PR #53004 完整报告

vllm-project/vllm

[ROCm][CI] Speed up `test_rocm_aiter_qk_norm_rope_kvcache_fusion`

合并时间：2026-08-20 10:20

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/53004

执行摘要

- 一句话：裁剪 ROCm 融合测试参数化，耗时从 2.5 小时降至 6 分钟
- 推荐动作：值得一读，尤其是大型参数化测试的加速思路：裁剪冗余参数、用模块级清理替代逐用例清理、收集期跳过与版本门槛。不过要关注覆盖度变化，未来若要新增参数组合应充分评估必要性。

功能与动机

原测试因 1440 个参数化组合耗时 2.5 小时以上，成为 ROCm CI 的瓶颈。PR body 说明：use_shuffle_kv_layout="1" 的所有用例实际都被跳过，保留该维度没有意义；rms_norm_eps=1e-5 与 1e-6 的差异不带来实际收益；num_tokens 只需覆盖 5（小请求）与 2048（大序列）两个边界。同时，AITER < 0.1.20 时融合内核存在已知 abort，需要版本门槛避免 CI 反复失败。

实现拆解

1. 参数化裁剪：移除 use_shuffle_kv_layout 参数（含函数签名中的对应入参与 skip 分支）；num_tokens 从 [5, 16, 2048] 收敛为 [5, 2048]；rms_norm_eps 从 [1e-5, 1e-6] 收敛为 [1e-6]。组合数从 1440 降至 480。
2. 清理策略变更：新增 pytestmark = pytest.mark.skip_global_cleanup 跳过逐用例的全局清理；新增模块级 autouse fixture module_global_cleanup，只在模块结束时统一调用 cleanup_dist_env_and_memory()，避免每个用例重复初始化 / 销毁分布式环境带来的开销。
3. 收集期跳过与版本门槛：在模块加载时若 is_aiter_found_and_supported() 为假，则用 pytest.skip(allow_module_level=True) 直接跳过整个模块；skipif 条件从仅检查 AITER 存在改为 Version(version("amd_aiter")) >= Version("0.1.20")，并给出旧版本内核 abort 的原因说明。
4. 配套导入：新增 importlib.metadata.version 和 packaging.version.Version 用于版本比较；测试函数签名中移除 use_shuffle_kv_layout 参数并删除对应 skip 分支。

关键文件：

- tests/compile/passes/test_rocm_aiter_qk_norm_rope_kvcache_fusion.py（模块 编译测试；类别 test；类型 test-coverage；符号 module_global_cleanup, test_qk_norm_rope_kvcache_fusion, _run_qk_norm_rope_kvcache_fusion_test）：唯一变更文件。通过参数化裁剪（1440 → 480）、模块级清理与 AITER 版本门槛，将该测试

运行时间从 2.5 小时以上降至约 6 分钟，同时保持核心融合路径覆盖。

关键符号：module_global_cleanup, test_qk_norm_rope_kvcache_fusion, _run_qk_norm_rope_kvcache_fusion_test

