

PR #56130 完整报告

vllm-project/vllm

[ROCm][CI] Use a platform-independent GEMM in the merged-column fuser test

合并时间: 2026-09-10 04:08

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56130>

执行摘要

- 一句话: 修复 MergedColumn 测试在 ROCm 平台因调用平台特定 GEMM 导致的失败
- 推荐动作: 该 PR 值得精读, 特别是对关注测试平台独立性和 GEMM 调度机制的工程师。它展示了一个常见的测试问题: 当测试依赖平台特定后端时, 可能在不同硬件上失败。修复方案简洁有效, 通过 monkeypatch 实现了测试的平台无关性。建议关注 UnquantizedLinearMethod 中 GEMM 绑定的机制, 以及 default_unquantized_gemm 如何作为平台无关的回退实现。

功能与动机

该变更修复了 ROCm CI 上的测试失败。根据 PR body 和 Issue 评论, `test_merged_column_fuser_supports_any_number_of_linears` 在 AMD CI 上失败, 报错 `NotImplementedError: Could not run 'vllm::rocm_unquantized_gemm' with arguments from the 'CPU' backend`。这是因为 `UnquantizedLinearMethod` 在构造时绑定了平台特定的 GEMM, 而测试输入是 CPU 张量。维护者 `AndreasKaratzas` 在 Issue 评论中指出该回归源自 PR#55301。

实现拆解

1. 修改导入: 在 `tests/models/transformers/fusers/test_linear.py` 中, 将 `from vllm.model_executor.layers.utils import dispatch_cpu_unquantized_gemm` 替换为 `from vllm.model_executor.layers.utils import default_unquantized_gemm`。这移除了对 CPU 特定 GEMM 调度函数的依赖。
2. 添加 Monkeypatch: 在 `test_merged_column_fuser_supports_any_number_of_linears` 函数中, 新增一行 `monkeypatch.setattr(linear, "dispatch_unquantized_gemm", lambda linear_backend="auto": default_unquantized_gemm)`。这确保在测试执行期间, `UnquantizedLinearMethod` 调用的 GEMM 分发函数始终返回平台无关的默认实现, 避免绑定到 `rocm_unquantized_gemm`。
3. 移除显式 CPU GEMM 调用: 删除了原来在断言前调用 `dispatch_cpu_unquantized_gemm(merged, remove_weight=False)` 的代码行。由于 monkeypatch 已经在上层解决了 GEMM 后端绑定问题, 这一步现在变得多余。

关键文件:

- tests/models/transformers/fusers/test_linear.py (模块 线性层融合; 类别 test; 类型 test-coverage) : 本次变更的唯一文件, 修复了 ROCm 平台上的测试失败

关键符号: test_merged_column_fuser_supports_any_number_of_linears

关键源码片段

tests/models/transformers/fusers/test_linear.py

本次变更的唯一文件, 修复了 ROCm 平台上的测试失败

```
# 变更后的测试函数核心部分 (相对完整实现片段)
def test_merged_column_fuser_supports_any_number_of_linears(
    cls, default_vllm_config, monkeypatch
):
    from vllm.model_executor import parameter
    from vllm.model_executor.layers import linear
    from vllm.model_executor.layers.linear import MergedColumnParallelLinear
    from vllm.model_executor.layers.utils import default_unquantized_gemm
    from vllm.model_executor.models.utils import AutoWeightsLoader, WeightsMapper

    # 设置 TP 环境为单机 (rank=0, world_size=1)
    monkeypatch.setattr(linear, "get_tensor_model_parallel_rank", lambda: 0)
    monkeypatch.setattr(linear, "get_tensor_model_parallel_world_size", lambda: 1)
    # 关键修复: 将 dispatch_unquantized_gemm 替换为始终返回 default_unquantized_gemm 的
    lambda
    # 这确保无论在哪个平台 (NVIDIA/ROCm), 测试中调用的 GEMM 都是平台无关的默认实现
    # 避免因 UnquantizedLinearMethod 绑定平台特定 GEMM 而导致在 CPU 张量上失败
    monkeypatch.setattr(
        linear,
        "dispatch_unquantized_gemm",
        lambda linear_backend="auto": default_unquantized_gemm,
    )
    monkeypatch.setattr(parameter, "get_tensor_model_parallel_rank", lambda: 0)
    monkeypatch.setattr(parameter, "get_tensor_model_parallel_world_size", lambda: 1)
    with torch.device("meta"):
        meta = cls()
        fuser = MergedColumnParallelFuser.match(trace(meta), meta)
        assert fuser is not None
        assert set(fuser.linear_names) == {"proj_a", "proj_b", "proj_c", "proj_d"}
        assert fuser.shards == [(name, i) for i, name in enumerate(fuser.linear_names)]
        fuser.update_forward(meta)

    real = cls()
    x = torch.randn(2, 8)
    expected = real(x)
    weights = [
        (name, weight.detach().clone()) for name, weight in real.named_parameters()
    ]
    fused = fuser.fuse(real, "parallel", default_vllm_config)
```

```

root = nn.Module()
root.parallel = fused
mapper = WeightsMapper(orig_to_new_stacked=fuser.orig_to_new_stacked("parallel"))
# 通过 AutoWeightsLoader 加载权重，内部会调用已 monkeypatch 的 dispatch_unquantized_gemm
AutoWeightsLoader(root).load_weights(
    ((f"parallel.{name}", weight) for name, weight in weights), mapper=mapper
)

merged = getattr(fused, fuser.merged_name)
assert isinstance(merged, MergedColumnParallelLinear)
# 移除了原来的 dispatch_cpu_unquantized_gemm(merged, remove_weight=False) 调用
# 因为 monkeypatch 已经在上游解决了 GEMM 后端问题
for actual, reference in zip(fused(x), expected):
    torch.testing.assert_close(actual, reference)

```

评论区精华

Issue 评论中，AndreasKaratzas 触发 CI 运行并指出了回归来源：“The regression seems to have originated from:

- <https://github.com/vllm-project/vllm/pull/55301>”。这表明该问题是 PR#55301 引入的。Review 过程中，仅有 claude[bot] 自动评论（因 fork 限制禁用自动 review）和 AndreasKaratzas 的最终批准，没有进一步的技术讨论。
- 暂无高价值评论线程

风险与影响

- 风险：风险较低。该变更仅修改测试文件，不影响生产代码。变更的核心是替换 GEMM 调度逻辑，使其更通用且平台无关。潜在风险：1) monkeypatch 可能未完全模拟真实环境下的 GEMM 行为，但测试目的是验证 fuser 逻辑而非 GEMM 本身；2) 移除 `dispatch_cpu_unquantized_gemm` 调用后，测试不再显式处理权重，但 `AutoWeightsLoader` 已经加载了权重，这在原始代码中也是一致的。
- 影响：该变更影响范围有限：1) 对用户无直接影响，仅为测试修复；2) 对系统的影响是提高了 ROCm CI 的稳定性，确保相关测试在 AMD 平台上也能通过；3) 对团队的影响是修复了因 PR#55301 引入的回归问题，提升了 CI 的可靠性。
- 风险标记：测试平台依赖，monkeypatch 复杂度

关联脉络

- PR #55301 未知（根据 Issue 评论提及为回归来源）：根据 AndreasKaratzas 在 Issue 评论中的指出，该回归源自 PR#55301。需要检查该 PR 是否修改了 GEMM 调度机制或线性层融合相关代码。
- PR #55690 [Transformers backend] Enable QKV-Fuser for Gemma4: 同属于 transformers 后端融合器（fusers）模块，涉及类似的测试或实现变更。