PR #47379 完整报告

vllm-project/vllm

[Bugfix][Frontend][gpt-oss] Recover raw tail when Harmony parser ends non-terminal

合并时间：2026-07-04 22:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47379

执行摘要

- 一句话：修复 Harmony 解析器非终止状态内容丢失
- 推荐动作：建议立即合并。此 PR 解决了 issue #45736 报告的静默数据丢失问题，设计简洁，测试完善，且已获得审阅者 bbrowning 和 mgoin 的批准。

功能与动机

修复 issue #45736：vLLM 的 GPT-OSS Harmony 输出处理缺少终止状态校验与内容回退，在解析器非终止时静默返回 content: null、finish_reason="stop"，不记录异常也不尝试恢复。

实现拆解

1. vllm/parser/harmony.py：引入 _current_message_tokens 列表，在 process_chunk 中累积当前消息的 token ID。重写 flush() 方法，将 HarmonyError 从外部传播改为内部捕获：解码 _current_message_tokens 获得原始文本，构造一个 Segment（channel="final", delta= 文本 , completed_message=None）作为恢复输出，并模拟生成一个 Message 对象维持下游兼容。移除 flush() 的 finally 块中的重置逻辑，改为在异常处理部分统一重置。parse() 和 parse_delta() 中移除针对 HarmonyError 的 try-except，直接使用 flush() 返回的列表。
2. tests/parser/test_harmony.py：新增 malformed_msgs_str fixture 和 assert_parser_is_reset 辅助函数。将 test_flush_raises_and_resets_on_non_terminal_eos 替换为 test_flush_recovers_invalid_output、test_malformed_final_recovers_raw_content 等，验证 flush 在非终止状态下返回恢复的原始尾部内容，且解析器状态被重置。
3. vllm/entrypoints/openai/responses/context.py：适配 flush() 新签名。HarmonyContext.append_output 中调用 flush() 后得到的列表直接 extend 到 segments，并更新 last_append_flush_status 为 bool 类型，去除对 HarmonyError 的引用。HarmonyContext.__init__ 中 last_append_flush_status 类型改为 bool。
4. tests/entrypoints/unit_tests/test_context.py：更新 FakeHarmonyParser 的 _flush_results 类型为 list[list[Segment]]，enqueue_flush_result 接受 list[Segment]，flush() 返回 list[Segment]（空列表替代 None）。测试用例中使用新的 flush 结果结构验证上下文同步逻辑。

关键文件：

- vllm/parser/harmony.py（模块 解析器；类别 source；类型 core-logic；符号 flush, _current_message_tokens）：核心修复文件：修改 flush() 使其在 HarmonyError 时恢复原始尾部，而非抛出异常；新增 _current_message_tokens 累积 token；调整 parse()/parse_delta() 的错误处理。
- tests/parser/test_harmony.py（模块 测试；类别 test；类型 test-coverage；符号 malformed_msgs_str, assert_parser_is_reset, test_flush_recovers_invalid_output, test_malformed_final_recovers_raw_content）：测试文件：新增 fixture malformed_msgs_str、辅助函数 assert_parser_is_reset，替换原异常测试为恢复验证测试，覆盖各种非终止场景。
- vllm/entrypoints/openai/responses/context.py（模块 上下文；类别 source；类型 dependency-wiring；符号 append_output, HarmonyContext.init）：适配层文件：更新 append_output 以使用 flush 的新返回类型，移除对 HarmonyError 的依赖。
- tests/entrypoints/unit_tests/test_context.py（模块 测试；类别 test；类型 test-coverage；符号 enqueue_flush_result, flush）：测试适配文件：更新 FakeHarmonyParser 和测试用例以匹配新的 flush 返回类型和恢复行为。

关键符号：HarmonyParser.flush, HarmonyParser.parse, HarmonyParser.parse_delta, HarmonyContext.append_output

