PR #47003 完整报告

vllm-project/vllm

[ROCm][CI] Use spawn around the threaded OTLP test

合并时间：2026-06-30 05:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47003

执行摘要

- 一句话：ROCm CI OTLP 测试改用 spawn 并清理资源
- 推荐动作：推荐阅读该 PR，它展示了如何处理 fork 与 gRPC 线程冲突的典型方法：通过检测平台切换到 spawn 模式，并注重资源释放。对于在其他平台需类似调整的场景有参考价值。

功能与动机

PR body 指出：测试启动 fake OTLP/gRPC 后父进程已有 gRPC 工作线程，当 vLLM 通过 fork 启动 engine worker 时，该 fork 模式在 ROCm CI 上可能导致 engine 启动或关闭时 segfault。PR 的目标是测试 span 导出行为，而非 multiprocessing 启动模式，因此在 ROCm 上改用 spawn。

实现拆解

1. 导入新模块：在 tests/v1/tracing/test_tracing.py 文件顶部添加 from vllm.distributed import cleanup_dist_env_and_memory 和 from vllm.platforms import current_platform。
2. 平台条件判断：在 test_traces 函数中，使用 current_platform.is_rocm() 判断当前是否为 ROCm 平台，若是则设置环境变量 VLLM_WORKER_MULTIPROC_METHOD=spawn，避免 gRPC 线程活跃时 fork。
3. 生命周期管理：将 LLM 的创建和使用放入 try 块，finally 块中检查引擎是否已创建，若是则调用 llm.llm_engine.engine_core.shutdown(timeout=...) 显式关闭；对于 ROCm，timeout 设为 60 秒以应对可能的延迟。
4. 清理分布式环境：在 finally 中调用 cleanup_dist_env_and_memory() 释放分布式相关资源，确保测试结束后环境干净。

关键文件：

- tests/v1/tracing/test_tracing.py（模块 追踪；类别 test；类型 test-coverage；符号 test_traces）：唯一被修改的文件，包含所有针对 ROCm 测试稳定性的修正。

关键符号：test_traces

关键源码片段

tests/v1/tracing/test_tracing.py

唯一被修改的文件，包含所有针对 ROCm 测试稳定性的修正。

```
# 设置 spawn 模式（ROCm 专用）
if current_platform.is_rocm():
    m.setenv('VLLM_WORKER_MULTIPROC_METHOD', 'spawn')

model = 'facebook/opt-125m'
llm = None
try:
    llm = LLM(
        model=model,
        gpu_memory_utilization=0.3,
        disable_log_stats=False,
    )
    # 生成并验证 span 属性（省略具体断言）
finally:
    if llm is not None:
        # ROCm 平台使用更长 timeout
        shutdown_timeout = 60.0 if current_platform.is_rocm() else 5.0
        llm.llm_engine.engine_core.shutdown(timeout=shutdown_timeout)
    cleanup_dist_env_and_memory()
```

评论区精华

- Reviewer: yewentao256针对 shutdown 的 timeout 长度提出疑问："Should this be longer?"，担心 ROCm 上的关闭操作可能需要更多时间。
- Author: AndreasKaratzas回应并增加 ROCm 专属的 shutdown timeout 至 60 秒（非 ROCm 保持 5 秒），解决了该疑虑。
- 最终 reviewer 批准 PR。
- ROCm shutdown timeout 是否需要更长 (question): 作者将 ROCm 的 shutdown timeout 调整为 60 秒，非 ROCm 保持 5 秒。

风险与影响

- 风险：
  - 低风险：变更仅涉及单测试文件，且所有平台特定逻辑通过 current_platform.is_rocm() 隔离，不影响其他平台。
  - 可能的故障：若 shutdown timeout 60 秒仍不足，测试可能 timeout，但远超实际需要，风险较低。
  - 无产品代码影响：改动全部在测试目录下，不涉及任何生产路径。
- 影响：
  - 对用户：无直接影响。
  - 对系统 /CI：消除 ROCm CI 上一个已知的 flaky segfault，提升测试可靠性。
  - 对团队：报告预期更少的误报失败，减少人工重试；代码模式可复用。
  - 风险标记：仅测试变更，平台特定修复，低风险

关联脉络

- 暂无明显关联 PR