PR #50802 完整报告

vllm-project/vllm

[ROCm] Fix AITER all-reduce fusion coverage

合并时间：2026-08-06 14:52

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50802

执行摘要

- 一句话：补 ROCm AITER add-RMSNorm 死残差融合，重构稳定性测试
- 推荐动作：值得精读。核心亮点是：通过继承既有 pattern 并只取第一个输出，用约 30 行代码补齐一个生产覆盖缺口；测试设计（四个独立融合站点 + 宽松 FP8 包络 + 严格 bf16 断言）可作后续融合测试模板。对 ROCm 编译路径维护者有直接参考价值，评审中端到端精度验证的答复方式也值得借鉴。

功能与动机

PR #37646 引入 AITER add-RMSNorm 融合时缺少 dead-residual 形式，导致 residual 输出无人消费的图结构无法被融合；PR #42864 引入的 grouped-FP8 测试因链式量化使最终对比不稳定。两个回归均在 distributed compile step（Buildkite amd-ci）暴露。本 PR 补充缺失的生产 pattern，并让测试直接验证每个融合点，作者还指出本地在 2 张 MI355 GPU 上完整通过。

实现拆解

1. 新增 output-only 融合 pattern：在 vllm/compilation/passes/fusion/allreduce_rms_fusion.py 中定义 AiterAllreduceFusedAddRMSNormOutputOnlyPattern，继承既有 AiterAllreduceFusedAddRMSNormPattern，pattern 与 replacement 都只取元组第一个输出（RMSNorm 结果），使 all_reduce -> fused_add_rms_norm 且 residual 无消费者的子图也能融合为单个 AITER kernel；随后在 RocmAiterAllReduceFusionPass.__init__ 中与既有 pattern 一并注册，保留原双输出 pattern 供同时消费两个输出的调用方使用。
2. 重构 grouped-FP8 测试模型：TestAiterAllReduceRMSNormGroupQuantFP8Model.forward 从链式结构改为四个独立融合站点，每个站点从同一 bf16 激活出发并独立返回，避免前一站点的量化 / 反量化误差沿链复合；_dequantize_to_bf16 去掉 ref 参数改用 self.dtype，forward 返回 y0..y3、三个 residual 和两个 indexer 输出。
3. 强化断言与操作计数：rocm_aiter_group_quant_fusion_pass_on_test_model 对前 4 个 FP8 输出改用 AITER 5%/5% 包络（允许 5% 的孤立值超出），对 residual 与 bf16 indexer 输出保留 atol=1e-2, rtol=1e-2 严格断言；check_before_ops 改为 fully_replaced=True，并分别断言 fused quant op 与 fused indexer op 各触发 2 次。
4. 修复分布式测试端口冲突：从 vllm.utils.network_utils 导入 get_open_port()，在父进程 run_torch_spawn 中选取空闲端口并传给每个 spawn 的 rank，替代硬编码 MASTER_PORT: "12345"，两处 spawn 入口（test_all_reduce_fusion_pass_replace 与

test_rocm_aiter_all_reduce_rmsnorm_group_quant_fp8_fusion_pass_replace）均同步修改。

关键文件：

- vllm/compilation/passes/fusion/allreduce_rms_fusion.py（模块 编译融合；类别 source；类型 core-logic；符号 AiterAllreduceFusedAddRMSNormOutputOnlyPattern, pattern, _pattern, replacement）：生产代码核心变更：新增 AiterAllreduceFusedAddRMSNormOutputOnlyPattern，修复 PR #37646 遗漏的 dead-residual 融合形式，并在 RocmAiterAllReduceFusionPass 中注册，直接影响 ROCm AITER 编译产出的 kernel 融合行为。
- tests/compile/passes/distributed/test_fusion_all_reduce.py（模块 融合测试；类别 test；类型 test-coverage；符号 _dequantize_to_bf16, TestAiterAllReduceRMSNormGroupQuantFP8Model, all_reduce_fusion_pass_on_test_model, rocm_aiter_group_quant_fusion_pass_on_test_model）：测试配套核心：重构 grouped-FP8 测试模型为四个独立融合站点，消除链式量化误差复合；改用动态端口并新增按输出类型区分的断言，使测试能稳定验证所有融合 pattern。

关键符号：AiterAllreduceFusedAddRMSNormOutputOnlyPattern, AiterAllreduceFusedAddRMSNormPattern.pattern, AiterAllreduceFusedAddRMSNormPattern.replacement, RocmAiterAllReduceFusionPass.init, TestAiterAllReduceRMSNormGroupQuantFP8Model.forward, TestAiterAllReduceRMSNormGroupQuantFP8Model._dequantize_to_bf16, rocm_aiter_group_quant_fusion_pass_on_test_model, all_reduce_fusion_pass_on_test_model

