PR #40806 完整报告

vllm-project/vllm

[Bugfix] Fix the DSML token leakage in DSV4/3.2

合并时间：2026-04-26 08:58

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40806

执行摘要

- 一句话：修复 DSV4/3.2 流式推理中 DSML 标记泄漏
- 推荐动作：该 PR 值得仔细阅读，特别是 _extract_content 的设计和 partial_tag_overlap 的使用方式。对于实现流式标记解析的其他 parser（如 hermes, kimi-k2）有参考价值，展示了如何安全地处理跨 chunk 标记边界。

功能与动机

流式调用 tool 时，DSML 起始标记被分片到多个 chunk，导致部分标记字符作为 content 泄漏到客户端。该问题最初在内部 issue #31501 中报告，影响 DeepSeek V4 和 V3.2 模型。PR body 提供了复现脚本。

实现拆解

1. 引入 partial_tag_overlap 工具函数：从 vllm.tool_parsers.utils 导入，用于计算当前文本末尾与起始标记的最大重叠长度。
2. 添加 _sent_content_idx 状态变量：在 __init__ 中初始化，代替原有的 is_tool_call_started 标志，精确追踪已经发送过的内容位置。
3. 实现 _extract_content 方法：根据 partial_tag_overlap 或标记位置计算可发送的索引 sendable_idx，只发送尚未发送且确定不是标记前缀的部分，并将可能的标记前缀保留在缓冲区中。
4. 重写 extract_tool_calls_streaming：移除 is_tool_call_started 和 content_before 的分支逻辑，统一调用 _extract_content 获取当前 chunk 中可发送的纯文本，再调用 _extract_delta_tool_calls 提取已完成的 invoke 块。
5. 更新 _reset_streaming_state：重置 _sent_content_idx。
6. 新增 5 个测试用例：覆盖按块流式、按字符流式、所有可能分片边界、带前缀内容、以及伪标记（非标记的相似文本）场景。

关键文件：

- vllm/tool_parsers/deepseekv32_tool_parser.py（模块 工具解析；类别 source；类型 core-logic；符号 _extract_content）：核心修复文件，新增 _extract_content 方法，修改 extract_tool_calls_streaming 和 _reset_streaming_state，引入 partial_tag_overlap 防泄漏。

- tests/tool_parsers/test_deepseekv32_tool_parser.py（模块 测试；类别 test；类型 test-coverage；符号 test_no_marker_leak_chunked, test_no_marker_leak_with_prefix_chunked, test_no_marker_leak_char_by_char, test_no_marker_leak_all_split_points）：新增 5 个测试用例验证标记泄漏修复，覆盖多种分片场景。

关键符号：_extract_content, extract_tool_calls_streaming, _reset_streaming_state

关键源码片段

vllm/tool_parsers/deepseekv32_tool_parser.py

核心修复文件，新增 _extract_content 方法，修改 extract_tool_calls_streaming 和 _reset_streaming_state，引入 partial_tag_overlap 防泄漏。

```
def _extract_content(self, current_text: str) -> str | None:
    # 如果 start token 不在当前文本中，检查末尾是否为 start token 的前缀
    if self.tool_call_start_token not in current_text:
        # partial_tag_overlap 返回 current_text 末尾与 start token 的最大重叠字符数
        overlap = partial_tag_overlap(current_text, self.tool_call_start_token)
        # 只发送到重叠开始之前的部分，可能的前缀保留到下次
        sendable_idx = len(current_text) - overlap
    else:
        # 如果 start token 已出现，则只发送它之前的部分
        sendable_idx = current_text.index(self.tool_call_start_token)

    # 只发送尚未被发送的部分
    if sendable_idx > self._sent_content_idx:
        content = current_text[self._sent_content_idx : sendable_idx]
        self._sent_content_idx = sendable_idx
        return content
    return None

def extract_tool_calls_streaming(self, previous_text, current_text, delta_text, request):
    ...
    # 用 _extract_content 获取当前可发送的纯文本（不含标记前缀）
    content = self._extract_content(current_text)
    delta_tool_calls = self._extract_delta_tool_calls(current_text, request)
    if delta_tool_calls or content:
        return DeltaMessage(content=content, tool_calls=delta_tool_calls)
    ...
```

评论区精华

Gemini Code Assist 指出了早期实现中数据丢失的风险：当检测到潜在前缀时返回 None 跳过当前 delta_text，若后续非标记则丢失数据；同时测试 chunk_size 从 7 改为 12 掩盖了问题。sfeng33 确认评论有效，并在后续迭代中采用 _extract_content 加 partial_tag_overlap 的方案，利用 _sent_content_idx 精确追踪已发送位置，确保不论是否最终为标记，内容都不会丢失。最终评审通过。

- 流式 buffer 数据丢失风险 & 测试调整掩盖问题 (correctness): 采用了基于 _sent_content_idx 和 partial_tag_overlap 的缓冲方案，确保标记前缀在确认前被保留，确认非标记后仍可发送，不丢失数据。测试未使用大 chunk_size，验证了所有分片边界。

风险与影响

- 风险：核心风险是 partial_tag_overlap 的准确性：若重叠计算有误，可能导致内容多丢或少丢。但该函数是现有工具函数，已在其他 parser（如 hermes, kimi-k2）中验证过。兼容性风险：移除 is_tool_call_started 标志可能影响依赖该标志的外部代码（但此标志是内部状态，外部不可见）。性能风险：增加了一次 partial_tag_overlap 调用，复杂度 O(n)，但 n 是当前 chunk 长度，通常较小，影响可忽略。
- 影响：影响范围：所有使用 DeepSeek V4 或 V3.2 模型并启用 tool call 的流式推理用户。修复后，tool call 标记将不再泄漏到 content 中，提升了输出的准确性和一致性。新增的 5 项测试覆盖了多种分片边界，保证了回归防护。对非流式模式无影响。
- 风险标记：流式 buffer 逻辑复杂度，依赖 partial_tag_overlap 正确性，测试覆盖了分片边界

关联脉络

- 暂无明显关联 PR