

PR #32522 完整报告

sgl-project/sglang

[Fix]: render tool_reference schema regardless of tool_result part order

合并时间: 2026-07-31 13:57

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/32522>

执行摘要

- 一句话: 修复 GLM 模板对混合 tool_result 丢失 schema 的问题
- 推荐动作: 值得精读。核心看点是: 在协议边界拆分消息而不是修改聊天模板, 以适配不同厂商模板的渲染假设; 同时展现了 review 如何收敛过宽的通用方案。适合关注 Anthropic 兼容层、多模型聊天模板渲染的工程师阅读。建议后续补充更多边界场景测试。

功能与动机

PR body 指出: Anthropic 兼容客户端或代理返回同时包含可见内容与 tool_reference 块的 tool_result 时, GLM-5.1/5.2 会进入无限 ToolSearch 循环。原因是 GLM 模板只在 tool_reference 为消息第一个 content part 时展开引用工具; 若将引用移到最前则又会因引用分支只渲染工具定义而静默丢失文本。因此需要在 Anthropic 边界拆分消息, 让文本与 schema 都能渲染, 同时保持原有 part 顺序与 tool_call_id。

实现拆解

1. 变更入口: python/sglang/srt/entrypoints/anthropic/serving.py 的 `_convert_tool_result_content` 负责把 Anthropic tool_result 的 content 转成 OpenAI 格式的 tool 消息内容。
2. 核心改造: 函数返回类型从 `tuple[Union[str, list[dict]], str]` 改为 `tuple[list[Union[str, list[dict]]], str]`。内部把 `tool_content_parts` 按“是否为 tool_reference”切分成连续片段 (run), 单个纯文本片段折叠为字符串, 其他片段保留列表结构; 空内容以 `[""]` 兜底。
3. 调用点同步: `_emit_user_message` 中遍历 `tool_contents`, 为每个片段生成一条 `role: "tool"` 且共享同一个 `tool_call_id` 的 OpenAI 消息, 保持原 part 顺序。
4. 兼容性保障: 纯文本或纯引用结果仍与改动前等价 (单条消息); 字符串形式的 content 返回单元元素列表; OpenAI Chat Completions 路径未改动。
5. 测试配套: test/registered/unit/entrypoints/anthropic/test_serving.py 新增 `_tool_result_request` 构造器、GLM_TOOL_RESULT_TEMPLATE 模板模拟, 以及 3 个测试用例, 分别覆盖部分顺序保留、GLM 模板渲染文本与 schema、纯引用场景单条消息。相关 Anthropic 与工具内容测试共 61 项通过。

关键文件:

- python/sglang/srt/entrypoints/anthropic/serving.py (模块 消息转换; 类别 source; 类型 core-logic; 符号 `_convert_tool_result_content`, `_emit_user_message`): Anthropic 兼


```

elif item_type == "tool_reference":
    # Anthropic 用 tool_name, SGLang 模板匹配 name, 在边界处翻译
    ref_name = item.get("tool_name") or item.get("name")
    if ref_name:
        tool_content_parts.append(
            {"type": "tool_reference", "name": ref_name}
        )
elif item_type == "search_result":
    search_text = _text_from_search_result(item)
    if search_text:
        tool_text_parts.append(search_text)
        tool_content_parts.append({"type": "text", "text": search_text})

tool_text = "\n".join(tool_text_parts)

# 按“是否 tool_reference”切分连续片段, 保持原始 part 顺序不变
tool_content_groups: list[list[dict]] = []
for part in tool_content_parts:
    is_reference = part["type"] == "tool_reference"
    if (
        not tool_content_groups
        or (tool_content_groups[-1][0]["type"] == "tool_reference")
        != is_reference
    ):
        tool_content_groups.append([])
    tool_content_groups[-1].append(part)

# 每个片段对应一条 tool 消息: 单个纯文本片段折叠成字符串, 其余保留列表
tool_contents: list[Union[str, list[dict]]] = []
for group in tool_content_groups:
    if len(group) == 1 and group[0]["type"] == "text":
        tool_contents.append(group[0]["text"])
    else:
        tool_contents.append(group)
return tool_contents or [""], tool_text

# 非 list 的字符串 content 保持单条消息, 行为与改动前一致
tool_text = str(content) if content else ""
return [tool_text], tool_text

```

片段 2: `_emit_user_message` 中生成多条 `role: "tool"` 消息

```

elif block.type == "tool_result":
    tool_contents, tool_text = _convert_tool_result_content(block.content)

# 每个拆分片段都生成一条独立的 role "tool" 消息, 复用同一个 tool_call_id,
# 这样 GLM 模板可以在每条消息开头判断是否展开 tool_reference
for tool_content in tool_contents:
    openai_messages.append(
        {

```

```
    "role": "tool",
    "tool_call_id": tool_call_id,
    "content": tool_content,
  }
)
```

评论区精华

初审时 JustinTong0323 评论：

“The generic reorder is broader than the affected GLM/Anthropic path and can silently discard mixed tool output.”

即质疑“通用重排”影响面过宽、可能静默丢弃混合工具输出。随后补充提交 **fix: preserve mixed tool result content**，把方案收敛为保持全部内容、按连续片段拆分消息，最终 APPROVED 并合并。该讨论凸显了对协议边界适配作用域的校验：不应为了单一模板改变通用路径的语义。

- 通用重排 vs 按 GLM 模板拆分消息 (design): 维护者 APPROVED 并合并；最终方案不再重排 part，而是拆分工具消息，保留全部文本与 schema。

风险与影响

- 风险：

1. 返回类型变更：_convert_tool_result_content 从单值返回改为列表返回，所有调用点必须同步；当前仅 _emit_user_message 一处调用，已同步修改。
2. 作用域影响：分组逻辑对所有 Anthropic tool_result 生效（不限于 GLM），其他模型的聊天模板可能不期望“一条 tool_result 对应多条 tool 消息”，消息数量变化会影响日志、tokenizer 计数或中间件假设。
3. 边界场景未覆盖：测试未覆盖“多个连续 tool_reference 前后夹文本”“image 与 reference 混合”“空 content”等场景；空 content 会生成 [""] 兜底 tool 消息。
4. 回归风险：GLM 模板在 CJK 字符渲染、引用工具缺省等场景未在测试中验证。 - 影响：用户侧：修复 GLM-5.1/5.2 上 Anthropic 工具的无限 ToolSearch 循环，工具结果中的可见文本与引用 schema 都能完整渲染。系统侧：仅 Anthropic 入口转换层变化，OpenAI 路径不变；单条 tool_result 可能变为多条 tool 消息，需关注依赖消息数的下游逻辑。团队侧：测试覆盖较完整，61 项相关测试通过，CI 已跑通。 - 风险标记：转换层核心路径变更，返回类型升级需调用点同步，拆分逻辑影响所有 Anthropic 工具结果，边界场景测试覆盖不足

关联脉络

- 暂无明显关联 PR