关键源码片段

vllm/compilation/passes/fusion/allreduce_rms_fusion.py

生产代码核心变更：新增 AiterAllreduceFusedAddRMSNormOutputOnlyPattern，修复 PR #37646 遗漏的 dead-residual 融合形式，并在 RocmAiterAllReduceFusionPass 中注册，直接影响 ROCm AITER 编译产出的 kernel 融合行为。

```
class AiterAllreduceFusedAddRMSNormOutputOnlyPattern(
    AiterAllreduceFusedAddRMSNormPattern
):
    """Match the add-RMSNorm form when its residual output is dead."""

    @property
    def pattern(self):
        pattern = super().pattern

        def _pattern(
            residual: torch.Tensor, input: torch.Tensor, weight: torch.Tensor
        ) -> torch.Tensor:
            # 复用基类的双输出 pattern，只取第一个输出（RMSNorm 结果），
```

```
            # 从而匹配 residual 输出为死值（无消费者）的图结构。
            return pattern(residual, input, weight)[0]

        return _pattern

    @property
    def replacement(self):
        replacement = super().replacement

        def _replacement(
            residual: torch.Tensor, input: torch.Tensor, weight: torch.Tensor
        ) -> torch.Tensor:
            # 替换后同样只返回融合 kernel 的 RMSNorm 输出，
            # 与原始子图的可见输出数量保持一致。
            return replacement(residual, input, weight)[0]

        return _replacement

# 在 RocmAiterAllReduceFusionPass.__init__ 中与既有 pattern 一并注册，
# 使 dead-residual 的 add-RMSNorm 子图也能命中融合。
self.register(
    AiterAllreduceFusedRMSNormPattern(
        epsilon, self.model_dtype, self.device
    )
)
self.register(
    AiterAllreduceFusedAddRMSNormPattern(
        epsilon, self.model_dtype, self.device
    )
)
self.register(
    AiterAllreduceFusedAddRMSNormOutputOnlyPattern(
        epsilon, self.model_dtype, self.device
    )
)
```

tests/compile/passes/distributed/test_fusion_all_reduce.py

测试配套核心：重构 grouped-FP8 测试模型为四个独立融合站点，消除链式量化误差复合；改用动态端口并新增按输出类型区分的断言，使测试能稳定验证所有融合 pattern。

```
def forward(self, hidden_states):
    # 每个融合站点从同一 bf16 激活 z 独立出发、独立返回，
    # 避免四个量化 / 反量化误差沿链复合导致最终比较不稳定。
    z = torch.relu(hidden_states)
    x0 = tensor_model_parallel_all_reduce(z)
    rms0 = self.norm[0](x0)
    q0, s0 = self._group_quant(rms0)
    y0 = self._dequantize_to_bf16(q0, s0)

    x1 = tensor_model_parallel_all_reduce(torch.mm(z, self.w[0]))
```

```
    rms1, resid1 = self.norm[1](x1, z.clone())
    q1, s1 = self._group_quant(rms1)
    y1 = self._dequantize_to_bf16(q1, s1)

    x2 = tensor_model_parallel_all_reduce(torch.mm(z, self.w[1]))
    rms2, resid2 = self.norm[2](x2, z.clone())
    q2, s2 = self._group_quant(rms2)
    # 第二个消费者 rocm_unquantized_gemm 强制命中 with-indexer 变体。
    idx2 = torch.ops.vllm.rocm_unquantized_gemm(rms2, self.indexer_w[0], None)
    y2 = self._dequantize_to_bf16(q2, s2)

    x3 = tensor_model_parallel_all_reduce(torch.mm(z, self.w[2]))
    rms3, resid3 = self.norm[3](x3, z.clone())
    q3, s3 = self._group_quant(rms3)
    # rms3 同样带 indexer 消费者，确保两类 indexer pattern 均被覆盖。
    idx3 = torch.ops.vllm.rocm_unquantized_gemm(rms3, self.indexer_w[1], None)
    y3 = self._dequantize_to_bf16(q3, s3)
    return y0, y1, y2, y3, resid1, resid2, resid3, idx2, idx3

# AITER 的 kernel 级融合允许集体通信重排后少量 FP8 值偏离，
# 因此对 FP8 输出使用 5% 包络并允许 5% 的孤立值超出。
for site, (reference, actual) in enumerate(
    zip(results_unfused[:4], results_fused[:4])
):
    close = torch.isclose(reference, actual, atol=5e-2, rtol=5e-2)
    mismatch_ratio = float((~close).float().mean().item())
    assert mismatch_ratio <= 0.05, (
        f"FP8 site {site} mismatch ratio {mismatch_ratio:.2%} exceeds 5%"
    )

# residual 与 bf16 indexer 输出不经过 FP8 量化，保留严格精度断言。
torch.testing.assert_close(
    results_unfused[4:], results_fused[4:], atol=1e-2, rtol=1e-2
)
```

