

# PR #49915 完整报告

vllm-project/vllm

[CI][ROCm] Reduce kernel test runtime

合并时间: 2026-07-27 14:44

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49915>

## 执行摘要

- 一句话: ROCm 内核测试运行时优化, 过滤无效配置并提升并行度
- 推荐动作: 建议精读 `test_fused_quant_layernorm.py` 中的参数化重构方法, 该模式可推广到其他使用大量 `pytest.skip` 的测试套件中。CI 并行度调整可作为后续同类优化的参考。

## 功能与动机

ROCm CI 内核测试运行时间过长, PR body 明确指出需要减少内核测试 runtime, 对应的 Buildkite 链接显示相关流水线耗时偏高。通过过滤无效配置、提前跳过不支持的测试以及调整并行度来缩短 CI 反馈周期。

## 实现拆解

1. 过滤无效 RMSNorm 测试配置 (`tests/kernels/core/test_fused_quant_layernorm.py`):
  - 提取 `_is_valid_config` 函数, 将原来在测试函数体内部通过 `pytest.skip` 跳过的逻辑提前到参数化收集阶段, 减少不必要的测试项。
  - 新增 `_config_id` 函数生成可读性更高的参数 ID, 方便定位失败用例。
  - 移除 `GROUP_SIZES` 列表中的 `None`, 改为在 `itertools.product` 中显式包含 `(None, 0)`, 消除与原有组合的重复并避免无效的 `(None, 4)` 组合。
2. 模块级跳过 FP8 测试 (`tests/kernels/core/test_vit_fp8_scaling.py`):
  - 添加 `pytestmark = pytest.mark.skipif`, 在模块加载阶段统一跳过不满足 FlashInfer cuDNN FP8 预填充注意力支持的测试, 替代原来在每个 context manager 内部分支跳过, 减少执行开销。
3. CI 并行度提升 (`.buildkite/test-amd.yaml`):
  - 将 Kernels Core Operation Test 的并行度从 2 提升到 3, 并添加 `--shard-id` 分片参数。
  - 将 Transformers Nightly Models (Processing) 的并行度从 4 提升到 8。
  - 为 Kernels Core Operation Test 标签添加 `%N`, 在 Buildkite 中显示分片序号。

关键文件:

- `tests/kernels/core/test_fused_quant_layernorm.py` (模块 内核测试; 类别 test; 类型 test-coverage; 符号 `_is_valid_config`, `_config_id`): 核心变更文件, 通过提取 `_is_valid_config` 和 `_config_id` 函数在参数化收集阶段过滤无效组合, 避免运行时 skip 开销, 是降低测试运行时间的主要手段。

- tests/kernels/core/test\_vit\_fp8\_scaling.py (模块 内核测试; 类别 test; 类型 test-coverage) : 添加模块级 skipif, 避免在不受支持的平台上执行 FP8 缩放测试, 减少无意义的测试运行。
- .buildkite/test-amd.yaml (模块 CI 配置; 类别 config; 类型 configuration) : 调整 CI 并行度和分片配置, 直接提高测试并行执行效率。

关键符号: `_is_valid_config`, `_config_id`

## 关键源码片段

### tests/kernels/core/test\_fused\_quant\_layernorm.py

核心变更文件, 通过提取 `_is_valid_config` 和 `_config_id` 函数在参数化收集阶段过滤无效组合, 避免运行时 skip 开销, 是降低测试运行时间的主要手段。

