

PR #48846 完整报告

vllm-project/vllm

[Bugfix][Tool Parser] Preserve whitespace in parameter values (MiniMax M2, Qwen3, MiniCPM5 XML)

合并时间: 2026-07-18 00:45

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/48846>

执行摘要

- 一句话: 修复三个 tool parser 中参数值前后空白被错误剔除的问题
- 推荐动作: 建议精读本 PR, 了解对不同格式的空白处理策略 (内联直接保留、模板换外层裁剪、CDATA 条件 strip) 以及流式安全分析。对于维护其他 tool parser 的工程师, 应遵循空白保留的原则。

功能与动机

编码代理依赖精确字符串匹配来执行编辑操作, 参数值中合法的前导空格和尾部换行被 `str.strip()` 删除后, 会导致客户端匹配失败或缩进错误。多轮对话中每次重新解析都会加剧损失。Issue #48753 报告了 Qwen3 的同类问题, 经检查 MiniMax M2 和 MiniCPM5 XML 亦受影响。

实现拆解

1. MiniMax M2 ([vllm/parser/minimax_m2.py](#)): 移除 `_minimax_m2_arg_converter` 完整匹配循环和流式 `partial` 分支中对参数值调用的 `.strip()`, 保持值原样。理由: MiniMax 格式中值位于 `>` 和 `</parameter>` 之间, 空白是数据而非标记。该行为与同族的 glm47 转换器一致。
2. Qwen3 ([vllm/parser/qwen3.py](#)): 新增 `_trim_wrapping_newlines` 辅助函数, 仅移除首尾各一个换行 (因为 Jinja 模板在 `<parameter=NAME>` 和 `</parameter>` 前后各渲染了一个换行)。`_qwen3_arg_converter` 的两个路径均将 `.strip()` 替换为 `_trim_wrapping_newlines`。该设计确保了缩进和尾部多换行得到保留。
3. MiniCPM5 XML ([vllm/tool_parsers/minicpm5xml_tool_parser.py](#)): 在 `_parse_function_block` 和 `_parse_partial_params` 两处正则路径中, 将 `.strip()` 移入 `else` 分支, 仅在非 CDATA 值时 `strip`; `etree` 路径 (`param.text`) 直接去掉 `strip`, 因为 `lxml` 会将 CDATA 解析为普通字符串, 无法区分。这安全是因为 MiniCPM 输出值紧贴标签, 无额外空白。
4. 测试配套: 为每个 parser 新增 `TestParameterWhitespace` 类, 覆盖完整解析和流式分块场景, 确保空白保留。修正 `tests/parser/engine/test_qwen3.py` 中过时的预期值 (流式末尾空格不再被 `strip`)。

关键文件:

- `vllm/parser/qwen3.py` (模块 Qwen3 解析器; 类别 source; 类型 core-logic; 符号 `_trim_wrapping_newlines`, `_qwen3_arg_converter`) : 核心修复文件: 引入 `_trim_wrapping_newlines` 辅助函数, 替换了 `_qwen3_arg_converter` 中的 `str.strip()`, 实现了仅移除模板包裹换行的精确处理。
- `vllm/parser/minimax_m2.py` (模块 MiniMax 解析器; 类别 source; 类型 core-logic; 符号 `_minimax_m2_arg_converter`) : 核心修复文件: 在 `_minimax_m2_arg_converter` 的完整匹配和流式分支中移除 `param.value` 上的 `.strip()`, 保持值与 glm47 一致。
- `vllm/tool_parsers/minicpm5xml_tool_parser.py` (模块 MiniCPM5 解析器; 类别 source; 类型 core-logic; 符号 `_parse_function_block`, `_parse_partial_params`) : 核心修复文件: 在正则路径中将 `strip` 移入非 CDATA 分支, `etree` 路径完全去掉 `strip`, 确保 CDATA 值中的空白保留。
- `tests/tool_parsers/test_minimax_m2_tool_parser.py` (模块 MiniMax 测试; 类别 test; 类型 test-coverage; 符号 `TestParameterWhitespace`, `test_whitespace_preserved`, `test_whitespace_preserved_across_chunks`) : 新增 `TestParameterWhitespace` 类, 覆盖完整解析和流式分块场景下空白保留的端到端测试。
- `tests/tool_parsers/test_qwen3coder_tool_parser.py` (模块 Qwen3 测试; 类别 test; 类型 test-coverage; 符号 `TestParameterWhitespace`, `test_whitespace_preserved`, `test_whitespace_preserved_streaming`) : 新增 `TestParameterWhitespace` 类, 覆盖完整解析和流式分块场景下 Qwen3 空白保留的测试。
- `tests/tool_parsers/test_minicpm5xml_tool_parser.py` (模块 MiniCPM5 测试; 类别 test; 类型 test-coverage; 符号 `make_tools_write`, `TestParameterWhitespace`, `test_cdata_whitespace_preserved`, `test_cdata_whitespace_preserved_streaming`) : 新增 `TestParameterWhitespace` 类和 `make_tools_write` 辅助函数, 覆盖 CDATA 值中空白保留的完整和流式测试。
- `tests/parser/engine/test_qwen3.py` (模块 Qwen3 引擎测试; 类别 test; 类型 test-coverage) : 修正流式测试预期, 移除旧 `strip` 行为导致的错误预期值, 使测试通过。

