PR #43120 完整报告

vllm-project/vllm

[AMD][CI][BugFix] Fix Distributed Compile Unit Tests (2xH100-2xMI300) group

合并时间：2026-05-29 05:39

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43120

执行摘要

- 一句话：修复 ROCm 分布式编译单元测试的多个问题
- 推荐动作：建议技术管理者关注 PR 中平台差异处理的模式（如动态端口、条件注册），作为跨平台测试的参考；值得精读 collective_fusion.py 中的条件注册逻辑。

功能与动机

在 ROCm 平台上运行分布式编译单元测试时，发现 test_tp2_ar_rms.py 和 test_async_tp.py 存在多个失败原因：自定义 all-reduce 未正确禁用导致融合 passes 初始化跳过；ar_rms_fusion 在 ROCm 上对应 aiter_ar_rms_fusion 而非默认模式；测试端口冲突；CUTLASS scaled_mm 不可用时未跳过测试。

实现拆解

1. 融合 Pass 日志：在 vllm/compilation/passes/fusion/allreduce_rms_fusion.py 中为 RocmAiterAllReduceFusionPass 添加 __call__ 方法，应用模式匹配并记录匹配数量，便于单元测试检测。
2. 条件注册：在 vllm/compilation/passes/fusion/collective_fusion.py 的 AsyncTPPass 初始化中，注册 CUTLASS 模式前检查 torch.ops._C.cutlass_scaled_mm 是否存在，避免运行时错误。
3. 测试配置调整：在 tests/compile/fusions_e2e/test_tp2_ar_rms.py 的三个测试函数中设置 disable_custom_all_reduce=False，使 ROCm 融合 pass 正确初始化；并在 test_tp2_ar_rms_fusions 中根据平台选择 aiter_ar_rms_fusion 或 ar_rms_fusion 模式检查。
4. 模型匹配更新：在 tests/compile/fusions_e2e/models.py 中添加 aiter_ar_rms_fusion 匹配计数，确保 ROCm 测试通过。
5. 后端跳过逻辑：在 tests/compile/fusions_e2e/conftest.py 中增加 rocm_attn 与 gpt-oss-20b 的跳过条件，因为该后端不支持 attention sinks。
6. 端口冲突修复：在 tests/compile/passes/distributed/test_async_tp.py 中使用 get_open_port() 动态获取端口，避免固定端口 12345 被占用；并将 CUTLASS 相关测试参数包装为 pytest.param 并添加 skipif 条件。
7. 日志模式注册：在 tests/compile/fusions_e2e/common.py 中为 ROCm 平台将 aiter_ar_rms_fusion 加入 FUSION_LOG_PATTERNS。

关键文件：

- vllm/compilation/passes/fusion/allreduce_rms_fusion.py（模块 编译层；类别 source；类型 core-logic；符号 call）：添加了 __call__ 方法以应用模式匹配并记录日志，使单元测试能够检测融合 pass 的执行。
- vllm/compilation/passes/fusion/collective_fusion.py（模块 编译层；类别 source；类型 core-logic）：在 AsyncTPPass.__init__ 中增加 hasattr 检查，避免在无 CUTLASS 的平台上注册模式失败。
- tests/compile/passes/distributed/test_async_tp.py（模块 异步 TP 测试；类别 test；类型 test-coverage）：使用动态端口避免冲突；对 CUTLASS 测试参数添加 skipif 条件。
- tests/compile/fusions_e2e/test_tp2_ar_rms.py（模块 AR-RMS 测试；类别 test；类型 test-coverage）：设置 disable_custom_all_reduce=False 并调整模式匹配检查以支持 ROCm。
- tests/compile/fusions_e2e/models.py（模块 测试模型；类别 test；类型 test-coverage）：为各模型添加 aiter_ar_rms_fusion 匹配计数，确保 ROCm 测试通过。

