

PR #53657 完整报告

vllm-project/vllm

[Bugfix] Handle parenthesized Gemma4 tool calls

合并时间: 2026-08-25 20:21

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53657>

执行摘要

- 一句话: 修复 Gemma4 带括号工具调用的解析错误
- 推荐动作: 建议精读此 PR 的状态机转换设计, 尤其是从 TOOL_NAME 安全退出的处理, 理解 vLLM 解析器的设计模式。但需注意其测试覆盖不足, 可参考后续补充测试。

功能与动机

关联 Issue #53642 报告 Gemma4 在输出 `call:name(...)` 形式时解析器会吞掉整段 turn 的内容。PR body 明确指出 parser 只接受 `{...}`, 导致模型输出括号形式时状态机卡在 TOOL_NAME, 损坏工具名并吸收后续调用。

实现拆解

1. 新增 OPEN_PAREN 终端: 在 `gemma4_config()` 的 `terminals` 字典中加入 `OPEN_PAREN: '('`, 使解析器能够识别左括号作为工具参数开始的信号。
2. 增加状态转换: 在 `transitions` 中加入 `(TOOL_NAME, OPEN_PAREN) -> TOOL_ARGS`, 使括号形式和花括号形式一样进入参数解析状态; 同时加入 `(TOOL_NAME, TOOL_END) -> CONTENT`, 用于处理模型中可能出现的无参数工具调用 (直接以 `<tool_call>` 结束) 的情况, 确保状态机安全退出。
3. 调整参数转换器: 修改 `_gemma4_arg_converter`, 当文本末尾是右括号且左括号数少于右括号数时, 移除末尾右括号, 避免解析器将未闭合的括号当作参数内容处理。
4. 测试与配套: PR 中未新增或修改测试文件, 但本地运行了现有的 `test_gemma4_tool_parser.py` 和 `test_gemma4_streaming_reasoning.py`, 全部 111 个测试通过。

关键文件:

- `vllm/parser/gemma4.py` (模块 工具解析; 类别 `source`; 类型 `core-logic`; 符号 `_gemma4_arg_converter, gemma4_config`): 核心修复: 新增 OPEN_PAREN 终端、状态转换和参数转换器调整, 直接解决括号式工具调用解析问题。

关键符号: `gemma4_config, _gemma4_arg_converter`

关键源码片段

`vllm/parser/gemma4.py`

核心修复：新增 OPEN_PAREN 终端、状态转换和参数转换器调整，直接解决括号式工具调用解析问题。

```
# vllm/parser/gemma4.py

def _gemma4_arg_converter(raw_args: str, partial: bool) -> str:
    """Convert Gemma4 custom arg format to a JSON string."""
    text = raw_args.strip()
    # 处理括号式参数：当文本以右括号结尾且左括号数少于右括号数时，
    # 说明末尾括号是多余的，需要移除。
    if text.endswith(")") or text.endswith("(") and text.count("(") < text.count(")"):
        text = text[:-1]
    parsed = _parse_gemma4_args(text, partial=partial)
    return json.dumps(parsed, ensure_ascii=False)

@functools.cache
def gemma4_config() -> ParserEngineConfig:
    # 新增 OPEN_PAREN 终端，使解析器能识别左括号作为参数开始信号。
    terminals={
        ...
        "OPEN_BRACE": "{",
        "OPEN_PAREN": "(",
    },
    transitions={
        # 花括号形式和括号形式都进入 TOOL_ARGS 状态。
        (ParserState.TOOL_NAME, "OPEN_BRACE"): Transition(ParserState.TOOL_ARGS, ()),
        (ParserState.TOOL_NAME, "OPEN_PAREN"): Transition(ParserState.TOOL_ARGS, ()),
        # 安全退出：模型可能直接结束工具调用，此时从 TOOL_NAME 回到 CONTENT。
        (ParserState.TOOL_NAME, "TOOL_END"): Transition(ParserState.CONTENT, (EventType.
        TOOL_CALL_END,)),
        ...
    },
```

评论区精华

foma-agent 在评论中指出，PR 实现了 Issue 中的 Fix A+B，但现有测试无法覆盖括号路径，尤其是 Issue 中第三种场景（一个损坏的 `{` 调用后跟一个标准的 `{` 调用会合并成一个无法使用的 `tool_calls` 条目）。建议补充针对括号路径的测试。yewentao256 给出 LGTM 并合并。

- 测试覆盖不足 (testing): PR 作者未在评论中回应，合并者 yewentao256 给出 LGTM 并合并。测试覆盖问题未解决。

风险与影响

- 风险：
 1. 遗漏测试覆盖：未新增测试用例，回归风险由现有测试和 CI 承担，但括号路径的实际行为未被验证，尤其是不平衡括号和混合调用场景。

2. 状态机行为改变：新增 TOOL_NAME -> TOOL_END 转换可能影响合法的无参数工具调用，但当前实现是合理的安全退出。
3. 参数转换器逻辑：`text.endswith("(")` and `text.count("(") < text.count(")")` 条件对非括号结尾的字符串无影响，但括号内部有 `)` 的情况可能被误处理，需关注。- 影响：影响范围限于 Gemma4 模型的工具解析器（`vllm/parser/gemma4.py`），修复后能正确解析括号式工具调用，避免工具名损坏和内容丢失，提升 Gemma4 模型在 agent 场景下的可靠性。对用户而言，工具调用结果更准确；对系统而言，改动极小（+10/-1），不影响其他模型。- 风险标记：缺少测试覆盖，状态机行为变更

关联脉络

- PR #51262 [Bugfix][DeepSeek V4] Handle trailing system messages in prompt rendering: 同属提示渲染 / 解析类 bugfix，涉及 Rust/Python 渲染器同步修复，可参考其测试补充方式。
- PR #53553 [Bugfix][MM] Fix JinaVL processing cache order: 同为模型输出解析相关 bugfix，涉及缓存与哈希错配，与本 PR 无直接关联但同属解析健壮性改进。