PR #43023 完整报告

vllm-project/vllm

[ROCm][CI] Stabilize runner teardown between sampler tests

合并时间：2026-05-23 12:19

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43023

执行摘要

- 一句话：修复 ROCm 上 VllmRunner 连续测试时的 VRAM 泄漏
- 推荐动作：值得合并，改动简洁且针对性强。建议后续确认 wait_for_gpu_memory_to_clear 的超时机制在高负载环境下是否足够。

功能与动机

PR Body 指出 MI250 sampler 夜间测试失败原因不是断言失败，而是第一个 *engine* 通过后，后续 *engine* 在构造 LLM 时因 VRAM 仍被占用而失败。日志中也有 process-group teardown 警告。需要在 *engine* 进程关闭时清理分布式状态并等待 ROCm 延迟的显存释放。

实现拆解

1. EngineCore shutdown 增强（vllm/v1/engine/core.py）：在 shutdown 方法中调用 *cleanup_dist_env_and_memory*() 来拆除分布式进程组并释放缓存的内存（包括 *gc.unfreeze*()）。保留原有的 *gc.unfreeze*() 调用以确保冻结对象可见。
2. VllmRunner 退出时等待 ROCm 显存释放（tests/conftest.py）：在 VllmRunner.__exit__ 中新增 _wait_for_rocm_memory_release 方法，仅在 ROCm 平台生效，通过 wait_for_gpu_memory_to_clear 等待所有设备上已用显存低于 1-gpu_memory_utilization 比例，超时 120 秒。

关键文件：

- vllm/v1/*engine*/*core*.py（模块 引擎核心；类别 source；类型 dependency-wiring）：在 EngineCore.shutdown 中调用 *cleanup_dist_env_and_memory* 确保分布式环境正确清理。
- tests/conftest.py（模块 测试夹具；类别 test；类型 test-coverage；符号 _wait_for_rocm_memory_release）：在 VllmRunner 退出时添加 ROCm 专用显存等待逻辑，避免下一测试因显存不足失败。

关键符号：EngineCore.shutdown, VllmRunner._wait_for_rocm_memory_release, VllmRunner.exit

关键源码片段

vllm/v1/*engine*/*core*.py

在 EngineCore.shutdown 中调用 *cleanup_dist_env_and_memory* 确保分布式环境正确清理。

```
# vllm/v1/engine/core.py
# 导入新增：cleanup_dist_env_and_memory 用于集中清理分布式状态
from vllm.distributed import (
    cleanup_dist_env_and_memory,
    stateless_destroy_torch_distributed_process_group,
)

# ...

class EngineCore:
    # ...
    def shutdown(self):
        self.structured_output_manager.clear_backend()
        if self.model_executor:
            self.model_executor.shutdown()
        if self.scheduler:
            self.scheduler.shutdown()

        # 解冻 GC 堆，让 model weights、KV caches 等对象可被回收
        gc.unfreeze()
        # 拆除分布式进程组并释放缓存，内部也会调用 gc.unfreeze()
        cleanup_dist_env_and_memory()
    # ...
```

tests/conftest.py

在 VllmRunner 退出时添加 ROCm 专用显存等待逻辑，避免下一测试因显存不足失败。

```
# tests/conftest.py
class VllmRunner:
    # ...
    def _wait_for_rocm_memory_release(self, gpu_memory_utilization: float) -> None:
        # 仅在 ROCm 平台生效
        from tests.utils import wait_for_gpu_memory_to_clear
        from vllm.platforms import current_platform

        if not current_platform.is_rocm():
            return

        num_gpus = torch.accelerator.device_count()
        if num_gpus == 0:
            return

        # V1 启动需要 free_memory >= total * gpu_memory_utilization，
        # 这里等待已用显存降至 (1 - gpu_memory_utilization) 以下。
        # 设置 120 秒超时防止无限等待。
        wait_for_gpu_memory_to_clear(
            devices=list(range(num_gpus)),
            threshold_ratio=1.0 - gpu_memory_utilization,
            timeout_s=120,
```

```
        )

    def __exit__(self, exc_type, exc_value, traceback):
        # 在 shutdown 前获取 gpu_memory_utilization 配置
        gpu_memory_utilization = (
            self.llm.llm_engine.vllm_config.cache_config.gpu_memory_utilization
        )
        try:
            self.llm.llm_engine.engine_core.shutdown()
        except Exception:
            pass
        del self.llm
        cleanup_dist_env_and_memory()
        # 等待 ROCm 显存释放，避免后续测试 OOM
        self._wait_for_rocm_memory_release(gpu_memory_utilization)
```

评论区精华

gemini-code-assist[bot] 指出：删除 gc.unfreeze() 会重新引入内存泄漏。AndreasKaratzas 回应：cleanup_dist_env_and_memory() 内部已调用 gc.unfreeze()，但仍决定恢复显式 gc.unfreeze() 调用以保持 EngineCore 冻结 / 解冻不变式清晰。最终提交中保留了 gc.unfreeze()。

- 移除 gc.unfreeze() 导致内存泄漏 (correctness): AndreasKaratzas 解释 cleanup_dist_env_and_memory() 内部已调用 gc.unfreeze()，但为了明确性仍恢复了显式调用。

风险与影响

- 风险：低风险。cleanup_dist_env_and_memory() 是已有函数，仅在 EngineCore 退出时调用，不影响正常推理路径。_wait_for_rocm_memory_release 只针对 ROCm，且通过超时避免死锁。若 gpu_memory_utilization 读取失败会触发异常，但已有 try/except 包裹。
- 影响：直接影响 ROCm CI 的 sampler 测试稳定性，消除连续测试间显存竞争导致的间歇性失败。对其他平台无影响（函数内部有平台判断）。
- 风险标记：暂无

关联脉络

- PR #41577 [ROCm][CI] Fix ROCm LoRA Transformers fallback with full CUDA graphs: 同为 ROCm CI 稳定性相关 PR，涉及 CUDA Graph 兼容性修复，与本 PR 属于同一稳定性改进系列。
- PR #43017 [ROCm][CI] Stabilize Granite tool-use and test URL construction: 同为 ROCm CI 稳定性 PR，关注工具调用测试，与本 PR 目标一致。