关键源码片段

vllm/parser/harmony.py

核心修复文件：修改 flush() 使其在 HarmonyError 时恢复原始尾部，而非抛出异常；新增 _current_message_tokens 累积 token；调整 parse()/parse_delta() 的错误处理。

```
    def flush(self) -> list[Segment]:
        segments: list[Segment] = []
        try:
            self._harmony_parser.process_eos()
            msg = self._poll_completed_message()
        except HarmonyError:
            # 解析器在非终止状态结束，无法完成结构化解析。
            # 转而恢复原始输出：将累积的 token 解码为文本，
            # 并作为 final channel 的一个 Segment 提供给下游。
            logger.warning(
                "Harmony parser ended in a non-terminal state; returning the "
                "recovered raw output."
            )
            final_channel = "final"
            # 使用模型 tokenizer 解码当前消息 token
            text = self.model_tokenizer.decode(self._current_message_tokens)
            segments.append(
                Segment(
                    channel=final_channel,
                    recipient=None,
                    delta=text,
                    completed_message=None, # 无结构化消息
```

```
                )
            )
            # 构造一个 Message 对象以兼容后续处理流程
            msg = Message.from_role_and_content(Role.ASSISTANT, text).with_channel(
                final_channel
            )

        # 无论是否出错，都重置解析器状态以准备下一轮
        self._parser = None
        self._num_processed_messages = 0
        self._current_message_tokens.clear()

        if msg is None:
            return segments

        segments.append(
            Segment(
                channel=msg.channel,
                recipient=msg.recipient,
                delta="",
                completed_message=msg,
            )
        )
        return segments
```

tests/parser/test_harmony.py

测试文件：新增 fixture malformed_msgs_str、辅助函数 assert_parser_is_reset，替换原异常测试为恢复验证测试，覆盖各种非终止场景。

```
@pytest.fixture
def malformed_msgs_str() -> list[str]:
    return [
        "<|channel|>analysis<|message|>thinking<|end|>",
        "<|start|>assistant<|channel|>commentary<|message|>thinking<|end|>",
        '<|start|>assistant<|channel|>final {"answer": "hi"}<|return|>',
    ]

def assert_parser_is_reset(harmony_parser: HarmonyParser):
    assert harmony_parser._parser is None
    assert harmony_parser._num_processed_messages == 0
    assert harmony_parser._current_message_tokens == []

class TestFlush:
    def test_flush_recovers_invalid_output(self, harmony_parser, malformed_msgs_str):
        # 先处理前两个 malformed 片段（非 terminal EOS 触发异常前的正常 chunk）
        for msg_str in malformed_msgs_str[:-1]:
            chunk = harmony_parser.process_chunk(encode_output(msg_str))
            assert "".join(segment.delta for segment in chunk.segments) == "thinking"
```

```
# 最后一个片段以 <|return|> 结束，但缺少 <|message|> 导致解析器非终止
last_msg_str = malformed_msgs_str[-1]
harmony_parser.process_chunk(encode_output(last_msg_str))
flushed_segments = harmony_parser.flush()
# flush 现在应返回两个 segments：第一个是恢复的原始 tail，第二个是解析后的 Message
assert len(flushed_segments) == 2
delta_segment = flushed_segments[0]
message_segment = flushed_segments[1]

# 恢复的 segment 包含原始文本，channel 为 "final"
assert delta_segment.channel == "final"
assert delta_segment.recipient is None
assert delta_segment.delta == last_msg_str
# 解析后的 Message 也包含相同文本
assert message_segment.channel == "final"
assert message_segment.recipient is None
assert get_text(message_segment.completed_message) == last_msg_str
# 验证解析器已重置
assert_parser_is_reset(harmony_parser)
```

评论区精华

审阅者 bbrowning 批准并指出："This is a reasonable improvement. A future improvement that we discussed would be to see if we can recover that latest message a bit better, stripping the special tokens, role, and channel markers as well as EOS tokens if found which would leave us with just the important bits. But, that's additional scope and this is already a nice iterative improvement in handling this error." 该评论被作者认可，留作后续改进。

- 未来改进：去除恢复文本中的特殊标记 (design): 作者 yzong-rh 认可并留作后续改进，当前 PR 聚焦于基本的恢复机制。

风险与影响

- 风险：低风险。变更覆盖了修复的核心逻辑和匹配的测试用例，没有触及推理或工具调用核心路径之外的模块。flush() 返回类型变更影响 context.py 及测试，均已同步更新。潜在风险是恢复的原始文本可能包含特殊标记（如 channel 开头符），但这是已知的、可接受的限制（已计划后续改进）。
- 影响：影响范围限定于使用 GPT-OSS Harmony 解析器的 Responses API 用户。修复前，当解析器非终止时用户收到 content: null 的成功响应，产生数据丢失错觉；修复后，用户将得到模型生成的原始尾部内容，至少避免静默丢失。对正常路径无影响。团队维护成本低，测试覆盖充分。
- 风险标记：暂无

关联脉络

- PR #45736 [Bug]: GPT-OSS Harmony: vLLM silently returns content: null with finish_reason="stop" when its parser ends in a non-terminal state: 直接关联的 issue，描述了此 PR 修复的 bug