PR #40190 完整报告

vllm-project/vllm

[Frontend] Add `defer_loading` and `tool_reference` support for Anthropic and OpenAI APIs

合并时间：2026-04-29 19:35

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40190

执行摘要

- 一句话：为 Anthropic 和 OpenAI API 添加 defer_loading 和 tool_reference 支持
- 推荐动作：值得精读，尤其是理解前端消息转换架构和 Pydantic 序列化技巧。建议尽快补充单元测试和 E2E 测试（可使用 Qwen3 模型），并验证序列化签名在目标 Pydantic 版本下的兼容性。关注 _convert_block 中顶级 tool_reference 的丢失问题。

功能与动机

支持 Anthropic 的 tool search / defer loading 和 OpenAI 的等效功能，使标记了 defer_loading: true 的工具在初始提示中被排除，而通过多轮对话中的 tool_reference 块按需加载。参考了 SGlang 实现和官方文档。

实现拆解

1. 协议扩展：在 AnthropicTool 和 FunctionDefinition 中添加 defer_loading 字段；在 AnthropicContentBlock 中增加 tool_reference 类型和 tool_name 字段；新增 CustomChatCompletionContentToolReferenceParam TypedDict。
2. 请求序列化调整：在 ChatCompletionToolsParam 中通过 model_validator 自动传播 defer_loading 到 FunctionDefinition；通过 model_serializer 在序列化时移除 None 字段，避免影响 chat template 的键值检查。
3. Anthropic→OpenAI 消息转换：在 _convert_block 中注册 tool_reference 分支（当前仅 pass）；在 _convert_user_tool_result 中提取 tool_result 内容中的 tool_reference 项，作为额外的 tool 角色消息附加；在 _convert_tools 中传递 defer_loading。
4. 消息解析增强：在 MM_PARSER_MAP 中注册 tool_reference 解析入口；在 _parse_chat_message_content_mm_part 中处理简写格式；在 _parse_chat_message_content_part 中对 tool_reference 采用透传；修改工具角色内容归一化逻辑，保留非文本项。

配套测试仅手动验证，无自动化测试。

关键文件：

- vllm/entrypoints/chat_utils.py（模块 消息解析；类别 source；类型 core-logic；符号 CustomChatCompletionContentToolReferenceParam）：定义了 tool_reference 内容类型的 TypedDict，扩展了内容部分联合类型，添加了解析映射和分支处理，是消息解析的核心文件。

- vllm/entrypoints/openai/chat_completion/protocol.py（模块 协议定义；类别 source；类型 core-logic；符号 _propagate_defer_loading, _serialize）：在 ChatCompletionToolsParam 中添加 defer_loading 字段、传播逻辑和序列化方法，是 OpenAI 协议层的核心变更。
- vllm/entrypoints/openai/engine/protocol.py（模块 引擎协议；类别 source；类型 core-logic；符号 _serialize）：在 FunctionDefinition 中添加 defer_loading 字段和序列化方法，支持工具级别或函数级别指定。
- vllm/entrypoints/anthropic/serving.py（模块 转换服务；类别 source；类型 core-logic；符号 _convert_block, _convert_user_tool_result, _convert_tools）：更新 Anthropic→OpenAI 消息转换逻辑，提取 tool_reference 并附加消息，传递 defer_loading。
- vllm/entrypoints/anthropic/protocol.py（模块 协议定义；类别 source；类型 core-logic；符号 AnthropicTool, AnthropicContentBlock）：扩展 AnthropicTool 和 AnthropicContentBlock 以支持 defer_loading 和 tool_reference 类型。

关键符号：CustomChatCompletionContentToolReferenceParam, _propagate_defer_loading, _serialize, _convert_block, _convert_user_tool_result, _convert_tools

关键源码片段

vllm/entrypoints/chat_utils.py

定义了 tool_reference 内容类型的 TypedDict，扩展了内容部分联合类型，添加了解析映射和分支处理，是消息解析的核心文件。

```
# CustomChatCompletionContentToolReferenceParam：表示一个工具引用内容部分，
# 用于在消息中引用之前 defer 的工具。
class CustomChatCompletionContentToolReferenceParam(TypedDict, total=False):
    """A tool reference content param that only accepts a plain tool name.

    Example:
    {
        "name": "get_weather",
        "type": "tool_reference"
    }
    """

    name: str
    """The name of the tool being referenced."""

    type: Literal["tool_reference"]
    """The content type."""


# 将新类型加入 ChatCompletionContentPartParam 联合类型
ChatCompletionContentPartParam: TypeAlias = (
    OpenAIChatCompletionContentPartParam
```

```
    | ChatCompletionContentPartAudioParam
    | ChatCompletionContentPartInputAudioParam
    | ChatCompletionContentPartVideoParam
    | ChatCompletionContentPartRefusalParam
    | CustomChatCompletionContentPILImageParam
    | CustomChatCompletionContentSimpleImageParam
    | ChatCompletionContentPartImageEmbedsParam
    | ChatCompletionContentPartAudioEmbedsParam
    | CustomChatCompletionContentSimpleAudioParam
    | CustomChatCompletionContentSimpleVideoParam
    | CustomChatCompletionContentToolReferenceParam # 新增
    | str
    | CustomThinkCompletionContentParam
)


# 在 MM_PARSER_MAP 中注册 tool_reference 解析
MM_PARSER_MAP: dict[...] = {
    # ... 其他类型 ...
    "tool_reference": lambda part: cast(
        CustomChatCompletionContentToolReferenceParam, part
    ).get("name", None),
}


# 在 _parse_chat_message_content_mm_part 中处理 tool_reference 简写
if "tool_reference" in part:
    tool_reference_params = cast(
        CustomChatCompletionContentToolReferenceParam, part
    )
    tool_reference = tool_reference_params.get("name", None)
    return "tool_reference", tool_reference
```