评论区精华

- tjtanaa（评审人）在 allreduce_rms_fusion.py:1248 要求端到端验证："Let's show that this pattern is correctly implemented and does not cause accuracy degradation in end to end inferencing as the condition is much more complex in e2e. Passing unit tests does not mean it is working well end2end. Please pick a model that has this pattern."
- AndreasKaratzas 回复了具体验证数据：使用 meta-llama/Meta-Llama-3.1-8B-Instruct（model_impl="transformers"、BF16、TP2 on 2x MI300X、ROCM_ATTN、AITER 自定义 all-reduce、禁用编译缓存），编译匹配表确认融合站点从 64 增至 65 个 /rank，5-shot MMLU 评估 1140 题：parent 均值 66.71%，PR 均值 66.27%，delta 为 -0.44 个百分点。
- dllehr-amd 在 issue 评论区询问该测试是否在 AMD 硬件上运行，作者回应有 draft PR #50519 用于评估缺失覆盖；该讨论已闭环。

- 新融合 pattern 的端到端精度验证 (correctness): 作者用 Llama-3.1-8B-Instruct 在 2x MI300X TP2 上验证，融合站点从 64 增至 65/rank，5-shot MMLU 从 66.71% 变为 66.27%（-0.44 个百分点），评审人接受并批准。
- 分布式融合测试是否在 AMD 硬件上运行 (question): 作者回应已有 draft PR #50519 用于评估缺失覆盖，当前测试尚未纳入 AMD CI。

风险与影响

- 风险：
  - 核心编译路径变更：新 pattern 会改变 ROCm 上所有带 dead-residual add-RMSNorm 子图模型的编译输出，若 AITER 融合 kernel 与 unfused 链存在细微语义差异（如 all-reduce 与 RMSNorm 重排），可能引入精度波动；e2e 验证仅覆盖 Llama-3.1-8B 一个模型，对 DeepSeek 等更大模型的风险未充分覆盖。
  - 测试覆盖缺口：dllehr-amd 指出该分布式融合测试当前未在 AMD CI 上运行，回归可能在合入后不被及时发现，作者期望通过 draft PR #50519 补齐。
  - 断言容差较宽松：FP8 输出使用 5% 包络并允许 5% 孤立值超出，可能掩盖部分真实精度退化；不过 residual 与 indexer 输出仍保持 1e-2 严格断言。
  - 端口分配存在 TOCTOU 竞态：get_open_port() 在获取与绑定之间端口可能被复用，但测试并发度低，风险有限。
- 影响：
  - 用户 / 系统：ROCm MI300X/MI355X 上编译带 fused_add_rms_norm 的模型（如 DeepSeek、Llama 类）时，更多子图被融合为单个 AITER kernel，减少 kernel 启动次数，可能带来性能收益；此前未匹配时该子图会退化为多 kernel 执行。
  - 团队：为 ROCm 编译路径确立了更稳健的融合测试方法（独立站点 + 匹配计数 + 操作计数），后续回归更易被定位；MMLU 端到端验证流程也可复用。
  - 影响范围：严格限于 ROCm + AITER 编译路径，不触及 CUDA/XPU 等其他平台，默认行为对非 ROCm 平台无影响。
  - 风险标记：核心编译路径变更，E2E 验证仅覆盖单一模型，测试未纳入 AMD CI，FP8 断言容差宽松

关联脉络

- PR #37646 Add AITER allreduce add-RMSNorm fusion pattern: PR body 指出该 PR 引入的 add-RMSNorm 融合缺少 dead-residual 形式，是本 PR 修复的回归来源。
- PR #42864 Add grouped-FP8 all-reduce fusion distributed test: PR body 指出该 PR 引入的 grouped-FP8 测试因链式量化导致最终比较不稳定，本 PR 重构了该测试模型。
- PR #41825 Wire allreduce + rms_norm half of AITER fusion: AiterAllreduceFusedRMSNormGroupQuantFP8Pattern 的 docstring 提及该 PR 接通 all_reduce + rms_norm 融合，与本 PR 同属 RocmAiterAllReduceFusionPass 覆盖范围。
- PR #50519 Draft: evaluate missing AITER fusion coverage on AMD hardware: issue 评论中作者提到该 draft PR 用于评估缺失覆盖，与本 PR 的测试覆盖目标一致。