关键源码片段

tests/compile/passes/test_rocm_aiter_qk_norm_rope_kvcache_fusion.py

唯一变更文件。通过参数化裁剪（1440 → 480）、模块级清理与 AITER 版本门槛，将该测试运行时间从 2.5 小时以上降至约 6 分钟，同时保持核心融合路径覆盖。

```
# tests/compile/passes/test_rocm_aiter_qk_norm_rope_kvcache_fusion.py
# 本文件验证 QkNormRopeKvCacheFusionPass 在 ROCm AITER 上的融合正确性。
# 本次变更的三大提速点：参数化裁剪、模块级清理、AITER 版本门槛。

import os
from importlib.metadata import version

import pytest
import torch
from packaging.version import Version

import vllm.config
# ... 其余导入省略 ...

# 提速点 1：标记整个模块跳过逐用例的全局清理（原实现每个用例都清理分布式环境）
pytestmark = pytest.mark.skip_global_cleanup

# 提速点 2：AITER 不可用时，在收集阶段直接跳过整个模块，避免无意义的用例执行
if not is_aiter_found_and_supported():
    pytest.skip(
        "ROCm with supported AITER is required",
        allow_module_level=True,
    )

# 提速点 3：模块级 autouse fixture，只在模块结束时统一释放分布式环境与显存
@pytest.fixture(scope="module", autouse=True)
def module_global_cleanup():
    from vllm.distributed import cleanup_dist_env_and_memory
    yield
    cleanup_dist_env_and_memory()

# ... 模型定义与辅助函数省略 ...

# 融合配置仅保留 5 种真实生效的 head 组合（覆盖 full/partial rotary 与 neox 变体）
_FUSION_CONFIGS = [
    pytest.param(64, 8, 64, 64, True, id="full-neox"),
    pytest.param(64, 8, 64, 64, False, id="full-non_neox"),
    pytest.param(32, 8, 128, 64, True, id="glm4_moe"),
```

```
    pytest.param(32, 2, 128, 64, False, id="glm4_dense"),
    pytest.param(16, 2, 64, 32, True, id="partial_small_head"),
]

@pytest.mark.parametrize(
    "num_heads, num_kv_heads, head_size, rotary_dim, is_neox",
    _FUSION_CONFIGS,
)
@pytest.mark.parametrize(
    "attn_backend",
    [
        AttentionBackendEnum.ROCM_AITER_UNIFIED_ATTN,
        AttentionBackendEnum.ROCM_AITER_FA,
    ],
)
# 参数裁剪：num_tokens 只保留 5 和 2048，去除中间值 16，减少 1/3 组合
@pytest.mark.parametrize("num_tokens", [5, 2048])
@pytest.mark.parametrize("enable_aiter_triton_rope", [True, False])
@pytest.mark.parametrize("block_size", [16, 32, 64])
@pytest.mark.parametrize("kv_cache_dtype", ["auto", "fp8"])
# 参数裁剪：rms_norm_eps 只保留 1e-6，1e-5 与 1e-6 无实用差异
@pytest.mark.parametrize("rms_norm_eps", [1e-6])
@pytest.mark.parametrize("custom_op", ["+rotary_embedding", "+rms_norm"])

# AITER < 0.1.20 的 fused_qk_norm_rope_cache 内核会因块内 KV 缓存不连续而 abort，
# 因此把 skipif 条件从仅检查 AITER 存在提升为版本门槛。
@pytest.mark.skipif(
    not Version(version("amd_aiter")) >= Version("0.1.20"),
    reason="Requires AITER >= 0.1.20; older kernels abort on fused_qk_norm_rope_cache",
)
def test_qk_norm_rope_kvcache_fusion(...):
    # ... 用例主体省略 ...
    pass
```

评论区精华

本 PR 来自 fork，claude[bot] 自动审查被禁用，未产生实际代码评审建议。维护者 AndreasKaratzas 在最后直接批准（LGTM）。PR body 是主要设计说明，作者对参数裁剪理由的逐条解释（跳过集合、无用 epsilon、边界 token 数）是整个变更的核心依据。无未解决的讨论线程。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 覆盖度小幅下调：移除 num_tokens=16、rms_norm_eps=1e-5 以及 use_shuffle_kv_layout=1 组合。前两者对融合 pass 的行为影响很小，且

use_shuffle_kv_layout=1 原本就全部跳过，风险可接受。

- 清理时机变化：从逐用例清理改为模块级一次性清理，若未来测试中新增跨用例状态污染，可能更难定位。当前该文件仅此一个测试函数，风险有限。
- 版本判断强依赖：version("amd_aiter") 依赖包名与 packaging 解析，若某环境包名注册不同，会抛 PackageNotFoundError。好在 is_aiter_found_and_supported() 已先行拦截未安装场景，但已安装但注册名不同时仍需关注。
- 收集期跳过易被忽略：模块级 pytest.skip 会导致测试在 CI 报告中直接消失，若环境配置变化导致 AITER 不可用，测试缺失可能不易被察觉。
- 影响：对最终用户无任何影响（纯测试变更）。对 ROCm CI 是显著正向影响：单测试从 2.5 小时降至约 6 分钟，能大幅缩短 AMD 构建队列的反馈周期。对维护该测试的工程师，需要理解参数裁剪的背景与版本门槛的含义，否则后续可能误增冗余参数。
- 风险标记：覆盖度削减，清理时机变更，依赖版本判断，收集期模块跳过

关联脉络

- 暂无明显关联 PR