PR #43862 完整报告

vllm-project/vllm

[Bugfix] fix crash in postprocess for null tool args

合并时间：2026-06-03 13:17

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/43862

执行摘要

- 一句话：修复 tool_call arguments 为 "null" 字符串时的崩溃
- 推荐动作：此 PR 值得合并，问题定位清晰，修复方式简单安全，测试充分。对于关注工具调用稳定性的团队，可直接参考此修复。

功能与动机

修复 issue #43851：GLM-5.1 模型在 tool_call arguments 为字符串 "null" 时，因 None.items() 调用崩溃。用户多轮对话中模型调用零参数工具后，客户端原样回传 assistant 消息，arguments 变为字符串 "null" 触发该 bug。

实现拆解

1. 修改 _postprocess_messages 函数（vllm/entrypoints/chat_utils.py）：将原一行 function["arguments"] = json.loads(content) 改为两行，先用变量接收解析结果，再判断是否为 None，若为 None 则赋值为空字典 {}。
2. 添加回归测试（tests/entrypoints/test_chat_utils.py）：新增 test_postprocess_messages_null_arguments_string 函数，构造包含 arguments="null" 的 assistant 消息，调用 _postprocess_messages 后验证 arguments 被正确转换为 {}。

关键文件：

- vllm/entrypoints/chat_utils.py（模块 入口解析；类别 source；类型 core-logic；符号 _postprocess_messages）：核心修复文件：修改 _postprocess_messages 函数，处理 json.loads 返回 None 时的边界情况
- tests/entrypoints/test_chat_utils.py（模块 测试；类别 test；类型 test-coverage；符号 test_postprocess_messages_null_arguments_string）：回归测试：验证 arguments="null" 被正确转换为 {}，避免回归

关键符号：_postprocess_messages, test_postprocess_messages_null_arguments_string

关键源码片段

vllm/entrypoints/chat_utils.py

核心修复文件：修改 _postprocess_messages 函数，处理 json.loads 返回 None 时的边界情况

```
# vllm/entrypoints/chat_utils.py
# _postprocess_messages 函数内，处理 tool_call arguments 的关键分支

            # if arguments is None or empty string, set to {}
            if content := function.get("arguments"):
                if not isinstance(content, (dict, list)):
                    # 解析 JSON 字符串，例如 "null" 会被解析为 Python None
                    parsed = json.loads(content)
                    # 修复：若解析结果为 None，也转为 {} 以避免 chat template 中
                    # 调用 None.items() 崩溃（如 GLM 的 Jinja2 模板）
                    function["arguments"] = parsed if parsed is not None else {}
                else:
                    function["arguments"] = {}
```

tests/entrypoints/test_chat_utils.py

回归测试：验证 arguments="null" 被正确转换为 {}，避免回归

```
# tests/entrypoints/test_chat_utils.py

def test_postprocess_messages_null_arguments_string():
    """arguments="null" must not reach the chat template as Python None.

    json.loads("null") returns None, which causes Jinja2 templates that call
    tc.arguments.items() to raise 'None' has no attribute 'items'.
    The function should coerce it to {} instead.
    """
    # 构造典型的 assistant 消息，其中 tool_call 的 arguments 为字符串 "null"
    messages: list[ConversationMessage] = [
        {
            "role": "assistant",
            "content": None,
            "tool_calls": [
                {
                    "id": "call_1",
                    "type": "function",
                    "function": {"name": "get_current_time", "arguments": "null"},
                }
            ],
        }
    ]
    _postprocess_messages(messages)
    tool_calls = messages[0]["tool_calls"]
    assert tool_calls is not None
    # 验证 arguments 被正确转换为空字典 {}
    assert tool_calls[0]["function"]["arguments"] == {}
```

评论区精华

Reviewer AndreasKaratzas 确认变更逻辑简单（仅多加一次 if 判断），并等待另一位 reviewer 确认后强制合入。未发现关于设计或正确性的争议。

- 变更正确性和 CI 状态确认 (design): 无需额外修改，可直接合并。

风险与影响

- 风险：风险极低：改动仅影响 json.loads 返回 None 的边界情况，且语义与已有注释一致（' if arguments is None or empty string, set to {}'）。测试覆盖了该路径，且不影响其他正常 JSON 解析。
- 影响：直接影响使用 tool calling 功能的用户，特别是模型（如 GLM）可能返回 arguments="null" 的场景。间接影响 chat template 渲染，避免 Jinja2 模板因 None.items() 崩溃。影响范围窄，仅涉及辅助函数内的一条分支。
- 风险标记：边界情况修复，测试覆盖充分

关联脉络

- 暂无明显关联 PR