

PR #47311 完整报告

vllm-project/vllm

[Bugfix][Tool Parser] poolside_v1: accept tool calls without newline after function name

合并时间: 2026-07-03 03:08

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/47311>

执行摘要

- 一句话: 修复 poolside_v1 工具解析器忽略无换行 tool call 的 bug
- 推荐动作: 该 PR 值得审查, 因为一行修复解决了一个静默丢弃 tool call 的问题, 测试覆盖良好, 且设计上考虑了向后兼容。对于维护工具解析的工程师, 可关注其正则设计模式。

功能与动机

poolside_v1 的非流式 `extract_tool_calls` 使用 `func_detail_regex` 要求函数名后必须有一个换行符才能匹配, 但某些模型 (如 Laguna) 输出格式为 `<tool_call>get_weather<arg_key>city</arg_key>...</tool_call>`, 缺少换行符, 导致 tool call 被忽略且原始标记泄露到生成文本中。本次变更旨在兼容两种格式。

实现拆解

实现变更包括两步:

1. 修改非流式正则: 在 `vllm/tool_parsers/poolside_v1_tool_parser.py` 中将 `func_detail_regex` 由强制要求函数名后换行的形式改为使换行可选, 兼容无换行格式。
2. 补充回归测试: 在 `tests/tool_parsers/test_poolside_v1_tool_parser.py` 中添加三个新测试用例:
 - `test_no_newline_after_name_non_streaming`: 验证非流式下无换行格式正确解析。
 - `test_newline_after_name_still_parses_non_streaming`: 验证带换行格式仍能解析 (回归保障)。
 - `test_no_newline_after_name_streaming`: 验证流式路径下无换行格式也能逐步累积正确结果。流式路径因自带 fallback 逻辑, 无需额外修改。测试辅助函数 `_weather_tool` 和 `_build_weather_request` 被抽取用于构建标准请求。注意: 所有修改仅针对非流式路径的正则, 流式路径未改动。

关键文件:

- `vllm/tool_parsers/poolside_v1_tool_parser.py` (模块 工具解析; 类别 source; 类型 core-logic; 符号 `PoolsideV1ToolParser`, `func_detail_regex`): 核心修复: 修改非流式正则使换行可选
- `tests/tool_parsers/test_poolside_v1_tool_parser.py` (模块 测试; 类别 test; 类型 test-coverage; 符号 `_weather_tool`, `_build_weather_request`,

test_no_newline_after_name_non_streaming, test_newline_after_name_still_parses_no_n_streaming)：新增三个回归测试，覆盖无换行和带换行、流式和非流式场景

关键符号：test_no_newline_after_name_non_streaming,
test_newline_after_name_still_parses_non_streaming,
test_no_newline_after_name_streaming

关键源码片段

vllm/tool_parsers/poolside_v1_tool_parser.py

核心修复：修改非流式正则使换行可选

代码文件：vllm/tool_parsers/poolside_v1_tool_parser.py

```
# 修改前的正则要求函数名后必须有换行符：
# r'<tool_call>([\n]*)\n(.*)</tool_call>'
# 导致无换行的 tool call（如 Laguna 输出）无法匹配提取。
# 修改后正则使换行可选，并放宽了空白要求：
self.func_detail_regex = re.compile(
    r'<tool_call>\s*([\n]+?)\s*\n?\s*(<arg_key>.*?)?</tool_call>', re.DOTALL
    # \s* 开头空白
    # ([\n]+?) 非贪婪捕获函数名（不含 < 或 \n）
    # \s*\n?\s* 换行可选，允许周围空白
    # (<arg_key>.*?)? 参数块保持可选
)
```

tests/tool_parsers/test_poolside_v1_tool_parser.py

新增三个回归测试，覆盖无换行和带换行、流式和非流式场景

代码文件：tests/tool_parsers/test_poolside_v1_tool_parser.py

```
def _weather_tool() -> dict[str, Any]:
    # 定义一个简单的天气查询工具 schema
    return {
        'type': 'function',
        'function': {
            'name': 'get_weather',
            'description': 'Get the weather for a city',
            'parameters': {
                'type': 'object',
                'properties': {'city': {'type': 'string'}},
                'required': ['city'],
            },
        },
    }

def _build_weather_request(*, tool_choice: str) -> ChatCompletionRequest:
    # 构建一个带工具定义的 ChatCompletionRequest
    return ChatCompletionRequest.model_validate({
        'model': 'poolside-test',
```

```

        'messages': [{'role': 'user', 'content': 'weather in Paris?'}],
        'tools': [_weather_tool()],
        'tool_choice': tool_choice,
    })

def test_no_newline_after_name_non_streaming() -> None:
    # 回归测试：无换行格式在非流式路径下应正确解析
    request = _build_weather_request(tool_choice='auto')
    parser = _make_parser(request)

    model_output = (
        '<tool_call>get_weather'
        '<arg_key>city</arg_key><arg_value>Paris</arg_value>'
        '</tool_call>'
    )

    result = parser.extract_tool_calls(model_output, request)

    # 验证工具调用被正确提取
    assert result.tools_called
    assert len(result.tool_calls) == 1
    assert result.tool_calls[0].function.name == 'get_weather'
    args = json.loads(result.tool_calls[0].function.arguments)
    assert args == {'city': 'Paris'}

```

评论区精华

无实质 review 讨论；claude bot 自动评论无法审查 fork 提交，但项目维护者 @sfeng33 和 @mgoin 均批准（LGTM）。作者在 Issue 评论中指出 CI 失败与本次变更无关。

- CI 失败与 PR 无关 (other): 未深入讨论，但其他维护者已批准。

风险与影响

- 风险：风险极低。仅改动非流式路径中的一行正则，且新正则向后兼容原始 GLM-4 格式。测试覆盖了非流式和流式两种场景，以及带换行和无换行的变体。流式路径未修改，原有逻辑得到保持。可能存在极端输入如包含特殊字符的调用，但正则设计时已考虑非贪婪匹配和空白优化。
- 影响：影响范围仅限于 poolside_v1 工具解析器。对于使用该解析器的模型（如 GLM-4），现可处理不带换行的 tool call 输出，减少工具调用遗漏。对系统其他部分无影响。用户升级后无需配置变更。
- 风险标记：低风险，向后兼容，单行修改

关联脉络

- 暂无明显关联 PR