关键符号: `_trim_wrapping_newlines`, `_minimax_m2_arg_converter`, `_qwen3_arg_converter`, `_parse_function_block`, `_parse_partial_params`

关键源码片段

`vllm/parser/minimax_m2.py`

核心修复文件: 在 `_minimax_m2_arg_converter` 的完整匹配和流式分支中移除 `param.value` 上的 `.strip()`, 保持值与 glm47 一致。

```
def _minimax_m2_arg_converter(raw_args: str, partial: bool) -> str:
    params: dict[str, object] = {}
    for match in _PARAM_RE.finditer(raw_args):
        name = match.group("name").strip() # name 仍 strip
        if not name:
            continue
        # 保持值原样 (与 glm47 转换器一致): 值位于 `>` 和 `</parameter>` 之间, 空白是数据
        params[name] = match.group("value")
    if partial:
```

```
remaining = _PARAM_RE.sub("", raw_args)
m = _PARTIAL_PARAM_RE.search(remaining)
if m:
    name = m.group("name").strip()
    value = m.group("value")
    if name:
        # 保持原始值, 不 strip
        params[name] = value
return json.dumps(params, ensure_ascii=False)
```

评论区精华

Review 讨论:

- yzong-rh 在测试 diff 上指出, 应将 MiniMax 测试从直接 converter 调用改为 `_feed + _collect_tool_calls` 端到端方式, 提高代表性; 同时收紧文档字符串。作者采纳了建议。
- arpera 发现 Qwen3 也有相同问题, 并要求全面检查其他 parser。作者确认 glm47、deepseek 等不受影响, 并修复了 Qwen3 和 MiniCPM5。
- bbrowning 在合并前发现 Qwen3 流式测试因行为变更而失败, 自己手动修正了预期值; 同时他在 approval 中分析了 `_trim_wrapping_newlines` 的流式安全性: 尾随换行可能暂时缺失但后续到达时会被正确包含, 保持前缀不变性。
- 测试风格: 使用端到端方式替代直接 converter 调用 (testing): 作者采纳并更新了测试代码。
- 扩展修复至 Qwen3 和 MiniCPM5 XML parser (correctness): 作者修复了问题, 并提供了每个 parser 的复现测试。
- Qwen3 流式测试预期值更新 (testing): 测试得以通过, 并在合并前解决。
- `_trim_wrapping_newlines` 在流式场景安全性分析 (correctness): 确认不影响前缀不变性, 结论安全。

风险与影响

- 风险:
 1. 行为破坏: 如果用户先前依赖 strip 行为 (例如自动去除参数值首位空白), 升级后可能接收到含空白参数值, 导致下游处理异常。但这正是 bug 修复, 且符合规范预期。
 2. 流式前缀稳定性: Qwen3 的 `_trim_wrapping_newlines` 在分块不完全时可能暂时缺少一个换行, 但 bbrowning 的分析确认最终输出正确, 前缀不变性保持。
 3. 测试可靠性: 新增的 TestParameterWhitespace 覆盖了完整和流式路径, 但未覆盖跨多轮对话或极高并发场景。
 4. 性能: 仅更改 parser 逻辑, 不影响推理核心, 性能影响可忽略。
- 影响:
 - 用户影响: 使用 MiniMax M2、Qwen3 和 MiniCPM5 模型进行工具调用的用户将获得正确的参数值, 文本编辑等场景可靠性提升。需要客户端适配可能多余的空白。
 - 系统影响: 仅涉及 parser 模块, 不触及推理、调度等核心路径。无性能变化。
 - 团队影响: 代码清晰度提高, 测试增强; 为未来 parser 实现提供了空白保留的范例。

- 风险标记: multiple-parsers-changed, streaming-behavior-change, test-coverage-added

关联脉络

- PR #48753 Qwen3 tool parser whitespace strip issue: 报告了 Qwen3 parser 同样存在 whitespace strip 的问题, 本 PR 修复之。