PR #37529 完整报告

vllm-project/vllm

[ROCm] Enable MORI EP for unquantized MoE with AITER backend

合并时间：2026-03-30 15:19

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/37529

执行摘要

该 PR 修复了在 ROCm 平台上，使用未量化 MoE 模型和 AITER 专家后端时，MORI 专家并行（EP）静默失效的 bug，通过调整调度逻辑确保 token 正确分发到远程 GPU，避免约 87.5% 专家贡献丢失，显著提升模型输出质量。

功能与动机

为什么做：在ROCm环境中，当启用MORIEP（--all2all-backendmori）配合未量化（BF16）MoE 模型和 AITER 后端时，原有代码导致 UnquantizedFusedMoEMethod.maybe_make_prepare_finalize() 返回 None，静默跳过 MORI dispatch/combine。这使得每个 GPU 仅运行本地专家，丢失大部分专家贡献，模型输出严重退化（如 gsm8k 基准测试准确性下降）。PR body 明确描述："The model appears to work but produces degraded output."

实现拆解

关键改动文件：

- unquantized_fused_moe_method.py：移除 AITER 后端的特殊返回 None 逻辑，确保 maybe_make_prepare_finalize() 调用父类方法生成 MoriPrepareAndFinalize；在 select_gemm_impl() 中添加 AiterExperts 支持，代码示例：
- all2all_utils.py：区分量化与未量化调度，当未量化时设置 quant_dtype=moe.in_dtype、scale_dim=0 和 scale_type_size=0，避免因 quant_config.quant_dtype 为 None 导致的崩溃。
- layer.py：初始添加验证守卫，但在 review 后移除，因修复核心在其他文件。

评论区精华

主要讨论点：

1. scale_type_size 清晰度优化：gemini-code-assist[bot] 指出："当 scale_dim 为 0 时，设置 scale_type_size 为 torch.float32.itemsize 不一致 ... clearer to explicitly set scale_type_size to 0"。作者基于 MORI 测试更新，提升代码可读性。
2. 守卫设计简化：tjtanaa 提问："Do you think there is a better way rather than keep on listing each and every flag?" 作者回应移除了守卫，强调它"incomplete and unrelated to the actual fix"，展示聚焦核心问题的设计决策。

风险与影响

技术风险：

- 回归风险：调度逻辑变更可能影响其他后端（如 TritonExperts）或量化配置，需充分测试。
- 性能影响：启用 MORI dispatch 增加跨 GPU 通信，但修复了专家贡献丢失，整体准确性提升，基准测试显示吞吐量稳定。
- 兼容性：需与 #37418 协调，确保 FP8 和未量化路径均正常工作。 影响评估：
- 用户：修复输出质量退化，提升模型推理准确性，尤其对于大规模 MoE 模型。
- 系统：ROCm 平台专家并行性恢复正常，增强分布式推理可靠性。
- 团队：揭示 MoE 调度路径的持续优化，需关注跨 PR 协同测试。

关联脉络

与历史 PR 的关系：本 PR 是 #37418 的 companion，后者修复 FP8 dispatch 路径，两者共同完善 MoE 专家并行调度。从近期历史 PR 分析看，vllm-project/vllm 仓库频繁涉及 ROCm 优化（如 PR #38450 修复交叉注意力调度）和 MoE 改进（如 PR #38329 修复 TRT-LLM 内核），显示对异构硬件和专家模型支持的持续投入。本 PR 填补了未量化场景下的关键漏洞，是 ROCm + MoE 功能演进中的重要一环。