

PR #47375 完整报告

vllm-project/vllm

[CI] Fix flaky lora test

合并时间: 2026-07-16 10:23

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47375>

执行摘要

- 一句话: 通过字符串格式化修复 LoRA 测试 flaky
- 推荐动作: 该 PR 是典型的 flaky 测试修复, 改动小而直接。对于关注 vLLM ROCm 测试稳定性的同学值得一读。建议在 Quark v0.12 发布后重新启用测试以验证修复。

功能与动机

在 ROCm 平台上, LoRA 测试 `test_gptoss_tp.py` 生成的文本中空格数量与期望字符串不一致, 导致测试随机失败。作者在 PR body 中明确说明: "On ROCm platform the generated string may have less spaces than the expected strings, so add reformatting."

实现拆解

1. 新增 `reformat` 函数 (位于 `tests/lora/test_gptoss_tp.py` 第 45-50 行): 该函数执行两步标准化操作——先移除逗号前后的空白字符, 再移除重复的连续空白。这样生成的文本与期望文本在格式上可准确比较。
2. 修改 `generate_and_test` 函数中的断言逻辑: 原代码简单使用 `" ".join(...)` 拆分并重新组合字符串, 然后 `assert generated.startswith(expected)`。新代码引入两个标准化字符串 `compactGeneratedStr` 和 `compactExpectedStr`, 并采用两种比较策略: 如果原始字符串不匹配, 再尝试标准化后的字符串进行比较; 若两者都不匹配, 则抛出 `AssertionError`。这样既兼容了原有行为, 又增加了对空格变化的容错。
3. 删除无效的 TODO 注释: 原 `generate_and_test` 函数上方的 TODO 注释 `# TODO: make the Mx4p4MoeBackend.TRITON spawn-safe...` 被删除, 可能是因为该问题已由其他方式解决或已过时。

关键文件:

- `tests/lora/test_gptoss_tp.py` (模块 测试; 类别 `test`; 类型 `test-coverage`; 符号 `reformat`)
: 唯一的变更文件, 包含核心修复: 新增 `reformat` 函数并增强断言逻辑。

关键符号: `reformat`

关键源码片段

`tests/lora/test_gptoss_tp.py`

唯一的变更文件, 包含核心修复: 新增 `reformat` 函数并增强断言逻辑。

```

def reformat(text: str) -> str:
    # 移除逗号前后的所有空格
    text = ",".join(map(str.strip, text.split(",")))
    # 移除重复的连续空白
    text = " ".join(map(str.strip, text.split()))
    return text

def generate_and_test(llm: vllm.LLM, lora_path: str, lora_id: int) -> None:
    prompts = [...]
    sampling_params = vllm.SamplingParams(temperature=0, max_tokens=64)
    outputs = llm.generate(prompts, sampling_params, lora_request=...)
    generated_texts: list[str] = []
    for output in outputs:
        prompt = output.prompt
        generated_text = output.outputs[0].text.strip()
        generated_texts.append(generated_text)
        print(f"Prompt: {prompt!r}, Generated text: {generated_text!r}")
    for i in range(len(EXPECTED_LORA_OUTPUT)):
        # 生成文本可能含不同数量的空白，因此统一格式化后再比较
        compactGeneratedStr = reformat(generated_texts[i])
        compactExpectedStr = reformat(EXPECTED_LORA_OUTPUT[i])
        if not generated_texts[i].startswith(
            EXPECTED_LORA_OUTPUT[i]
        ) and not compactGeneratedStr.startswith(compactExpectedStr):
            raise AssertionError(
                f"Generated: {generated_texts[i]}, Expected: {EXPECTED_LORA_OUTPUT[i]}"
            )

```

评论区精华

Reviewer [tjtanaa](#) 在评论中指出被删除的 TODO 注释可能已过时，询问是否保留。从最终提交来看，作者选择删除该注释。此外，Claude Code Review 自动提示来自 fork 的 PR 无法自动审查。

- 删除过时的 TODO 注释 (other): 作者选择直接删除该注释。

风险与影响

- 风险：
 - 回归风险：极低。修改仅涉及测试代码，且断言逻辑增强了对空格变化的容忍度，不会对生产代码产生任何影响。
 - 误判风险：reformat 函数虽然严格规范化了逗号前后的空格，但如果期望输出本身包含有意义的逗号前后空格（如自然语言），可能导致比较过于宽松。不过在本测试中，期望输出是 SQL 查询语句，空格无语义影响，因此风险较低。
 - 测试跳过：当前该测试组在 ROCm 上被跳过，因此修改的实际效果无法在 CI 中验证。需在 Quark v0.12 发布后重新启用测试方可确认修复生效。

- 影响：
 - 影响范围：仅影响 tests/lora/test_gptoss_tp.py 一个测试文件，且仅涉及一个测试函数 test_gpt_oss_lora。
 - 用户影响：无。测试修复不影响最终用户。
 - 团队影响：消除 ROCm 平台上该测试的随机失败，提高 CI 稳定性。但暂时仍被跳过，需等待 Quark 依赖更新。
 - 风险标记：测试跳过，低回归风险

关联脉络

- PR #48451 [feature]Add int4 quantization support for emulation moe backend: 均为涉及 ROCm 后端的量化 /MoE 相关 PR，且本 PR 测试依赖 Quark（量化库）版本。