PR #44283 完整报告

vllm-project/vllm

[Anthropic] Support system role messages inside messages array

合并时间：2026-06-03 02:13

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44283

执行摘要

- 一句话：支持 Anthropic messages 数组内联 system 角色
- 推荐动作：该 PR 解决了一个实际的客户端兼容性问题，实现简洁且测试充分，推荐合并。设计上值得关注的点是：通过先收集再合并的方式处理两处 system 信息来源，而不是分别追加，避免消息顺序错误。

功能与动机

根据 issue #44000，Claude Code CLI 2.1.154+ 版本会在 messages 数组中发送 role=system 的消息，而 vLLM 的 Anthropic 协议仅允许 user 和 assistant 角色，导致请求被拒绝并返回 400 错误。本 PR 扩展了角色枚举并调整了消息处理逻辑，以兼容此类客户端行为。

实现拆解

1. 扩展消息角色枚举：在 vllm/entrypoints/anthropic/protocol.py 中，将 AnthropicMessage.role 的类型从 Literal['user', 'assistant'] 改为 Literal['user', 'assistant', 'system']，使得 Pydantic 模型能够接受 system 角色。
2. 重构系统消息处理：在 vllm/entrypoints/anthropic/serving.py 的 _convert_system_message 方法中，不再仅处理顶层 system 字段，而是先收集顶层 system 内容，然后遍历 messages 数组，提取所有 role=system 的消息内容（支持字符串和内容块），并统一拼接到 system_parts 列表中。最后将所有部分合并为一个 system 消息。
3. 跳过重复处理：在 _convert_messages 方法中，增加对 role=system 的跳过逻辑，避免将内联 system 消息再次转换为普通消息导致重复。
4. 测试覆盖：新增 TestInlineSystemMessageInMessagesArray 测试类，包含五个测试用例，覆盖内联 system 与顶层 system 合并、纯字符串内联、列表内容内联、多个内联 system 以及内联 system 与顶层 system 字符串共存等场景。

关键文件：

- tests/entrypoints/anthropic/test_anthropic_messages_conversion.py（模块 测试；类别 test；类型 test-coverage；符号 TestInlineSystemMessageInMessagesArray, test_inline_system_merged_with_top_level_system, test_inline_system_string_only, test_inline_system_list_content）：新增 140 行测试，全面覆盖内联 system 消息的合并、

字符串、列表、多个内联、与顶层 system 共存等场景，是验证正确的关键。

- vllm/entrypoints/anthropic/serving.py（模块 服务层；类别 source；类型 core-logic；符号 _convert_system_message, _convert_messages）：核心变更：重构 _convert_system_message 以支持从 messages 数组中提取 system 消息；_convert_messages 跳过 system 角色。
- vllm/entrypoints/anthropic/protocol.py（模块 协议层；类别 source；类型 core-logic；符号 AnthropicMessage）：角色枚举扩展，允许 messages 数组中出现 system 角色。

关键符号：_convert_system_message, _convert_messages, AnthropicMessage

关键源码片段

vllm/entrypoints/anthropic/serving.py

核心变更：重构 _convert_system_message 以支持从 messages 数组中提取 system 消息；_convert_messages 跳过 system 角色。

```
@classmethod
def _convert_system_message(
    cls,
    anthropic_request: AnthropicMessagesRequest | AnthropicCountTokensRequest,
    openai_messages: list[dict[str, Any]],
) -> None:
    """Convert Anthropic system message to OpenAI format.
    Now also extracts system messages embedded in the messages array.
    """
    system_parts: list[str] = []

    # 1. Process top-level system field
    if anthropic_request.system:
        if isinstance(anthropic_request.system, str):
            system_parts.append(anthropic_request.system)
        else:
            for block in anthropic_request.system:
                if block.type == "text" and block.text:
                    # Strip Claude Code's attribution header to improve prefix caching
                    if block.text.startswith("x-anthropic-billing-header"):
                        continue
                    system_parts.append(block.text)

    # 2. Extract inline system messages from the messages array
    for msg in anthropic_request.messages:
        if msg.role != "system":
            continue
        if isinstance(msg.content, str):
            system_parts.append(msg.content)
        else:
            for block in msg.content:
                if block.type == "text" and block.text:
```

```
                if block.text.startswith("x-anthropic-billing-header"):
                    continue
                system_parts.append(block.text)

    # 3. Emit a single merged system message
    if system_parts:
        openai_messages.append({"role": "system", "content": "".join(system_parts)})


@classmethod
def _convert_messages(
    cls, messages: list, openai_messages: list[dict[str, Any]]
) -> None:
    """Convert Anthropic messages to OpenAI format, skipping system messages."""
    for msg in messages:
        if msg.role == "system":
            continue # Already handled in _convert_system_message
        openai_msg: dict[str, Any] = {"role": msg.role}
        if isinstance(msg.content, str):
            openai_msg["content"] = msg.content
        else:
            cls._convert_message_content(msg, openai_msg, openai_messages)
        if not (msg.role == "user" and "content" not in openai_msg):
            openai_messages.append(openai_msg)
```

评论区精华

无 reviewer 讨论，仅有一位贡献者（aleksandaryanakiev）表示该方案更好，关闭了自己的 PR。没有未解决的争议。

- 暂无高价值评论线程

风险与影响

- 风险：本次变更仅放宽了请求验证和增加了消息处理逻辑，对现有请求兼容。风险较低：
  - 合并顺序：顶层 system 内容先于内联 system 内容，如果用户同时提供两者，内联 system 会追加在后面，这可能改变某些依赖顺序的客户端行为，但 Anthropic 官方 API 也支持两者。
  - 性能影响：增加了一次遍历 messages 数组的步骤，但仅在转换时，影响微乎其微。
  - 向后兼容：所有之前合法的请求仍然合法，且处理逻辑一致。
  - 影响：用户影响：之前因为 400 错误无法使用的 Claude Code CLI 2.1.154+ 用户现在可以正常工作。对于其他使用 Anthropic 兼容 API 的客户端，内联 system 消息也被支持。

系统影响：处理路径没有增加显著开销。

团队影响：无，单文件改动且经过测试。

- 风险标记：兼容性变更，低风险

关联脉络

- 暂无明显关联 PR