

PR #48852 完整报告

vllm-project/vllm

[Bugfix][Tool Parser] Fix dropped streaming arguments in Jamba and InternLM2 parsers

合并时间: 2026-07-25 21:34

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48852>

执行摘要

- 一句话: 修复 Jamba 和 InternLM2 流式工具调用参数丢失 bug
- 推荐动作: 建议快速合并。修复针对性强, 测试恰当, 已被 reviewer 批准。此 PR 也是更早修复 #48755 的改进版本。

功能与动机

在流式工具调用中, 第一次工具参数变为非空时, Jamba 和 InternLM2 解析器通过 `str.index` 计算流式 delta。原始 delta 文本可能不在 `json.dumps` 的重新序列化结果中严格出现 (如包含结构文本或格式差异)。当 delta 不是子串时, `str.index` 引发 `ValueError`, 被外围 `except` 吞掉, 导致 delta 变为 `None` 且 `prev_tool_call_arr` 不推进, 因此参数完成也不触发分支, 客户端最终收到空参数的调用。这种情况在粗粒度 delta (如 `stream_interval > 1`) 时真实发生。

实现拆解

1. 引入 `is_complete_json` 工具函数: 在两个解析器的导入语句中添加 `is_complete_json`, 用于判断当前累计 JSON 是否完整。
2. 替换 `str.index` 为 `str.find` 并增加分支: 在 `extract_tool_calls_streaming` 方法中, 当首次参数出现时, 将原来的 `cur_arguments_json.index(delta_text)` 改为 `cur_arguments_json.find(delta_text)`。如果找到, 按原逻辑截取; 如果找不到但 `is_complete_json(parsable_arr)` 为真, 则发送整个参数; 否则返回 `None` 等待更多文本。
3. 保持增量截断不变: 当参数已部分存在时, 继续使用已有的 `extract_intermediate_diff` 计算增量, 不受影响。
4. 新增回归测试: 为 Jamba 和 InternLM2 各自添加一个独立测试, 模拟粗粒度 delta 场景 (工具名和参数分步到达), 验证最终累积的参数可正确解析为预期 JSON。
5. 验证现有测试: 运行两个解析器的全部现有测试, 确认未回退。

关键文件:

- `vllm/tool_parsers/jamba_tool_parser.py` (模块 工具解析器; 类别 source; 类型 core-logic; 符号 `extract_tool_calls_streaming`): 核心源码, 修复 Jamba 解析器流式工具调用参数定位逻辑, 使用 `str.find` 替代 `str.index` 并增加完整性与等待判断。
- `vllm/tool_parsers/internlm2_tool_parser.py` (模块 工具解析器; 类别 source; 类型 core-logic; 符号 `extract_tool_calls_streaming`): 核心源码, 修复 InternLM2 解析器相同问题, 与 Jamba 修复一致。

- tests/tool_parsers/test_internlm2_tool_parser.py (模块 工具解析器测试; 类别 test; 类型 test-coverage; 符号 test_streaming_arguments_in_single_delta) : 新增回归测试 test_streaming_arguments_in_single_delta, 验证粗粒度 delta 下参数不丢失。
- tests/tool_parsers/test_jamba_tool_parser.py (模块 工具解析器测试; 类别 test; 类型 test-coverage; 符号 test_extract_tool_calls_streaming_arguments_in_single_delta) : 新增回归测试 test_extract_tool_calls_streaming_arguments_in_single_delta, 验证 Jamba 解析器粗粒度 delta 下参数不丢失。

关键符号: JambaToolParser.extract_tool_calls_streaming,
Internlm2ToolParser.extract_tool_calls_streaming,
test_streaming_arguments_in_single_delta, test_extract_tool_calls_streaming_arguments_in_single_delta

