：对于未使用 PP 的场景（PP=1），get_ep_group().rank 与 get_ep_group().rank_in_group 行为一致，无影响。
- 影响：
  1. 用户影响：修复后，使用 GPT-OSS 模型且同时开启 expert parallelism 与 pipeline parallelism（--enable-expert-parallel 和 --pipeline-parallel-size > 1）的用户将不再遇到权重加载静默失败的问题，模型可正确加载并使用 MoE 层。
  2. 系统影响：无性能或资源占用影响。
  3. 团队影响：提醒其他模型作者在实现权重加载时注意 EP 组 rank 的正确获取方式，推广使用 rank_in_group。 - 风险标记：缺少测试覆盖

## 关联脉络

- PR #46945 [Bugfix][Responses] Set completed status for Harmony function calls: 相同标签 gpt-oss，同属 GPT-OSS 生态的 bug 修复