PR #35540 完整报告

vllm-project/vllm

[Bugfix] Fix empty channel/recipient in harmony for /v1/responses

合并时间：2026-05-12 16:45

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/35540

执行摘要

- 一句话：修复 /v1/responses 中 function_call_output 缺失 channel/recipient
- 推荐动作：建议尽快合并并发布，因为该修复直接提升 gpt-oss 等依赖 responses API 的工具调用准确率。开发者可关注后续 reasoning 分支健壮性改进以及测试文件合并建议。

功能与动机

PR body 指出：function_call_output 消息未正确设置 channel 和 recipient，导致 Harmony 格式转换后缺失 commentary channel 和 assistant recipient，影响了工具调用响应在聊天记录中的正确展示。该问题在 BFCL 测试中显著降低了 gpt-oss 的准确率。

实现拆解

1. 在 vllm/entrypoints/openai/responses/harmony.py 的 response_input_to_harmony 函数中，function_call_output 分支在构造消息后添加 msg.with_channel('commentary') 和 msg.with_recipient('assistant')，使输出与 chat completions 路径保持一致。
2. 新增 tests/entrypoints/openai/responses/test_response_input_to_harmony.py，覆盖 response_input_to_harmony 所有类型分支（message、reasoning、function_call_output、function_call 等），验证 role、channel、recipient、content 等字段正确。
3. 新增 tests/entrypoints/openai/parser/test_harmony_render_parity.py，针对每种消息场景（user、assistant、reasoning、function_call、function_call_output、组合），从 chat completions 和 responses 两条路径分别生成 Harmony 消息，断言它们内容一致，且 render_for_completion 输出相同 token 序列。

关键文件：

- vllm/entrypoints/openai/responses/harmony.py（模块 入口；类别 source；类型 core-logic；符号 response_input_to_harmony）：核心修复文件：在 response_input_to_harmony 函数的 function_call_output 分支中增加了 with_channel 和 with_recipient 调用，补全 Harmony 消息格式。
- tests/entrypoints/openai/parser/test_harmony_render_parity.py（模块 渲染对比；类别 test；类型 test-coverage；符号 _system, TestResponseInputToHarmonyRenderParity, test_user_message, test_assistant_final_message）：新增的跨 API 渲染对比测试，验证 chat completions 路径和 responses 路径对等效输入产生相同的 Harmony 消息和渲染

token 序列，确保 prompt 一致性。

- tests/entrypoints/openai/responses/test_response_input_to_harmony.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestResponseInputToHarmonyMessage, test_user_message_string_content, test_no_type_key_defaults_to_message_branch, test_system_message）：新增的 response_input_to_harmony 单元测试，覆盖每个 type 分支以及边界情况（缺失 type、数组 content 等），确保修复正确且不退化。

关键符号：response_input_to_harmony, TestResponseInputToHarmonyRenderParity, TestResponseInputToHarmonyMessage

关键源码片段

vllm/entrypoints/openai/responses/harmony.py

核心修复文件：在 response_input_to_harmony 函数的 function_call_output 分支中增加了 with_channel 和 with_recipient 调用，补全 Harmony 消息格式。

```
elif response_msg["type"] == "function_call_output":
    call_id = response_msg["call_id"]
    call_response: ResponseFunctionToolCall | None = None
    # 从历史响应中反向查找匹配的 function call，获取函数名
    for prev_response in reversed(prev_responses):
        if (
            isinstance(prev_response, ResponseFunctionToolCall)
            and prev_response.call_id == call_id
        ):
            call_response = prev_response
            break
    if call_response is None:
        raise ValueError(f"No call message found for {call_id}")
    # 构造 Tool 角色消息，作者为 functions.<name>
    msg = Message.from_author_and_content(
        Author.new(Role.TOOL, f"functions.{call_response.name}"),
        response_msg["output"],
    )
    # 修复：添加 channel 和 recipient，与 chat completions 路径对齐
    msg = msg.with_channel("commentary")
    msg = msg.with_recipient("assistant")
```

