

PR #48098 完整报告

vllm-project/vllm

[Bugfix] Fix parallel_tool_calls=null crash in Responses API from_request()

合并时间: 2026-07-15 18:01

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48098>

执行摘要

- 一句话: 修复 Responses API 中 parallel_tool_calls=null 崩溃
- 推荐动作: 值得合并。修复清晰且测试充分, 技术栈简单, 无明显风险。团队可将此 PR 作为 Pydantic 模型默认值处理的一个良好范例。

功能与动机

Fixes #48097. 根据 OpenAI API 文档, parallel_tool_calls 默认值为 true, 显式传入 null 应被视为默认值。原代码将 request.parallel_tool_calls (None) 直接传递给 ResponsesResponse 的强制 bool 字段, 引发 ValidationError。修复后 null 被正确解析为 True。

实现拆解

1. 在 vllm/entrypoints/openai/responses/protocol.py 的 ResponsesResponse.from_request() 方法中, 将 parallel_tool_calls 赋值改为条件表达式: 若 request.parallel_tool_calls 不为 None 则使用原值, 否则使用 ResponsesRequest 模型中该字段的默认值 (即 True)。此方式避免了硬编码默认值, 与模型定义保持同步。
2. 在 tests/tool_use/test_responses_request_validations.py 中导入 ResponsesResponse, 新增两个回归测试: test_responses_response_parallel_tool_calls_null_resolves_to_default 通过 from_request() 生产路径验证 null 被解析为 True、True 保留为 True、False 保留为 False; test_responses_request_parallel_tool_calls_null_accepted 验证请求模型接受 None 值。
3. 根据 Copilot 审查意见, 初始测试未覆盖 from_request() 路径, 随后在 commit 2c3acc5 中修正为使用真正的生产路径。

关键文件:

- vllm/entrypoints/openai/responses/protocol.py (模块 协议层; 类别 source; 类型 core-logic; 符号 from_request): 核心修复: 在 from_request() 中处理 parallel_tool_calls 为 None 的情况, 将其解析为字段默认值 True。
- tests/tool_use/test_responses_request_validations.py (模块 请求验证; 类别 test; 类型 test-coverage; 符号 test_responses_response_parallel_tool_calls_null_resolves_to_default, test_responses_request_parallel_tool_calls_null_accepted): 增加回归测试, 覆盖 parallel_tool_calls 为 None、True、False 三种情况, 并通过 from_request() 生产路径验

证。

关键符号: from_request, test_responses_response_parallel_tool_calls_null_resolves_to_default, test_responses_request_parallel_tool_calls_null_accepted

关键源码片段

vllm/entrypoints/openai/responses/protocol.py

核心修复: 在 from_request() 中处理 parallel_tool_calls 为 None 的情况, 将其解析为字段默认值 True。

```
# vllm/entrypoints/openai/responses/protocol.py
@classmethod
def from_request(
    cls,
    request: ResponsesRequest,
    sampling_params: SamplingParams,
    model_name: str,
    created_time: int,
    output: list[ResponseOutputItem],
    status: ResponseStatus,
    usage: ResponseUsage | None = None,
    input_messages: ResponseInputOutputMessage | None = None,
    output_messages: ResponseInputOutputMessage | None = None,
    kv_transfer_params: dict[str, Any] | None = None,
    ec_transfer_params: dict[str, Any] | None = None,
) -> "ResponsesResponse":
    incomplete_details: IncompleteDetails | None = None
    if status == "incomplete":
        incomplete_details = IncompleteDetails(reason="max_output_tokens")
    # ... 其他参数构造 ...
    return cls(
        # ... 其他字段 ...
        # 修复: 若 request.parallel_tool_calls 为 None 则使用模型字段的默认值 (True)
        parallel_tool_calls=request.parallel_tool_calls
        if request.parallel_tool_calls is not None
        else ResponsesRequest.model_fields["parallel_tool_calls"].default,
        temperature=sampling_params.temperature,
        tool_choice=request.tool_choice,
        tools=request.tools,
        # ...
    )
```

