

PR #50340 完整报告

vllm-project/vllm

[CI][ROCm] Stabilize LLM GC teardown check

合并时间: 2026-07-30 18:42

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/50340>

执行摘要

- 一句话: 稳定 ROCm 上 LLM GC 弱引用测试
- 推荐动作: 值得合并, 方案简洁有效。重试条件精确, 不会弱化测试的检测能力。

功能与动机

ROCm 平台下 fixture 清理后 LLM 对象偶尔未被立即 GC, 导致 weakref 断言间歇性失败。直接强制循环收集会掩盖测试本意检测的循环引用问题。

实现拆解

在 `tests/basic_correctness/test_basic_correctness.py` 的 `test_vllm_gc_ed` 函数上添加 `@pytest.mark.flaky` 装饰器, 仅在 ROCm 条件下对 `AssertionError` 进行最多 2 次重试, 每次间隔 5 秒。

关键文件:

- `tests/basic_correctness/test_basic_correctness.py` (模块测试; 类别 test; 类型 test-coverage; 符号 `test_vllm_gc_ed`): 为 `test_vllm_gc_ed` 添加 pytest flaky 重试, 仅针对 ROCm 平台的 `AssertionError`。

关键符号: `test_vllm_gc_ed`

关键源码片段

`tests/basic_correctness/test_basic_correctness.py`

为 `test_vllm_gc_ed` 添加 pytest flaky 重试, 仅针对 ROCm 平台的 `AssertionError`。

```
# ROCm 可以偶尔在 fixture 清理后保留对象。仅重试该断言;
# 在此处收集循环垃圾会掩盖此测试旨在捕获的循环引用。
@pytest.mark.flaky(
    reruns=2,
    reruns_delay=5,
    only_rerun="AssertionError", # 只对断言错误重试
    condition=current_platform.is_rocm(), # 仅 ROCm 平台生效
)
def test_vllm_gc_ed():
    """Verify vllm instance is GC'ed when it is deleted"""
```

```
llm = LLM("hmellor/tiny-random-LlamaForCausalLM")
weak_llm = weakref.ref(llm)
del llm
# 如果有任何对 vllm 的循环引用，此断言会失败
assert weak_llm() is None
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：仅修改测试逻辑，不影响生产代码。重试条件精确限定在 ROCm 平台和断言错误，不会掩盖其他类型的失败。
- 影响：影响范围仅限于 ROCm CI 中 `test_vllm_gc_ed` 测试的稳定性，减少因偶发 GC 延迟导致的误报。
- 风险标记：暂无

关联脉络

- PR #50284 [CI] Stabilize speculator memory teardown: 类似地通过重试机制稳定 ROCm CI 中的内存清理测试。