PR #52844 完整报告

vllm-project/vllm

[Bugfix][Rust Frontend] Reject n > 1 in the `/inference/v1/generate` route

合并时间：2026-08-20 15:41

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52844

执行摘要

本 PR 修复 Rust 前端（VLLM_USE_RUST_FRONTEND=1）下 /inference/v1/generate 静默忽略 n > 1 的缺陷：共享类型 vllm_text::SamplingParams 没有 n 字段，反序列化时该键被静默丢弃，请求 n=4 仍只返回单条结果。修复在路由层新增 GenerateSamplingParams 包装类型捕获 n，并在 validate_request_compat 中显式拒绝 n != 1，返回与 OpenAI 路由一致的 400 错误（param=n）。改动仅限 Rust 前端路由层，共享 crate 未动，单元测试与路由级回归测试齐全，PR 已由 BugenZhao 批准合并。

功能与动机

Issue #52843 报告：VLLM_USE_RUST_FRONTEND=1 时，向 /inference/v1/generate 发送 sampling_params: {"n": 4} 的非流式请求返回 HTTP 200 但只有单条 choices（index: 0），"No error/warning but the other completions are silently missing"。

根因有二：

- vllm_text::SamplingParams 刻意不含 n 字段（并行采样在 v1 中由 Python 前端 ParentRequest fan-out 实现），所以 n 键在 serde 反序列化时被丢弃，collect_generate 硬编码返回单条选择。
- Rust OpenAI 兼容路由（completions/validate.rs、chat_completions/validate.rs）早已用 bail_invalid_request!(param = "n", "Only n=1 is supported.") 显式拒绝，raw generate 路由却没有同等校验。

本 PR 使 raw generate 路由与 OpenAI 路由校验契约对齐，是 Python 前端修复 #52399 的 Rust 对应实现。

实现拆解

1. 数据契约（types.rs）：新增 GenerateSamplingParams { n: Option<u32>, #[serde(flatten)] inner: SamplingParams }，GenerateRequest.sampling_params 类型由 SamplingParams 改为 GenerateSamplingParams。serde(flatten) 保证 JSON 结构不变，只是把 n 单独捕获出来。
2. 校验（validate.rs）：validate_request_compat 中新增 request.sampling_params.n.unwrap_or(1) != 1 即返回 400 的逻辑；同时 max_tokens、prompt_logprobs 访问改为 sampling_params.inner.*。unwrap_or(1) 确保未传 n 的请求不受影响。

3. 消费端适配：convert.rs 的 prepare_generate_request 从 inner 读取 logprobs/prompt_logprobs 并透传 inner 给引擎；render.rs 的 lower_render_request 构造 GenerateSamplingParams { n: None, inner: text_request.sampling_params }；generate.rs 导出新类型。
4. 测试：validate.rs 内联两个单元测试（n=4 拒绝、n=1 接受）；tests.rs 新增 HTTP 路由级测试 raw_generate_rejects_parallel_sampling，断言 400 且 error.param == "n"。
5. 验证：cargo test -p vllm-server --lib、fmt、clippy 全绿，Buildkite CI #84762 已触发。无配置与部署配套改动。

rust/src/server/src/routes/inference/generate/validate.rs

校验主路径：新增对 sampling_params.n 的显式校验（拒绝 n != 1），并把 max_tokens、prompt_logprobs 访问迁移到 inner；同时内联两个单元测试覆盖 n=4 拒绝与 n=1 接受。

```
// SPDX-License-Identifier: Apache-2.0
// SPDX-FileCopyrightText: Copyright contributors to the vLLM project

use super::types::GenerateRequest;
use crate::error::{ApiError, bail_invalid_request};

/// 为 Rust token generate 路由强制执行最小兼容性契约。
pub(crate) fn validate_request_compat(
    request: &GenerateRequest,
    served_model_names: &[String],
) -> Result<(), ApiError> {
    // 模型名校验：不在已服务模型列表内则拒绝。
    if let Some(model) = request.model.as_ref()
        && !served_model_names.iter().any(|n| n == model)
    {
        return Err(ApiError::model_not_found(model.clone()));
    }

    // stream_options 仅在流式请求下有意义。
    if request.stream_options.is_some() && !request.stream {
        bail_invalid_request!(
            param = "stream_options",
            "stream_options are only supported when stream=true."
        );
    }

    // Rust 前端不实现并行采样（v1 中由 Python 前端 ParentRequest fan-out 承担），
    // `vllm_text::SamplingParams` 刻意省略 `n` 字段，因此这里显式捕获并拒绝 `n != 1`，
    // 避免反序列化时静默丢键、最终只返回单条结果。`unwrap_or(1)` 保证未传 `n` 时不误伤。
    if request.sampling_params.n.unwrap_or(1) != 1 {
        bail_invalid_request!(param = "n", "Only n=1 is supported.");
    }

    if request.token_ids.is_empty() {
```

```
        bail_invalid_request!(
            param = "token_ids",
            "token_ids must contain at least one token ID."
        );
    }

    // 其余采样参数统一通过包装类型的内层 SamplingParams 访问。
    if request.sampling_params.inner.max_tokens == Some(0) {
        bail_invalid_request!(
            param = "sampling_params",
            "max_tokens must be greater than 0."
        );
    }

    if let Some(prompt_logprobs) = request.sampling_params.inner.prompt_logprobs {
        // prompt_logprobs 需为非负值或 -1，且流式请求不支持。
        if prompt_logprobs < 0 && prompt_logprobs != -1 {
            bail_invalid_request!(
                param = "sampling_params",
                "`prompt_logprobs` must be a non-negative value or -1."
            );
        }

        if request.stream {
            bail_invalid_request!(
                param = "sampling_params",
                "`prompt_logprobs` are not available when `stream=true`."
            );
        }
    }

    Ok(())
}
```

