PR #47208 完整报告

vllm-project/vllm

[CI][Bugfix] Rerun test_engine_log_metrics_ray on Ray GCS startup timeout

合并时间：2026-07-02 10:32

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47208

执行摘要

- 一句话：对 ROCM CI 上 Ray GCS 超时测试增加重试机制
- 推荐动作：推荐合入以消除 ROCm CI 上的不稳定测试。该 PR 模式（使用 pytest-rerunfailures 的 only_rerun 限定重试范围）值得作为测试不稳定问题的标准处理方式。

功能与动机

tests/v1/metrics/test_ray_metrics.py::test_engine_log_metrics_ray 在 ROCm CI 上间歇性失败，错误为 RuntimeError: Timed out waiting for file .../gcs_server_port_...。这是 Ray 集群启动竞态问题，而非 vLLM 缺陷。Ray 的 30 秒超时不可配置，唯一可行的方案是重试测试（重新 ray.init() 会获得新的 GCS 子进程和新的 30 秒窗口）。

实现拆解

1. 导入 current_platform：在文件顶部新增 from vllm.platforms import current_platform，用于条件判断。
2. 添加 @pytest.mark.flaky 装饰器：在 test_engine_log_metrics_ray 函数上方添加该装饰器，参数为 reruns=2（最多重试 2 次）、reruns_delay=5（重试间隔 5 秒）、only_rerun="Timed out waiting for file"（仅当异常信息包含该字符串时重试）、condition=current_platform.is_rocm()（仅在 ROCm 平台生效）。
3. 添加详细注释：在装饰器前添加多行注释，解释 GCS 超时原因及重试策略。

关键文件：

- tests/v1/metrics/test_ray_metrics.py（模块 测试；类别 test；类型 test-coverage）：唯一变更文件，新增 @pytest.mark.flaky 装饰器并导入 current_platform，用于对 Ray GCS 超时失败进行重试。

关键符号：未识别

关键源码片段

tests/v1/metrics/test_ray_metrics.py

唯一变更文件，新增 @pytest.mark.flaky 装饰器并导入 current_platform，用于对 Ray GCS 超时失败进行重试。

```
# SPDX-License-Identifier: Apache-2.0

from unittest.mock import MagicMock

import pytest
import ray

from vllm.config.model import ModelDType
from vllm.platforms import current_platform # 新增导入，用于条件判断
from vllm.sampling_params import SamplingParams
from vllm.v1.engine.async_llm import AsyncEngineArgs, AsyncLLM
from vllm.v1.metrics.ray_wrappers import (
    RayCounterWrapper,
    RayGaugeWrapper,
    RayHistogramWrapper,
    RayPrometheusMetric,
    RayPrometheusStatLogger,
)

MODELS = ["distilbert/distilgpt2"]


# The first .remote() call starts a local Ray cluster via ray.init(), whose
# GCS server occasionally fails to start within Ray's fixed 30s bootstrap
# window on ROCm CI (RuntimeError: "Timed out waiting for file
# .../gcs_server_port_..."). 该超时不可配置，因此重试整个测试：
# 重新 ray.init() 会获得新的 GCS 进程。限定重试范围到该错误，
# 以便真正的失败仍然立即失败。
@pytest.mark.flaky(
    reruns=2, # 最多重试 2 次
    reruns_delay=5, # 重试间隔 5 秒
    only_rerun="Timed out waiting for file", # 仅对 GCS 超时重试
    condition=current_platform.is_rocm(), # 仅在 ROCm 平台生效
)
@pytest.mark.parametrize("model", MODELS)
@pytest.mark.parametrize("dtype", ["half"])
@pytest.mark.parametrize("max_tokens", [16])
def test_engine_log_metrics_ray(
    example_prompts,
    model: str,
    dtype: ModelDType,
    max_tokens: int,
) -> None:
    """Simple smoke test, verifying this can be used without exceptions."""
    @ray.remote(num_gpus=1)
    class EngineTestActor:
        async def run(self):
            engine_args = AsyncEngineArgs(
                model=model, dtype=dtype, disable_log_stats=False, enforce_eager=True
```

```
)
engine = AsyncLLM.from_engine_args(
    engine_args, stat_loggers=[RayPrometheusStatLogger]
)
for i, prompt in enumerate(example_prompts):
    results = engine.generate(
        request_id=f"request-id-{i}",
        prompt=prompt,
        sampling_params=SamplingParams(max_tokens=max_tokens),
    )
    async for _ in results:
        pass
```

评论区精华

无 review 评论。维护者 AndreasKaratzas 直接批准了该 PR。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：仅修改测试文件，增加重试逻辑并限定了重试条件和平台。only_rerun 参数确保非 GCS 超时的真正失败不会被掩盖；condition 参数确保仅 ROCm 平台启用重试，不影响其他平台。
- 影响：影响范围限定于 ROCm CI 上 test_engine_log_metrics_ray 测试的稳定性，减少因 Ray 集群启动超时导致的误报。对其他平台和产品功能无影响。
- 风险标记：仅测试变更

关联脉络

- PR #47029 [Bugfix] Prevent padding placeholders from reaching embeddings: 同为修复 ROCm CI 上测试不稳定问题的 bugfix PR，使用了类似的 pytest.mark.flaky 模式。