tests/entrypoints/openai/parser/test_harmony_render_parity.py

新增的跨 API 渲染对比测试，验证 chat completions 路径和 responses 路径对等效输入产生相同的 Harmony 消息和渲染 token 序列，确保 prompt 一致性。

```
def test_reasoning_item(self):
    # chat completions 路径：assistant 消息仅含 reasoning 字段，无 content
    chat_msgs = parse_chat_input_to_harmony_message(
        {
            "role": "assistant",
            "reasoning": "I should call get_weather.",
```

```
            "content": "",
        }
    )
    # responses 路径：type=reasoning 的输入项
    resp_msgs = [
        response_input_to_harmony(
            {
                "type": "reasoning",
                "content": [
                    {"type": "reasoning_text", "text": "I should call get_weather."}
                ],
            },
            prev_responses=[],
        )
    ]
    expected = [
        {
            "role": "assistant",
            "channel": "analysis",
            "content": "I should call get_weather.",
        }
    ]
    verify_harmony_messages(chat_msgs, expected)
    verify_harmony_messages(resp_msgs, expected)
    # 最终渲染出的 token 序列必须完全一致
    assert render_for_completion([_system()] + chat_msgs) == render_for_completion(
        [_system()] + resp_msgs
    )
```

tests/entrypoints/openai/responses/test_response_input_to_harmony.py

新增的response_input_to_harmony单元测试，覆盖每个type分支以及边界情况（缺失type、数组 content 等），确保修复正确且不退化。

```
def test_assistant_message_gets_final_channel(self):
    # type="message", role="assistant" 应自动获得 final channel
    msg = response_input_to_harmony(
        {"type": "message", "role": "assistant", "content": "The answer is 42."},
        prev_responses=[],
    )
    assert msg.author.role == Role.ASSISTANT
    assert msg.channel == "final"
    assert msg.content[0].text == "The answer is 42."

def test_reasoning_gets_analysis_channel(self):
    # type="reasoning" 应获得 analysis channel
    msg = response_input_to_harmony(
        {
            "type": "reasoning",
            "content": [{"type": "reasoning_text", "text": "Thinking hard."}],
```

```
        },
        prev_responses=[_REASONING_ITEM],
    )
    assert msg.channel == "analysis"
```

评论区精华

- gemini-code-assist[bot] 建议将 reasoning 分支的 assert len(content)==1 改为更健壮的拼接处理，但作者 kg6-sleipnir 认为超出当前 scope，未采纳。
- bbrowning 批准 PR，指出 chat completions 路径此前已修复，此 PR 将 responses 路径对齐；同时提议后续合并测试文件以消除冗余。
- chaunceyjiang 询问最新代码是否仍存在该问题，作者确认问题仍在，最后批准合并。
- reasoning 分支 assert 健壮性 (correctness): 未采纳，作者认为超出当前 scope，保留原写法。
- 测试文件合并建议 (testing): 暂不合并，优先修复问题，后续再优化测试组织。

风险与影响

- 风险：风险极低：核心改动仅两行，仅影响 function_call_output 消息的处理路径；两条新增测试套件覆盖了所有 type 分支和跨 API 渲染一致性，回包问题可以尽早发现。但 reasoning 分支中的 assert len(content)==1 在面对多 content 元素时仍可能崩溃，不过此问题已超出本次修复范围。
- 影响：影响范围集中在 /v1/responses 端点使用 Harmony 格式的场景（gpt-oss 等工具调用服务）。修复后 function_call_output 消息将正确携带 commentary channel 和 assistant recipient，使得客户端解析聊天历史时不再丢失消息类型，显著提升函数调用链的准确性。对不使用 responses API 或 Harmony 格式的用户无影响。
- 风险标记：低回归风险，测试覆盖增强

关联脉络

- 暂无明显关联 PR