PR #52730 完整报告

vllm-project/vllm

[XPU][CI] fix hf runner

合并时间：2026-08-19 11:30

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52730

执行摘要

- 一句话：修复 XPU 测试中 HfRunner 退出内存阈值超时
- 推荐动作：值得快速浏览，作为一个典型的跨平台测试基础设施修复示例：当某个平台的内存回收语义与已有特殊处理平台一致时，直接复用其基线记录机制是最低成本的方案。可关注 tests/utils.py 中 record_gpu_memory_usage_stats 与 wait_for_memory_to_settle 的具体实现，理解动态阈值的设计约束。

功能与动机

PR body 指出 CI 失败的直接原因：During hf_runner.__exit__, the default threshold is 0.100; GPU0 usage was 17.74 GiB (76%), far exceeding the threshold, resulting in a timeout error after 240 seconds. 修复目标是让 XPU 平台在 HfRunner 进入时记录内存基线，从而在退出时基于实际占用情况计算动态阈值，而非使用固定的 10% 阈值。

实现拆解

1. 扩展平台判断：在 tests/conftest.py 的 HfRunner.__enter__ 中，将条件从 current_platform.is_rocm() 改为 current_platform.is_rocm() or current_platform.is_xpu()，使 XPU 与 ROCm 一样在进入时记录物理设备内存使用统计。
2. 同步更新注释与语义：将注释由“ROCm”改为“ROCm/XPU”，并补充说明 vllm worker 进程仍存活并持有 GPU 内存的场景，使后续维护者理解该基线是为 __exit__ 的惰性内存回收等待服务的。
3. 退出逻辑保持不变：HfRunner.__exit__ 仍调用 wait_for_memory_to_settle(threshold_ratio=getattr(self, "threshold_ratios", None))，但 XPU 现在能获得动态阈值，而不再回退到默认固定值。
4. CI 验证：通过 issue 评论触发 Buildkite CI 进行验证，并重跑失败作业，最终由维护者批准合入。

关键文件：

- tests/conftest.py（模块 测试框架；类别 test；类型 test-coverage）：唯一变更文件，修复 HfRunner 在 XPU 上的内存基线缺失问题，是本次 CI 超时的直接根因所在。

关键符号：HfRunner.enter, HfRunner.exit

关键源码片段

tests/conftest.py

唯一变更文件，修复 HfRunner 在 XPU 上的内存基线缺失问题，是本次 CI 超时的直接根因所在。

```
# HfRunner.__enter__: 在 ROCm/XPU 上记录开启时的内存基线。
# 这样 __exit__ 时 wait_for_memory_to_settle 会基于“进入时已占用内存 + 5%”的动态阈值，
# 而不是使用固定的 10% 默认阈值（该默认值在 GPU 已被大量占用时会超时）。
def __enter__(self):
    if current_platform.is_rocm() or current_platform.is_xpu():
        # 记录 ROCm/XPU 上的起始内存占用，便于关闭时等待内存回落到该水平。
        # 在 vllm worker 进程仍存活并持有 GPU 内存的情况下，
        # hf_runner.__exit__ 调用时固定阈值会失效，因此必须动态计算。
        from tests.utils import (
            get_physical_device_indices,
            record_gpu_memory_usage_stats,
        )

        if (device_count := current_platform.device_count()) > 0:
            devices = get_physical_device_indices(devices=list(range(device_count)))
            mem_usage_stats = record_gpu_memory_usage_stats(devices=devices)
            self.threshold_ratios = {
                # 每个设备使用“当前使用率 + 5% 缓冲”作为退出等待阈值
                device: 0.05 + mem_used / mem_tot
                for device, (mem_used, mem_tot) in mem_usage_stats.items()
            }
    return self


def __exit__(self, exc_type, exc_value, traceback):
    from tests.utils import wait_for_memory_to_settle

    del self.model
    cleanup_dist_env_and_memory()
    # ROCm/XPU 释放 VRAM 是惰性的，等待内存回落到基线，
    # 避免紧随其后的 HF 模型启动时因内存守卫检查而 OOM。
    wait_for_memory_to_settle(
        threshold_ratio=getattr(self, "threshold_ratios", None)
    )
    if hasattr(self, "threshold_ratios"):
        del self.threshold_ratios
```

评论区精华

Review 层面没有实质技术讨论：claude[bot] 提示该 PR 来自 fork，自动审核被禁用，需要维护者手动触发；jikunshang 直接批准。Issue 评论里主要是 /ci run、/ci retry 等 CI 触发指令，说明整个 PR 的讨论焦点在验证 CI 修复效果上，而非代码设计权衡。

- fork PR 自动审核 (other): 无实质代码讨论，维护者 jikunshang 直接批准。

风险与影响

- 风险：风险很低，但需注意：HfRunner.__enter__ 的内存基线采用“已用内存 + 5%”动态阈值，如果 XPU 上测试结束时显存释放的行为与 ROCm 不同，仍可能出现阈值偏紧或偏松的情况；另外，该改动仅覆盖 HfRunner，其他测试 runner（如 VllmRunner）未同步调整，后续若遇到类似超时需排查是否还有遗漏的平台分支。整体看，影响局限在测试基础设施，不涉及生产推理路径。
- 影响：影响范围限定在测试基础设施：XPU/Intel CI 上使用 HfRunner 的测试不再因固定 10% 内存阈值而超时，CI 稳定性提升。对生产用户无影响，对 ROCm 既有行为无影响（条件扩展为或关系，原逻辑不变）。团队内其他平台（如 CPU、CUDA）未被改动，可视为按需扩展平台覆盖。
- 风险标记：测试基础设施变更，平台分支扩展，动态阈值依赖平台行为

关联脉络

- PR #52672 [XPU] upgrade requirements/test/xpu.txt: 同属 XPU 测试 /CI 稳定性修复批次，都在降低 Intel GPU 平台测试的失败率。
- PR #52842 [CI][Bugfix] Complete DeepSeek-V4 FSE test fixture contract: 同样以修复 CI 回归为目标，涉及测试 fixture 契约补全，与本 PR 同属测试基础设施修复。