tests/tool_use/test_responses_request_validations.py

增加回归测试, 覆盖 parallel_tool_calls 为 None、True、False 三种情况, 并通过 from_request() 生产路径验证。

```
# tests/tool_use/test_responses_request_validations.py

# 回归测试: parallel_tool_calls=null 在 Responses API 中触发 crash 的修复
```

```

# (from_request() 将 None 传递给非可选 bool 字段导致 ValidationError)
@pytest.mark.parametrize(
    "value, expected",
    [
        (True, True),
        (False, False),
        (None, True), # null 必须解析为文档规定的默认值 true
    ],
)
def test_responses_response_parallel_tool_calls_null_resolves_to_default(
    value, expected
):
    # 构造一个请求, 设置 parallel_tool_calls 为 value
    request = ResponsesRequest.model_validate(
        {"input": "Hello", "model": "test-model", "parallel_tool_calls": value}
    )
    # 通过 from_request() 生产路径构造响应
    sampling_params = request.to_sampling_params(default_max_tokens=16)
    r = ResponsesResponse.from_request(
        request=request,
        sampling_params=sampling_params,
        model_name="test-model",
        created_time=0,
        output=[],
        status="completed",
        usage=None,
    )
    # 验证 parallel_tool_calls 是否与预期一致
    assert r.parallel_tool_calls == expected

def test_responses_request_parallel_tool_calls_null_accepted():
    """客户端发送 null 必须在请求验证阶段被接受。"""
    req = ResponsesRequest.model_validate(
        {"input": "Hello", "model": "test-model", "parallel_tool_calls": None}
    )
    # 请求级别应该保留 None, 后续由 from_request() 来解析
    assert req.parallel_tool_calls is None

```

评论区精华

Copilot 审查指出了三点:

1) 初始测试直接手动解析 None→True, 未验证 from_request() 生产路径; 作者已修正。 2) 建议使用 model_fields 的 default 而非硬编码 True, 避免第二真理源; 作者采纳。 3) 建议将注释中的 'unhandled Pydantic 500' 改为更准确的 'Pydantic ValidationError during response construction', 并集中导入 ResponsesResponse; 作者均已采纳。 审查者 chaunceyjiang 起初在本地无法复现, 作者提供了 curl 和 SDK 的复现步骤后确认并合并。

- 测试未覆盖真正的崩溃路径 (testing): 作者更新测试, 改为通过 `from_request()` 生产路径验证, 确保测试覆盖实际崩溃代码。
- 使用 `model_fields.default` 而非硬编码 `True` (design): 作者采纳, 在 commit 44b3926 中修改。
- 修改注释描述准确的失败模式 (documentation): 作者采纳, 在 commit d02368b 中修改。
- 集中导入 `ResponsesResponse` (style): 作者采纳, 在 commit d02368b 中修改。

风险与影响

- 风险: 变更范围极小, 仅在 `from_request()` 中增加一行条件表达式。使用 `model_fields.default` 确保与模型定义默认值保持一致, 降低未来默认值变更时的脱节风险。测试覆盖了 `None`、`True`、`False` 三种情况, 并通过 `from_request()` 生产路径验证, 回归风险低。无性能影响。
- 影响: 用户: 发送 `parallel_tool_calls=null` 的请求不再返回 400 错误, 而是正常返回结果, 符合 API 文档默认行为。系统: 仅影响 `Responses` API 端点, 不影响其他 API。团队: 测试覆盖完善, 避免类似回归。
- 风险标记: 低风险, 测试覆盖完善

关联脉络

- PR #44955 Fix `parallel_tool_calls: null` treated as false instead of default true: 之前修复了 `ChatCompletion` 路径中 `parallel_tool_calls=null` 被当作 false 的问题, 但未覆盖 `Responses` API。本 PR 补齐了这一差距。