PR #49496 完整报告

vllm-project/vllm

[Rust Frontend] Fix finish reason for named tool choices

合并时间：2026-07-28 18:53

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49496

修复 Rust 前端命名工具选择的 finish_reason

执行摘要

该 PR 修复了 Rust 前端 chat completions 响应中，当使用强制命名函数工具选择（tool_choice 指定具体函数）时，finish_reason 错误返回 "tool_calls" 的问题。通过新增 is_named_tool_choice 标志并调整 finish_reason 决定逻辑，named 模式现在正确返回 "stop"，与 Python 前端行为对齐。改动涉及 3 个文件，包含测试覆盖。

功能与动机

Rust 前端在 chat completions 路径中，只要响应包含工具调用，无论工具选择模式如何，都返回 finish_reason: "tool_calls"。根据 OpenAI 规范，当请求通过 tool_choice 强制指定了具体函数（named function choice）时，响应应当返回 "stop" 而非 "tool_calls"。Python 前端已正确处理该逻辑，Rust 前端的不一致会导致依赖 finish_reason 的客户端行为异常。PR body 指出：

> The Rust chat completions path returned finish_reason: "tool_calls" whenever a response contained a tool call. For a forced named function choice, the expected finish reason is "stop". The previous behavior caused the Rust and Python frontends to return different terminal reasons for the same request in both streaming and non-streaming modes.

实现拆解

1. 新增标志字段：在 convert.rs 的 ResponseOptions 结构体中添加 is_named_tool_choice: bool，并注释说明。
2. 检测命名工具选择：在 prepare_chat_request 函数中通过 matches!(&request.tool_choice, Some(ToolChoice::Function { .. })) 判断是否为强制命名函数调用，并将结果设置到 options.is_named_tool_choice。
3. 传递标志到响应构建：在 chat_completions.rs 的 collect_chat_completion 和 chat_completion_chunk_stream 函数中，从 ResponseOptions 解构出 is_named_tool_choice 并传递给 finish_reason 计算逻辑。
4. 调整 finish_reason 判断逻辑：修改 chat_finish_reason_to_openai 函数接口，将其参数从 saw_tool_calls: bool 改为 use_tool_calls_finish_reason: bool。在调用侧计算 saw_tool_calls && !is_named_tool_choice，使得 named 模式下即使有工具调用也不返回 "

tool_calls"。

5. 测试覆盖：在 tests.rs 中新增 weather_tool_call_output_specs 辅助函数抽取通用输出规格，新增 named_tool_choice_uses_stop_finish_reason 测试，同时覆盖 streaming 与 non-streaming 两种模式，断言 finish_reason 为 "stop"。同时重构现有测试 tool_calls_are_mapped_to_tool_call_sse_chunks 使用新的辅助函数。

rust/src/server/src/routes/openai/chat_completions.rs

核心逻辑改动：在 collect_chat_completion 和 chat_completion_chunk_stream 中引入 is_named_tool_choice 标志，调整 chat_finish_reason_to_openai 的参数语义，实现 named 模式下 finish_reason 为 stop。

```
// 在 collect_chat_completion 函数中，从 ResponseOptions 解构 is_named_tool_choice
async fn collect_chat_completion(
    stream: ChatEventStream,
    request_id: String,
    response_model: String,
    created: u64,
    ApiServerOptions { .. }: ApiServerOptions,
    ResponseOptions {
        // ... 其他字段
        is_named_tool_choice, // 新增：是否为强制命名函数选择
    }: ResponseOptions,
) -> Result<ChatCompletionResponse, ApiError> {
    // ... 收集中间输出 ...
    let saw_tool_calls = message.tool_calls().next().is_some();
    // 当 is_named_tool_choice 为 true 时，即使 saw_tool_calls 为 true，
    // 也不使用 "tool_calls" 作为 finish_reason，而是保留引擎返回的 stop
    let finish_reason =
        chat_finish_reason_to_openai(&finish_reason, saw_tool_calls && !is_named_tool_choice)?
            .to_string();
    // ... 构建响应 ...
}

// chat_finish_reason_to_openai 函数的参数语义从 saw_tool_calls 改为 use_tool_calls_finish_
reason
fn chat_finish_reason_to_openai(
    finish_reason: &FinishReason,
    use_tool_calls_finish_reason: bool, // 当该参数为 true 且引擎 FinishReason 为 Stop 时返回
    "tool_calls"
) -> Result<&'static str, ApiError> {
    match finish_reason {
        FinishReason::Stop(_) if use_tool_calls_finish_reason => Ok("tool_calls"),
        FinishReason::Stop(_) => Ok("stop"),
        FinishReason::Length => Ok("length"),
        FinishReason::Abort => Ok("abort"),
        // ...
    }
}
```

rust/src/server/src/routes/openai/chat_completions/convert.rs

新增 is_named_tool_choice 字段定义与初始化逻辑，是数据契约变更的入口。

```
#[derive(Debug, Clone, Default, PartialEq)]
pub(super) struct ResponseOptions {
    // ... 原有字段 ...
    /// Whether the request forces one named function tool.
    /// 当 tool_choice 为 {"type": "function", "function": {"name": ...}} 时为 true。
    pub is_named_tool_choice: bool,
}

// 在 prepare_chat_request 函数中初始化该字段
let is_named_tool_choice = matches!(&request.tool_choice, Some(ToolChoice::Function { .. }));
// 然后构造 PreparedRequest 时将 is_named_tool_choice 传入 options
let prepared = PreparedRequest {
    options: ResponseOptions {
        // ... 其他字段 ...
        is_named_tool_choice,
    },
    chat_request,
};
```

评论区精华

审查人 BugenZhao 直接批准（评论 "LGTM"），未提出修改意见。没有其他审查评论或讨论。

风险与影响

- 风险：新增字段 is_named_tool_choice 在其他构造路径中可能遗漏（但当前仅一处构造路径）；chat_finish_reason_to_openai 参数语义变更需确保所有调用点已更新（已全部更新）。整体风险较低。
- 影响：使用强制命名函数工具的客户端将收到正确的 finish_reason: "stop"，与 Python 前端一致。Rust 前端行为与 OpenAI 规范对齐，减少了客户端兼容性问题。无性能或兼容性影响。

关联脉络

该 PR 与近期历史 PR #49774、#49992 等同属 Rust 前端演进的一部分，逐步对齐 Python 前端的工具调用行为。此前 #49774 修复了推测解码的 draft 缓冲区问题，而本 PR 专注于工具调用的 finish_reason 一致性。目前尚无直接关联的 Issue 或后续计划，但为更完整的工具支持奠定了基础。