vllm/entrypoints/openai/chat_completion/protocol.py

在 ChatCompletionToolsParam 中添加 defer_loading 字段、传播逻辑和序列化方法，是 OpenAI 协议层的核心变更。

```
# ChatCompletionToolsParam: 包装 function 的 tool 参数，增加 defer_loading 支持
class ChatCompletionToolsParam(OpenAIBaseModel):
    type: Literal["function"] = "function"
    function: FunctionDefinition
    defer_loading: bool | None = None # 新增字段

    @model_validator(mode="after")
    def _propagate_defer_loading(self) -> "ChatCompletionToolsParam":
        # 当 Tool 级别设置了 defer_loading 而 Function 级别未设置时，
        # 自动传播到 FunctionDefinition 中，避免重复指定。
        if self.defer_loading is not None and self.function.defer_loading is None:
            self.function.defer_loading = self.defer_loading
```

```
        return self

    @model_serializer(mode="wrap")
    def _serialize(self, handler):
        # 序列化时移除 defer_loading 字段（若为 None），以免干扰 chat template
        data = handler(self)
        if self.defer_loading is None:
            data.pop("defer_loading", None)
        return data
```

评论区精华

- 顶级 tool_reference 块处理：gemini-code-assist 指出 _convert_block 中 tool_reference 分支为 pass，可能丢失模型直接输出的内容。作者认为只在 tool_result 内处理即可，但未充分讨论模型自主输出场景。状态 partially resolved。
- 序列化签名不完整：Copilot 指出 _serialize 方法仅接受 handler，但 Pydantic v2 要求接受 handler 和 info，可能引发 TypeError。代码未修改，存在隐患。
- TypedDict 字段缺少 Required：Copilot 建议对 name 和 type 使用 Required，与其他内容部分 TypedDict 保持一致。未采纳。
- defer_loading 传播优先级模糊：Copilot 指出 Tool 级别与 Function 级别同时设置时的冲突处理不明确。当前逻辑是 Tool 仅补充 Function 未设时的值，若两者均设则 Function 优先。视为设计决定。
  - 顶级 tool_reference 块处理 (correctness): 作者解释只在 tool_result 内提取处理，但未充分探讨模型直接输出 tool_reference 的场景，该分支仍为 pass。
  - model_serializer 签名缺失 info 参数 (correctness): 代码未修改，认为在当前 pydantic 版本中可能兼容，但存在隐患未解决。
  - TypedDict 字段缺少 Required (style): 作者未处理，视为样式不一致，无运行时影响。
  - defer_loading 传播优先级模糊 (design): 当前逻辑是 Tool 仅补充 Function 未设时的值；若两者均设则 Function 优先。无冲突检测，视为设计决定。

风险与影响

- 风险：
  - 序列化签名不兼容：FunctionDefinition._serialize 和 ChatCompletionToolsParam._serialize 签名缺少 info 参数，若使用较新 Pydantic v2 版本可能触发 TypeError，阻断请求序列化。
  - 顶级 tool_reference 被吞没：如果模型直接输出 tool_reference 块（而非在 tool_result 内），当前 _convert_block 仅 pass 会导致该块被忽略，破坏对话完整性。
  - 测试缺失：无单元测试或 E2E 测试，功能正确性依赖手动验证，回归风险高。
  - 类型系统不严谨：CustomChatCompletionContentToolReferenceParam 字段未标记 Required，类型检查器无法捕获非法输入。
- 影响：

- 用户：支持 Anthropic 和 OpenAI 工具延迟加载，提升多轮工具使用体验，尤其适合大型工具集场景。
- 系统：无性能影响，仅增加少量字段和条件分支。
- 团队：提供可扩展的 tool_reference 基础设施，方便未来实现按需工具加载和工具搜索。
- 风险标记：序列化签名不兼容，顶级 tool_reference 丢失，测试覆盖缺失

关联脉络

- PR #41110 [Frontend]Responses API supports Tool/Function calling with streaming with named tool/function: 该 PR 也修改了 chat_utils.py 和 protocol.py，涉及 tool-calling 前端支持，且当前 PR 的 commit 历史中包含合并 #41110 的记录。