关键源码片段

vllm/tool_parsers/jamba_tool_parser.py

核心源码, 修复 Jamba 解析器流式工具调用参数定位逻辑, 使用 str.find 替代 str.index 并增加完整性与等待判断。

```
# JambaToolParser.extract_tool_calls_streaming 中首次参数出现分支 (修复后)
# ...
elif cur_arguments and not prev_arguments:
    cur_arguments_json = json.dumps(cur_arguments, ensure_ascii=False)
    logger.debug('finding %s in %s', new_text, cur_arguments_json)

    # `new_text` 可能不在重新序列化的 JSON 中精确出现
    match_start = cur_arguments_json.find(new_text)
    if match_start != -1:
        # 正常截取: 从开头到匹配位置加上 delta 长度
        arguments_delta = cur_arguments_json[
            : match_start + len(new_text)
        ]
    elif is_complete_json(parsable_arr):
        # 本 delta 中已完整: 发送整个参数, 避免丢失
        arguments_delta = cur_arguments_json
    else:
        # 还不完整: 等待更多文本
        return None
    logger.debug('First tokens in arguments received: %s', arguments_delta)
    delta = DeltaMessage(
        tool_calls=[
            DeltaToolCall(
                index=self.current_tool_id,
                function=DeltaFunctionCall(
                    arguments=arguments_delta
                ).model_dump(exclude_none=True),
            )
        ]
    )
```

```
)  
self.streamed_args_for_tool[self.current_tool_id] += arguments_delta
```

vllm/tool_parsers/internlm2_tool_parser.py

核心源码，修复 InternLM2 解析器相同问题，与 Jamba 修复一致。

```
# Internlm2ToolParser.extract_tool_calls_streaming 中首次参数出现分支（修复后）  
# ...  
elif cur_arguments and not prev_arguments:  
    cur_arguments_json = json.dumps(cur_arguments, ensure_ascii=False)  
  
    match_start = cur_arguments_json.find(delta_text)  
    if match_start != -1:  
        # 正常截取：从开头到匹配位置加上 delta 长度  
        arguments_delta = cur_arguments_json[  
            : match_start + len(delta_text)  
        ]  
    elif is_complete_json(parsable_arr):  
        # 本 delta 中已完整：发送整个参数，避免丢失  
        arguments_delta = cur_arguments_json  
    else:  
        # 还不完整：等待更多文本  
        return None  
    delta = DeltaMessage(  
        tool_calls=[  
            DeltaToolCall(  
                index=self.current_tool_id,  
                function=DeltaFunctionCall(  
                    arguments=arguments_delta  
                ).model_dump(exclude_none=True),  
            )  
        ]  
    )  
    self.streamed_args_for_tool[self.current_tool_id] += arguments_delta
```

评论区精华

review 中无重大争议。bbrowning 批准并指出变更逻辑自包含且测试通过。作者在 PR 评论中说明了 DCO 问题和 CI 失败均与本次变更无关，bbrowning 确认最终 CI 全部通过。

- DCO 签名一致性要求 (other): 作者合并了补丁提交，统一为同一签名格式，DCO 随后变绿。
- CI 失败与本次变更无关 (other): 经重新运行确认通过。

风险与影响

- 风险：风险较低。变更集中在两个解析器的首次参数出现分支中，但引入了新的条件判断。如果 is_complete_json 错误地返回 true 可能导致发送不完整参数，但 partial_json_parser 的完整性校验通常准确。如果不完整且找不到 delta，返回 None 导致客户端暂未收到更新，但这是正确的等待行为，避免发送可能破坏累积状态的字符串。测试

验证了主要场景。

- 影响：用户：修复了使用 stream 调用工具时参数可能为空的问题，提升可靠性和用户体验。
系统：仅影响工具调用解析路径，性能无显著变化。团队：代码改动量小，测试覆盖了新增的回归场景。
- 风险标记：低风险，回归测试覆盖

关联脉络

- PR #48755 [Bugfix][Tool Parser] Fix streaming tool call arguments for InternLM2: 本 PR 取代了 #48755，修复了相同的 InternLM2 流式参数丢失问题但采用了更稳健的分支策略（保留增量截断，使用 is_complete_json 判断完整性），原作者同意关闭 #48755。