# tests/kernels/core/test\_fused\_quant\_layernorm.py

```
def _is_valid_config(
    hidden_size: int,
    has_scale_ub: bool,
    quant_dtype: torch.dtype,
    group_size: list[int] | None,
    tma_alignment: int,
) -> bool:
    # 隐藏层大小不能被 group 大小整除时无效
    if group_size is not None and hidden_size % group_size[1] != 0:
        return False
    # group 量化不支持 scale_ub
    if group_size is not None and has_scale_ub:
        return False
    # TMA alignment 只支持 groupwise FP8
    if (group_size is None or quant_dtype != current_platform.fp8_dtype()) and tma_alignment != 0:
        return False
    # 当 TMA alignment 不增加 padding 时跳过 (不改变数据布局)
    if group_size is not None and tma_alignment != 0 and hidden_size // group_size[1] % tma_alignment == 0:
        return False
    # scale_ub 只对 FP8 有意义
    return not (has_scale_ub and quant_dtype != current_platform.fp8_dtype())
```

```
def _config_id(
    num_tokens: int,
    hidden_size: int,
    has_scale_ub: bool,
    quant_dtype: torch.dtype,
    group_size: list[int] | None,
    tma_alignment: int,
```

```

) -> str:
    quant = str(quant_dtype).removeprefix("torch.")
    group = "per-token" if group_size is None else f"group-{group_size[1]}"
    return (
        f"{num_tokens}x{hidden_size}-{quant}-{group}"
        f"tma-{tma_alignment}-scale-ub-{has_scale_ub}"
    )

# 在收集阶段过滤，避免每个用例调用 pytest.skip 触发全局 teardown
RMS_NORM_CONFIGS = [
    pytest.param(
        num_tokens, hidden_size, has_scale_ub, quant_dtype, group_size, tma_alignment,
        id=_config_id(num_tokens, hidden_size, has_scale_ub, quant_dtype, group_size, tma_
        alignment),
    )
    for (num_tokens, hidden_size), has_scale_ub, quant_dtype, (group_size, tma_alignment)
    in itertools.product(
        NUM_TOKENS_HIDDEN_SIZES, SCALE_UBS, QUANT_DTYPES,
        [(None, 0), *itertools.product(GROUP_SIZES, TMA_ALIGNMENTS)],
    )
    if _is_valid_config(hidden_size, has_scale_ub, quant_dtype, group_size, tma_alignment)
]

```

## tests/kernels/core/test\_vit\_fp8\_scaling.py

添加模块级 skipif，避免在不受支持的平台上执行 FP8 缩放测试，减少无意义的测试运行。

```

# tests/kernels/core/test_vit_fp8_scaling.py

# 模块级标记：若 FlashInfer cuDNN FP8 预填充注意力不受支持则跳过整个文件
pytestmark = pytest.mark.skipif(
    not is_flashinfer_cudnn_fp8_prefill_attn_supported(),
    reason="FlashInfer cuDNN FP8 prefill attention not supported",
)

```

## 评论区精华

Review 中 [tjtanaa](#) 指出 `GROUP_SIZES` 中原有 `None` 被移除，质疑是否丢失了 per-token 测试覆盖。[AndreasKaratzas](#) 解释已将 `None` 移动至 `itertools.product` 的组合列表中，同时消除了原先产生的重复和无效组合（如 `(None, 0)` 重复、`(None, 4)` 总是被 skip）。该讨论已解决。

- `GROUP_SIZES` 中 `None` 被移除引发的覆盖性讨论 (correctness): 确认测试覆盖未丢失，反而消除了冗余和无效组合。

## 风险与影响

- 风险:

1. 测试覆盖风险：过滤逻辑 `_is_valid_config` 可能遗漏某些边界组合，但原作者通过保留 `(None, 0)` 组合确保了 per-token 覆盖，且逻辑是从原来的 skip 条件直接转换，风险较低。
2. 并行度风险：增加并行度可能因资源竞争导致偶发失败，但 CI 环境通常隔离且原并行度较低，适度提升风险可控。
3. 模块级 skipif: `test_vit_fp8_scaling.py` 中 skipif 条件判断依赖的函数 `is_flashinfer_cudnn_fp8_prefill_attn_supported` 在模块导入时执行，若该函数有副作用或延迟，可能影响其他测试，但概率很小。
  - 影响：该 PR 主要影响 ROCm CI 的测试流水线，预期可显著缩短内核测试和 Transformer 处理测试的执行时间。对用户无直接功能影响，对开发者体验有正向改进。团队可更快获得 CI 反馈，减少等待时间。
  - 风险标记：测试过滤可能遗漏边界组合，并行度增加可能资源争用

## 关联脉络

- 暂无明显关联 PR