

PR #49985 完整报告

vllm-project/vllm

[Bugfix][CPU] Fall back to torch for unaligned swigluoai on NEON/vec MoE

合并时间: 2026-07-28 02:35

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49985>

执行摘要

- 一句话: 修复 CPU NEON/vec MoE 路径未正确回退 swigluoai 激活
- 推荐动作: 值得阅读的修复。展示了在异构平台上维护多后端 MoE 路径时, 补丁容易被遗漏的风险。对于团队而言, 可以学习如何通过早期守卫与常量对齐来隔离这类问题。

功能与动机

PR #49591 为 swigluoai 添加了 grouped-gemm 对齐守卫, 但只作用于 AMX 路径。NEON/vec 分支被遗漏, 检测到 `w13_input_size % 4 == 0` 和 `w2_input_size % 4 == 0` 时返回 `True` 并调用 `forward_grouped_gemm`, 而 swigluoai 的交错布局需要 `w2_input_size` 是 tile 大小 (`_MOE_GROUPED_GEMM_N_TILE`) 的倍数, 否则 C++ 内核会触发 `TORCH_CHECK` 或产生错误结果。这导致 Arm CPU Test 中的 `test_cpu_fused_moe_unaligned_intermediate_size_swigluoai` 确定性失败, 影响了作者的 PR #48391 等其它变更。

实现拆解

1. 定位问题: 在 `vllm/model_executor/layers/fused_moe/cpu_fused_moe.py` 文件的 `check_grouped_gemm` 方法中, NEON 和 vec 路径均未考虑激活函数类型, 允许 swigluoai 在非对齐情况下通过。
2. 添加守卫: 在 NEON/vec 分支之前, 插入一个提前返回条件: 如果 `layer.activation == MoEActivation.SWIGLUOAI` 且 `w2_input_size` 不能被 `_MOE_GROUPED_GEMM_N_TILE` (常量, 与 C++ 内核 tile 大小一致) 整除, 则返回 `(False, "none")`, 强制使用逐专家 torch 循环。
3. 无其它改动: 该 PR 只修改了一个文件, 增加 6 行代码, 无删除。测试方面已有 `test_cpu_fused_moe_unaligned_intermediate_size_swigluoai` 专门验证此行为, 因此无需新增测试。

关键文件:

- `vllm/model_executor/layers/fused_moe/cpu_fused_moe.py` (模块 MoE 层; 类别 source; 类型 data-contract): 核心修复文件, 修改 `check_grouped_gemm` 方法在 NEON/vec 路径前添加 swigluoai 对齐守卫。

关键符号: `vllm/model_executor/layers/fused_moe/cpu_fused_moe.py::CPUFusedMoEMethod::check_grouped_gemm`

关键源码片段

vllm/model_executor/layers/fused_moe/cpu_fused_moe.py

核心修复文件，修改 `check_grouped_gemm` 方法在 NEON/vec 路径前添加 `swigluoai` 对齐守卫。

```
def check_grouped_gemm(self, layer: torch.nn.Module) -> tuple[bool, str]:
    """检查当前层是否可以使用 grouped GEMM。"""
    # ... 前置检查省略
    if not (w13_output_size % 32 == 0 and w2_output_size % 32 == 0):
        return False, "none"

    # SWIGLUOAI 的交错布局要求 w2_input_size 是 N_TILE 的倍数,
    # 否则 C++ grouped-gemm 内核会崩溃或产生错误结果。
    # 该守卫对所有非 AMX 路径（NEON 和 vec）均生效。
    if (
        layer.activation == MoEActivation.SWIGLUOAI
        and w2_input_size % _MOE_GROUPED_GEMM_N_TILE != 0
    ):
        return False, "none"

    supports_neon = current_platform.get_cpu_architecture() == CpuArchEnum.ARM
    if supports_neon:
        if (
            dtype == torch.bfloat16
            and w13_input_size % 4 == 0
            and w2_input_size % 4 == 0
        ):
            return True, "neon"
        return False, "none"

    return True, "vec"
```

评论区精华

作者 `oops-oom` 在合并后提出一个后续问题：同样 C++ 约束 (`output_size_13%(2*N_TILE)`) 适用于所有激活，因此 Arm 上 SILU/GELU 的未对齐 `intermediate` 也会触发 `TORCH_CHECK`，只是由于通用测试在 Arm 上被跳过而未被发现。作者询问是否应该泛化守卫并添加 Arm 测试，还是保持范围。维护者 `bigPYJ1151` 回复这部分即将重构，建议本次只修复 CI 问题。

- 是否应泛化守卫至其它激活 (question): 维护者表示该部分即将重构，建议本次只解决 CI 问题。

风险与影响

- 风险：风险极低：变更仅新增一个早期的回退条件，未修改任何现有逻辑或数据类型；回退路径 (`forward_torch`) 已经经过充分测试；守卫条件与 AMX 路径中已使用的 C++ 内核对齐约束一致。

- 影响：影响范围小：仅影响 ARM NEON 和通用 vec 平台上使用 swigluoai 激活且 w2_input_size 不是 tile 整数倍的 MoE 层。修复了 Arm CPU 测试的确定性失败，同时避免了运行时未定义行为或崩溃。对其它激活（如 SILU、GELU）无影响，因为它们不满足 swigluoai 分支条件。
- 风险标记：暂无

关联脉络

- PR #49591 [Bugfix][CPU] Add swigluoai grouped-gemm alignment check: 本 PR 是对 #49591 的补丁，修复了其遗漏的 NEON/vec 路径。
- PR #48391 [Bugfix][CPU] ...: 作者正是因该 PR 触发了 Arm CPU Test 而发现此预存回归。