

PR #46091 完整报告

vllm-project/vllm

[Bugfix] [Parser] Fix empty tool block silently dropping subsequent content

合并时间: 2026-06-19 07:17

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/46091>

执行摘要

- 一句话: 修复空 tool block 导致后续内容被丢弃的 bug
- 推荐动作: 建议立即合并。该 PR 修复了在实际使用中已触发的严重 bug (输出截断), 变更范围小, 测试充分。关注其修复模式 (状态机增加边缘 case 转移 + 引擎层条件放宽) 值得在类似解析器中推广。

功能与动机

作者在日常代码审查中, 通过多种编码助手 (Codex CLI + Qwen 3.6-27B、Claude Code + Nemotron 3 Super) 实际触发了该 bug。模型输出类似 `<tool_call>func_name<arg_key>key</arg_key><arg_value>value</arg_value></tool_call>` 的内容时, 解析器检测到开始和结束的 tool_call 特殊 token 但中间无有效内容, 导致卡在 TOOL_PREAMBLE 状态, 后续所有输出被丢弃。本 PR 旨在修复此问题, 确保空 tool block 后正常恢复内容输出。

实现拆解

1. Qwen3 解析器状态转移补充 (vllm/parser/qwen3.py): 在 TOOL_PREAMBLE 状态下新增针对 TOOL_END 标记的转移规则: `(ParserState.TOOL_PREAMBLE, "TOOL_END") → Transition(ParserState.CONTENT, (EventType.TOOL_CALL_END,))`。当解析器处于 TOOL_PREAMBLE 状态且立即遇到结束标记时, 正确回到 CONTENT 状态并发射结束事件。
2. Gemma4 解析器状态转移补充 (vllm/parser/gemma4.py): 与 Qwen3 相同, 在 TOOL_PREAMBLE 状态下新增针对 TOOL_END 的转移规则, 处理空 tool block 场景。
3. Parser Engine 延迟内容释放修正 (vllm/parser/engine/parser_engine.py): 在 `_events_to_delta` 方法中, 原本的释放条件是 `if self._deferred_content and not seen_tool_event:`, 这会在遇到工具事件 (`seen_tool_event=True`) 时阻止延迟内容的释放。但空 tool block 会导致 `seen_tool_event` 为 True 而 `tool_call_deltas` 为空, 此时应该释放内容。修改为 `if self._deferred_content and (not seen_tool_event or not tool_call_deltas):`。
4. 测试配套 (tests/parser/engine/trace_builder.py): 新增 empty-tool-block 场景, 模拟空 tool block 后跟正常内容。同时为 Qwen3、MiniMax M2、Gemma4 的 segments 生成函数增加了处理 `tool_calls=[]` 的分支, 生成相应的 `<tool_call></tool_call>` 特殊 token 序列, 使得测试能够覆盖空 tool block 的端到端行为。

关键文件:

- `vllm/parser/qwen3.py` (模块 解析器; 类别 source; 类型 core-logic) : 核心修复文件之一 : 为 Qwen3 解析器添加 `TOOL_PREAMBLE` → `TOOL_END` 转移, 使空 tool block 能正确恢复状态。
- `vllm/parser/gemma4.py` (模块 解析器; 类别 source; 类型 core-logic) : 核心修复文件之一: 为 Gemma4 解析器添加与 Qwen3 相同的 `TOOL_PREAMBLE` → `TOOL_END` 转移。
- `vllm/parser/engine/parser_engine.py` (模块 解析器; 类别 source; 类型 core-logic) : 引擎层修复: 修正 deferred content 的释放条件, 确保空 tool block 场景下缓存内容能被正常释放到输出流。
- `tests/parser/engine/trace_builder.py` (模块 测试; 类别 test; 类型 test-coverage) : 测试文件: 新增 empty-tool-block 场景并修改 segments 生成函数以支持空工具列表, 为所有解析器提供回归测试覆盖。

关键符号: 未识别

关键源码片段

`vllm/parser/qwen3.py`

核心修复文件之一: 为 Qwen3 解析器添加 `TOOL_PREAMBLE` → `TOOL_END` 转移, 使空 tool block 能正确恢复状态。

```
# vllm/parser/qwen3.py 关键片段
# 在 qwen3_config() 的 transitions 字典中新增以下条目:

# 新增: 当 TOOL_PREAMBLE 状态直接遇到 TOOL_END 标记时 (即空 tool block) ,
# 正确回到 CONTENT 状态并发射 TOOL_CALL_END 事件。
(ParserState.TOOL_PREAMBLE, "TOOL_END"): Transition(
    ParserState.CONTENT,
    (EventType.TOOL_CALL_END,),
),

# 原有的 TOOL_PREAMBLE → FUNC_PREFIX 转移保持不变, 处理正常工具调用。
```

`vllm/parser/gemma4.py`

核心修复文件之一: 为 Gemma4 解析器添加与 Qwen3 相同的 `TOOL_PREAMBLE` → `TOOL_END` 转移。

```
# vllm/parser/gemma4.py 关键片段
# 在 gemma4_config() 的 transitions 字典中, 原有 (TOOL_PREAMBLE, "CALL_PREFIX")
# 之前新增:

(ParserState.TOOL_PREAMBLE, "TOOL_END"): Transition(
    ParserState.CONTENT,
    (EventType.TOOL_CALL_END,),
),

# 注意插入位置: 该转移必须排在 CALL_PREFIX 之前,
# 因为如果同时匹配, 先定义的优先; 空 tool block 应优先匹配 TOOL_END。
```

vllm/parser/engine/parser_engine.py

引擎层修复：修正 deferred content 的释放条件，确保空 tool block 场景下缓存内容能被正常释放到输出流。

```
# vllm/parser/engine/parser_engine.py _events_to_delta 方法
# 第 675 行：修改条件，从 "not seen_tool_event" 变为 "not seen_tool_event or not tool_call_deltas"

if self._deferred_content and (not seen_tool_event or not tool_call_deltas):
    # 释放缓存内容：当没有工具事件，或者有工具事件但没有实际工具 delta 时（空 tool block），
    # 都应该把之前因等待工具而延迟的内容释放到 content_parts 开头。
    content_parts.insert(0, self._deferred_content)
    self._deferred_content = ""
```

评论区精华

该 PR 没有 review 评论，只获得了一个批准（sfeng33: "LGTM, thanks for the work!"）。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更集中在小范围的状态转移规则和单一条件判断上。Qwen3 和 Gemma4 解析器的新增转移仅在 TOOL_PREAMBLE 状态下对 TOOL_END 生效，不影响正常工具调用流程。parser_engine 的修改在逻辑上是正确的，不会导致已存在的工具 delta 被意外跳过。测试覆盖了新增的空 tool block 场景，且通过了 2333 个测试用例。
- 影响：直接影响使用 Qwen3 和 Gemma4 模型且启用工具调用的用户，修复了空 tool block 导致输出截断的问题。间接影响所有使用 parser_engine 的解析器，使 deferred content 释放逻辑更健壮。无性能影响。
- 风险标记：暂无

关联脉络

- PR #46047 [Bugfix] [Parser] Fix Qwen3 latent bug in partial params dropping values containing <: 同样是 Qwen3 解析器相关的 bug 修复，属于同一功能线（tool-calling parser）。