rust/src/server/src/routes/inference/generate/types.rs

数据契约核心：新增 GenerateSamplingParams 包装类型，用 serde(flatten) 在路由层捕获共享 vllm_text::SamplingParams 刻意省略的 n 字段，是本修复的技术基础。

```
/// 供 generate API（token-in/token-out）使用的采样参数类型。
///
/// 包装 [`SamplingParams`] 以额外捕获 `n` 字段：共享的 `vllm_text::SamplingParams`
/// 故意不包含 `n`（并行采样由更高层处理，Rust 前端不实现）。若路由层不捕获 `n`，
/// 反序列化时该键会被静默丢弃，最终只返回单条生成结果。
#[serde_with::skip_serializing_none]
#[derive(Debug, Clone, Deserialize, Serialize)]
pub struct GenerateSamplingParams {
    /// 输出序列数量，本路由仅支持 1。
    pub n: Option<u32>,
    /// 其余采样参数经 flatten 后与原 JSON 结构保持一致，直接透传给引擎。
```

```
    #[serde(flatten)]
    pub inner: SamplingParams,
}

// GenerateRequest 中的使用方式：sampling_params 字段类型改为 GenerateSamplingParams，
// 反序列化时 `n` 被捕获到外层，其余键全部下沉到 inner，校验与引擎透传互不干扰。
pub struct GenerateRequest {
    // ...
    pub sampling_params: GenerateSamplingParams,
    // ...
}
```

评论区精华

- BugenZhao 在 validate.rs 上唯一的 review 评论是一条 suggestion，把条件从 n.unwrap_or(1) > 1 改为 n.unwrap_or(1) != 1：

> 建议 if request.sampling_params.n.unwrap_or(1) != 1 {

理由是 > 1 会放行 n == 0 这类同样非法的值，而 != 1 与 OpenAI 路由 "Only n=1 is supported" 的语义完全一致。该建议已在第二个 commit（BugenZhao 提交的 validate.rs 更新）中落实。

- Codex 自动 review："Didn't find any major issues."（审查 commit 9b0841d542）
- BugenZhao 最终 APPROVED："LGTM, thanks!"

风险与影响

风险

- API 行为变更：n > 1 从 " 静默返回 200 单条结果 " 变为 " 显式 400"，依赖旧行为的客户端会开始收到错误，但这是修复本意，且与 OpenAI 路由一致，属于合理收紧。
- 类型包装连带影响：GenerateRequest.sampling_params 类型变化波及所有内部访问点，若未来新增消费代码直接访问 sampling_params.max_tokens 等字段，会因字段移到 inner 而在编译期报错（Rust 类型系统兜底，运行期无隐患）；render 复用路径已在 lower_render_request 同步。
- 回归面：测试全部内联在 validate.rs 的 mod tests 与 routes/tests.rs，未新增独立测试文件；路由级测试覆盖了 HTTP 协议层，单元测试覆盖了边界值。

影响

- 用户：仅影响 VLLM_USE_RUST_FRONTEND=1 下调用 raw generate 传 n > 1 的客户端，从静默丢结果变为可诊断的 400 错误。
- 系统：改动限定在 Rust 前端路由层（generate.rs 模块与 render.rs），引擎侧与共享 vllm_text crate 行为不变，OpenAI 兼容路由无影响。
- 团队：为 Rust 前端补齐与 Python 前端一致的参数校验契约，并保留 n 字段的显式占位，为将来在 Rust 前端实现并行采样（ParentRequest fan-out）预留数据入口。

关联脉络

- Issue #52843：本 PR 直接关闭。
- PR #52399：Python 前端修复同一静默丢 n 问题（CUMULATIVE output kind 下丢 choices），其回归测试在 Rust 前端下被跳过，本 PR 以 Rust 侧测试补齐该缺口。
- 演进方向：vLLM 正把前端逻辑从 Python 迁移至 Rust（vllm-text 系列），本 PR 体现 " 路由层兼容性契约逐步对齐 " 的迁移策略——在不改共享 crate 的前提下，用 serde(flatten) 包装类型在路由边界捕获并校验路由特有约束。