

PR #53976 完整报告

vllm-project/vllm

[Bugfix][Test] Fix off-by-one error in sampled token rank causing flaky logprobs test

合并时间: 2026-09-01 02:49

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53976>

执行摘要

- 一句话: 修复采样 token 排名 off-by-one, 消除 logprobs 测试 flaky
- 推荐动作: 该 PR 值得简要阅读: 它展示了测试工具函数与生产代码 (logprobs.py) 对排名约定不一致导致的隐蔽 flaky 问题, 以及 reviewer 主动要求覆盖相邻同类函数的最佳实践。对于维护测试基础设施的工程师有参考价值。

功能与动机

测试 `tests/v1/engine/test_output_processor.py::test_logprobs_processor[None-5-Request OutputKind.DELTA]` 在 CI 中非确定性 (flaky) 失败, 根因是 `tests/v1/engine/utils.py` 中 `_create_random_top_token_test_vector()` 构造测试数据时, `choice_tensor[0]` 被强制设为 `sampled_token_id`, 而 `torch.randperm` 可能把 `sampled_token_id` 也随机放进 `choice_tensor[1:]` (概率约 $\text{num_logprobs} / \text{vocab_size}$, 如 $5/128000 \approx 0.004\%$)。此时工具函数返回 0-based 索引, 而 `vllm/logprobs.py::append_logprobs_for_next_position` 用 `range(1, num_logprobs + 1)` 赋 1-based 排名, 重复 token id 导致 dict 条目被覆盖, 最终测试断言 `ref_sampled_token_rank == smp_lp_rank` 失败。

实现拆解

1. 定位根因: 在 `tests/v1/engine/utils.py` 的 `_create_random_top_token_test_vector()` 中发现, 当 `sampled_token_id` 同时出现在 `choice_tensor[0]` (被强制设置) 和 `choice_tensor[1:]` (`randperm` 随机) 时, 工具函数对排名计算采用 0-based 索引, 而消费方 `vllm/logprobs.py` 的 `append_logprobs_for_next_position` 采用 `range(1, num_logprobs + 1)` 的 1-based 排名, 断言因此失败。
2. 修复 `_create_random_top_token_test_vector`: 在 `sample_token_rank` 计算中加 `+ 1`, 并添加注释说明需匹配消费方 1-based top-k 排名约定。
3. 修复 `_create_random_top_token_test_matrix`: 根据 reviewer njhill 的要求, 对相邻的 `_create_random_top_token_test_matrix` 中计算 `prompt_token_ranks` 的行也加 `+ 1`, 因为该函数同样为测试构造 top-k 排名数据, 存在相同的约定不一致问题。
4. 测试验证: PR body 提供独立可复现脚本, 确定性构造最坏情况 `choice_tensor` (`sampled_token_id` 出现在位置 0 和 3), 证明修复前后断言行为变化。

关键文件:

- tests/v1/engine/utils.py (模块 测试工具; 类别 test; 类型 test-coverage; 符号 `_create_random_top_token_test_vector`, `_create_random_top_token_test_matrix`) : 核心变更文件: 修复 `_create_random_top_token_test_vector` 和 `_create_random_top_token_test_matrix` 中排名 off-by-one, 使其与 `vllm/logprobs.py` 的 1-based top-k 排名约定一致。

关键符号: `_create_random_top_token_test_vector`,
`_create_random_top_token_test_matrix`

关键源码片段

tests/v1/engine/utils.py

核心变更文件: 修复 `_create_random_top_token_test_vector` 和 `_create_random_top_token_test_matrix` 中排名 off-by-one, 使其与 `vllm/logprobs.py` 的 1-based top-k 排名约定一致。

```
def _create_random_top_token_test_vector(
    num_logprobs: int,
    lower: int,
    upper: int,
    sampled_token_id: int,
    adjust_num_logprobs: bool,
) -> tuple[torch.Tensor, int]:
    """构造 top-k logprobs 测试向量。

    `choice_tensor[0]` 被强制设为 `sampled_token_id`, 而 `torch.randperm` 可能
    把 `sampled_token_id` 也随机放进 `choice_tensor[1:]`。此时需要返回采样 token
    在 top-k 中的排名, 且该排名必须与消费方 `logprobs.py` 的 1-based 约定一致。
    """

    # 计算最终所需的 logprobs 数量
    total_logprobs = num_logprobs + 1 if adjust_num_logprobs else num_logprobs

    # 用 torch.randperm 生成随机 token id (无放回抽样)
    choice_tensor = torch.randperm(upper - lower)[:total_logprobs] + lower

    # 确保采样 token 一定出现在 tensor 中 (位置 0 固定)
    choice_tensor[0] = sampled_token_id

    # 如果 sampled_token_id 同时出现在 choice_tensor[1:] (randperm 随机命中)
    if sampled_token_id in choice_tensor[1:]:
        # 修复前这里是 0-based 索引, 与消费方 range(1, num_logprobs + 1) 的
        # 1-based 排名不一致, 导致断言偶发失败; 这里 + 1 对齐约定
        sampled_token_rank = (
            (choice_tensor[1:] == sampled_token_id).nonzero(as_tuple=True)[0].item() + 1
        )
    else:
        # 未命中时, 随机给一个区间外的排名
```

```
sampled_token_rank = random.randint(num_logprobs, 50700)
```

```
return choice_tensor, sampled_token_rank
```

评论区精华

reviewer njhill 在评论中指出 `_create_random_top_token_test_vector` 相邻的 `_create_random_top_token_test_matrix` 方法也存在相同问题，要求一并修复；作者 mayuyuace 确认并补充了对应修复。随后 njhill 批准 (APPROVED)。

- 相邻 `_create_random_top_token_test_matrix` 也存在相同 off-by-one 问题 (question): 作者补上 `_create_random_top_token_test_matrix` 的 + 1 修复, reviewer 批准。

风险与影响

- 风险：变更仅涉及测试工具函数 `tests/v1/engine/utils.py`，不影响生产代码。风险点在于：`_create_random_top_token_test_matrix` 也会被其他测试调用（如构造 prompt logprobs 的测试），其返回的 rank 语义从 0-based 变为 1-based，可能影响依赖旧语义的测试断言，但由于修复的是与消费方约定一致的语义，理论上只会消除原本的不一致。若其他测试对该函数返回的 rank 有隐性 0-based 依赖（如直接用于索引），可能引入新的断言失败，需要 CI 验证。
- 影响：影响范围限于测试基础设施：修复 `test_output_processor.py` 及相关使用这些工具函数的 logprobs 测试的非确定性失败，提升 CI 稳定性。对用户无直接影响，对团队而言减少 flaky 测试重跑成本。
- 风险标记：测试工具语义变更，依赖旧 0-based 语义的测试可能受影响

关联脉络

- PR #53433 [Bugfix] Reject empty bad-word tokenizations: 同为修复 frontend/logprobs 相关测试的 bugfix PR
- PR #53790 [Bugfix] NemotronHMTTP: add hf_to_vllm_mapper so quant exclusions reach the MTP draft: 同属近期 bugfix 系列