关键符号：call, AsyncTPPass.init, test_async_tp_pass_replace, async_tp_pass_on_test_model, test_tp2_ar_rms_fusions, test_tp2_ar_rms_fp8_fusions, test_tp2_ar_rms_fp4_fusions

关键源码片段

vllm/compilation/passes/fusion/allreduce_rms_fusion.py

添加了 __call__ 方法以应用模式匹配并记录日志，使单元测试能够检测融合 pass 的执行。

```
# 在 RocmAiterAllReduceFusionPass 类中，新增 __call__ 方法
# 该方法被 @VllmInductorPass.time_and_log 装饰，用于计时和日志
@VllmInductorPass.time_and_log
def __call__(self, graph: fx.Graph) -> None:
    # 应用之前注册的模式匹配 pass，返回匹配到的次数
    self.matched_count = self.pm_pass.apply(graph)
    # 累加总的匹配计数到全局表中，供单元测试正则匹配验证
    VllmPatternMatcherPass.match_table[self.pass_name] += self.matched_count
    logger.debug(
        "%s Replaced %s patterns", self.__class__.__name__, self.matched_count
    )
```

vllm/compilation/passes/fusion/collective_fusion.py

在 AsyncTPPass.__init__ 中增加 hasattr 检查，避免在无 CUTLASS 的平台上注册模式失败。

```
# 在 AsyncTPPass.__init__ 中，原来无条件注册现在加入条件检查
if self.model_dtype == torch.bfloat16:
    ScaledMMReduceScatterPattern(...).register(self.pm_pass)
    AllGatherScaledMMPattern(...).register(self.pm_pass)
    # 仅当 cutlass_scaled_mm 可用时才注册相应模式
    if hasattr(torch.ops._C, "cutlass_scaled_mm"):
        CutlassScaledMMReduceScatterPattern(
```

```
            self.model_dtype, self.device
        ).register(self.pm_pass)
        AllGatherCutlassScaledMMPattern(
            self.model_dtype, self.device
        ).register(self.pm_pass)
```

tests/compile/passes/distributed/test_async_tp.py

使用动态端口避免冲突；对 CUTLASS 测试参数添加 skipif 条件。

```
# 在 test_async_tp_pass_replace 中，使用 get_open_port 获取空闲端口
from vllm.utils.network_utils import get_open_port
master_port = str(get_open_port())

# 将 master_port 传递给 spawn 函数
torch.multiprocessing.spawn(
    async_tp_pass_on_test_model,
    args=(num_processes, test_model, batch_size, seq_len, hidden_size, dtype, dynamic, master_
    port),
    nprocs=nprocs,
)

# 在 async_tp_pass_on_test_model 函数中接受 master_port 参数
def async_tp_pass_on_test_model(rank, world_size, test_model, batch_size, seq_len, hidden_size,
dtype, dynamic, master_port="0"):
    # 使用动态端口设置环境变量
    os.environ["MASTER_PORT"] = master_port
```

评论区精华

BadrBasowid 建议将新增的日志逻辑移到基类 VllmFusionPatternMatcherPass 以避免重复，但 tjtanaa 指出单元测试依赖正则匹配特定类名，因此该日志必须保留在派生类中。最终决定保持当前实现。

- 日志方法位置 (design): 决定保持当前实现，不在基类中添加日志。

风险与影响

- 风险：主要风险在于平台条件分支（is_rocm()、hasattr(torch.ops._C, "cutlass_scaled_mm")）可能遗漏其他平台或硬件变体；端口动态分配确保无冲突，但需确认 get_open_port() 在所有环境可用。CUTLASS 检查条件可能遗漏其他后端，但已有 fallback。ROCm 特定模式 aiter_ar_rms_fusion 仅用于测试，不影响生产逻辑。
- 影响：影响范围限于 ROCm 平台上的编译测试组确保通过；对 CUDA 平台测试无影响。测试覆盖更准确，平台适配更健壮。
- 风险标记：平台条件分支，CUTLASS 可用性依赖，测试端口分配

关联脉络

- 暂无明显关联 PR