PR #52399 完整报告

vllm-project/vllm

[Bugfix][Frontend] Return all choices from /inference/v1/generate when n > 1

合并时间：2026-08-19 20:48

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52399

执行摘要

- 一句话：修复 /inference/v1/generate 在 n>1 时静默丢弃 choices
- 推荐动作：值得精读。修复本身简洁，但根因链条清晰（output_kind 默认值与 serve_tokens_full_generator 聚合逻辑的交互），测试设计覆盖真实回归场景。CI 调试过程还展现了 VLLM_USE_RUST_FRONTEND 下前端实现分叉，可作为理解 vLLM scale-out 前端架构与测试跳过策略的窗口。

功能与动机

关联 Issue #52398 报告 /inference/v1/generate 在 n > 1 时会静默丢弃 choices，请求方拿到缺失的响应且无任何报错。PR body 定位根因：serve_tokens 只为流式请求设置 sampling_params.output_kind，非流式请求保留默认的 CUMULATIVE；此时 serve_tokens_full_generator 只读取最后一次引擎步进更新的序列。同时 CompletionRequest.to_sampling_params 在非流式时已使用 FINAL_ONLY，因此 token-in-token-out 路径应与之一致。

实现拆解

1. 根因定位：在 vllm/entrypoints/scale_out/token_in_token_out/serving.py 的 serve_tokens 中，output_kind 原先只在 request.stream 为真时设为 RequestOutputKind.DELTA，非流式请求保留 SamplingParams 默认值 CUMULATIVE。
2. 修复：改为无条件设置 output_kind——流式请求用 DELTA，非流式请求用 RequestOutputKind.FINAL_ONLY。这样 serve_tokens_full_generator 聚合最终响应时能拿到全部 n 个序列，不再依赖上一次引擎步进的状态。
3. 回归测试：在 tests/entrypoints/scale_out/token_in_token_out/test_serving_tokens.py 新增 test_generate_returns_all_choices_when_n_greater_than_one，以 n=4 的非流式请求断言响应 choices 索引覆盖 [0, 1, 2, 3]。
4. CI 适配：测试在 VLLM_USE_RUST_FRONTEND=1 时跳过——该环境下端点由 Rust 前端接管，Rust 前端的原始 generate 路由静默忽略 n 且总返回单个 choice，本修复不适用；按既有 test_generate_sampling_mask 先例处理。
5. 手工验证配套：PR 作者在 H100 上对 Qwen/Qwen2.5-1.5B 以 n=2/4/8 各发送 10 次非流式请求，修复前响应 choices 数量随机缺失，修复后全部返回完整 n 个；多模态 Qwen2.5-VL-3Bn=4 也通过。n=1、流式、logprobs、prompt_logprobs 行为不变。

关键文件：

- vllm/entrypoints/scale_out/token_in_token_out/serving.py（模块 前端服务；类别 source；类型 core-logic；符号 serve_tokens）：核心修复文件。serve_tokens 中 output_kind 的设置逻辑直接决定 /inference/v1/generate 非流式请求在 n > 1 时能否返回全部 choices。
- tests/entrypoints/scale_out/token_in_token_out/test_serving_tokens.py（模块 回归测试；类别 test；类型 test-coverage；符号 test_generate_returns_all_choices_when_n_greater_than_one）：新增回归测试 test_generate_returns_all_choices_when_n_greater_than_one，断言 n=4 时返回索引 0..3，并在 Rust 前端下跳过。

关键符号：serve_tokens, test_generate_returns_all_choices_when_n_greater_than_one

关键源码片段

vllm/entrypoints/scale_out/token_in_token_out/serving.py

核心修复文件。serve_tokens 中 output_kind 的设置逻辑直接决定 /inference/v1/generate 非流式请求在 n > 1 时能否返回全部 choices。

```
if self.force_no_detokenize:
    sampling_params.detokenize = False

# 关键修复：此前只有流式请求会设置 output_kind，非流式请求保留
# SamplingParams 默认的 CUMULATIVE，导致 n > 1 时只有最后一次引擎
# 步进更新的序列能进入响应，其余 choices 被静默丢弃。
# 现在非流式请求改走 FINAL_ONLY，与 CompletionRequest.to_sampling_params
# 在 OpenAI 兼容路径上的行为保持一致。
sampling_params.output_kind = (
    RequestOutputKind.DELTA if request.stream else RequestOutputKind.FINAL_ONLY
)

self._log_inputs(
    request_id,
    engine_input,
    params=sampling_params,
    lora_request=lora_request,
)
```

评论区精华

DarkLight1337 在 CI 失败后直接指出需要关注失败用例："PTAL at the failing test"，并附 Buildkite 链接。qgallouedec 最初无法在本地复现（"Strange, I can't reproduce this."），随后定位到环境差异：失败任务以 VLLM_USE_RUST_FRONTEND=1 运行，端点由 Rust 前端接管，而非 serve_tokens，因此本修复不适用。qgallouedec 进一步说明 Rust 前端的能力边界："the raw generate route silently ignores n (the field isn't even part of its SamplingParams) and always returns a single choice"，并据此按既有先例跳过测试，最终 CI 转绿。

- CI 失败定位到 Rust 前端环境 (testing): 按既有 test_generate_sampling_mask 先例，在 VLLM_USE_RUST_FRONTEND 下跳过新回归测试，CI 转绿。
- Rust 前端不支持并行采样 (design): 本 PR 不修改 Rust 前端，需后续 PR 让 Rust generate 路由支持 n > 1；当前通过 skip 规避测试失败。

风险与影响

- 风险：变更使 /inference/v1/generate 非流式请求统一走 FINAL_ONLY，与 OpenAI 兼容路径一致，但 FINAL_ONLY 要求引擎在最终输出时一次性返回完整序列；需确认 serve_tokens_full_generator 在中断、取消或提前停止场景下仍能给出正确结果，风险较低。Rust 前端（VLLM_USE_RUST_FRONTEND=1）下 n > 1 仍会静默丢弃 choices，本修复未覆盖该路径，用户启用 Rust 前端时仍会遇到原问题。新回归测试在 Rust 前端下被跳过，该路径缺少自动回归保护。
- 影响：对用户：使用 /inference/v1/generate 非流式并行采样（n > 1）的调用方现在能拿到全部 choices，响应不再静默缺失。对系统：核心逻辑改动仅一行表达式，影响面集中在 serve_tokens，已覆盖 n=1、流式、logprobs、prompt_logprobs 回归。对团队：PR 调试过程暴露 Python 前端与 Rust 前端在并行采样支持上的能力差距，后续需在 Rust 前端补齐 n > 1 支持。
- 风险标记：输出语义变更，Rust 前端未覆盖，回归测试跳过

关联脉络

- PR #52131 [Frontend] Move api_server.py out openai folder: 同属 entrypoints 前端入口层演进，本次修复的 scale_out token-in-token-out 端点与前端入口重构互为背景。