PR #22702 完整报告

sgl-project/sglang

Support defer_loading field at function level for Chat Completions API

合并时间：2026-04-23 01:09

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/22702

执行摘要

- 一句话：为 Chat Completions API 添加函数级 defer_loading 字段支持，实现工具延迟加载和 GLM 特定扩展。
- 推荐动作：建议精读此 PR，特别是 protocol.py 中的 Pydantic 模型扩展和序列化逻辑，以学习如何优雅地添加可选字段并控制序列化行为。同时关注 serving.py 中的 Anthropic 集成方式，理解跨 API 协议映射的设计权衡。对于涉及协议扩展的项目，此 PR 提供了处理厂商特定扩展的实用模式。

功能与动机

根据 PR body，添加 defer_loading 支持是为了让标记为 defer_loading: true 的工具从模型的工具列表中隐藏，并通过 chat template（如 GLM-5.1）处理过滤。这是一个 GLM 特定的扩展，用于实现工具的动态加载和跨轮次解锁，而非 OpenAI API 的标准功能。

实现拆解

1. 扩展 OpenAI 协议模型：在 python/sglang/srt/entrypoints/openai/protocol.py 中，为 Function 类添加 defer_loading: Optional[bool] 字段和自定义序列化器 _serialize，确保字段仅在设置时序列化；为 Tool 类添加 defer_loading 字段和 _propagate_defer_loading 验证器，以从工具级别传播值到函数级别。同时新增 ChatCompletionMessageContentToolReferenceBlock 内容类型，用于 GLM 的 tool_reference 扩展。
2. 集成 Anthropic 端点逻辑：在 python/sglang/srt/entrypoints/anthropic/serving.py 中，修改工具转换逻辑以传递 defer_loading 字段，并添加 tool_reference 内容类型的处理，将 Anthropic 的 tool_name 字段映射为 name 以匹配聊天模板。同时调整 tool_choice 处理，确保当工具被过滤时正确处理。
3. 更新 Anthropic 协议定义：在 python/sglang/srt/entrypoints/anthropic/protocol.py 中，为 AnthropicContentBlock 添加 tool_reference 类型，并为 AnthropicTool 添加 defer_loading 字段，以支持请求解析。
4. 传递 tool_reference 到模板：在 python/sglang/srt/parser/jinja_template_utils.py 中，添加对 tool_reference 内容类型的处理，使其能通过模板路径传递，供聊天模板使用。
5. 添加测试覆盖：在 test/registered/unit/entrypoints/openai/test_protocol.py 中，新增 TestFunctionDeferLoading 测试类，验证 defer_loading 字段的序列化、传播和

tool_reference 内容类型的接受性。

关键文件：

- python/sglang/srt/entrypoints/openai/protocol.py（模块 协议模型；类别 source；类型 core-logic；符号 ChatCompletionMessageContentToolReferenceBlock, _serialize, _propagate_defer_loading）：核心协议模型变更，添加 defer_loading 字段、tool_reference 内容类型及序列化逻辑，是功能实现的基础。
- test/registered/unit/entrypoints/openai/test_protocol.py（模块 协议测试；类别 test；类型 test-coverage；符号 TestFunctionDeferLoading, test_function_defaults_preserve_strict, test_function_defer_loading_true_serialized, test_function_defer_loading_false_serialized）：测试配套变更，验证 defer_loading 字段的序列化、传播和 tool_reference 内容类型的接受性，确保功能正确性。
- python/sglang/srt/entrypoints/anthropic/serving.py（模块 Anthropic 服务；类别 source；类型 core-logic）：Anthropic 端点实现，集成 defer_loading 过滤和 tool_reference 解锁逻辑，是关键的业务逻辑变更点。
- python/sglang/srt/entrypoints/anthropic/protocol.py（模块 Anthropic 协议；类别 source；类型 data-contract）：Anthropic 协议定义更新，添加 tool_reference 内容类型和 defer_loading 字段，支持请求解析。
- python/sglang/srt/parser/jinja_template_utils.py（模块 模板解析；类别 source；类型 core-logic）：模板处理工具更新，确保 tool_reference 内容类型能传递到聊天模板，是功能集成的最后一环。

关键符号：_serialize, _propagate_defer_loading

关键源码片段

python/sglang/srt/entrypoints/openai/protocol.py

核心协议模型变更，添加 defer_loading 字段、tool_reference 内容类型及序列化逻辑，是功能实现的基础。

```
class Function(BaseModel):
    """Function descriptions."""
    description: Optional[str] = Field(default=None, examples=[None])
    name: str
    parameters: Optional[object] = None
    strict: bool = False # 保持默认值以确保下游代码（如 function_call_parser）看到预期形状
    defer_loading: Optional[bool] = None # 新增字段，用于标记工具是否延迟加载

    @model_serializer(mode="wrap")
    def _serialize(self, handler):
        data = handler(self)
        if self.defer_loading is None:
            data.pop("defer_loading", None) # 当 defer_loading 为 None
            时，从序列化输出中移除该字段，避免污染 JSON
        return data

class Tool(BaseModel):
```

```
    """Function wrapper."""
    type: str = Field(default="function", examples=["function"])
    function: Function
    defer_loading: Optional[bool] = None # 工具级别的 defer_loading 字段，可作为 function
    级别的快捷设置

    @model_validator(mode="after")
    def _propagate_defer_loading(self) -> "Tool":
        if self.defer_loading is not None and self.function.defer_loading is None:
            self.function.defer_loading = self.defer_loading #
            将工具级别的值传播到函数级别，便于统一处理
        return self
```

python/sglang/srt/entrypoints/anthropic/serving.py

Anthropic 端点实现，集成 defer_loading 过滤和 tool_reference 解锁逻辑，是关键的业务逻辑变更点。

```
def _convert_tool_result_content(content):
    if isinstance(content, list):
        tool_content_parts = []
        for item in content:
            item_type = item.get("type")
            if item_type == "tool_reference":
                # Anthropic SDK 使用 `tool_name` 字段，但 SGLang 聊天模板匹配 `name`，在此转换
                ref_name = item.get("tool_name") or item.get("name")
                if ref_name:
                    tool_content_parts.append({"type": "tool_reference", "name": ref_name})
            # 其他类型处理（如 text、image）省略
        return tool_content_parts
    # 简化示例，实际函数更复杂
```

评论区精华

Review 中，JustinTong0323 指出了关键 bug：在 anthropic/serving.py 中，item.get("name") 应改为 item.get("tool_name") 以匹配 Anthropic SDK v0.93.0 的 ToolReferenceBlockParam，否则tool_reference解锁路径无效。此外，讨论了测试覆盖不足、建议添加日志以便调试过滤逻辑，以及 tool_choice 处理中当工具被过滤时应返回错误而非静默降级。结论是 bug 被修复，但测试和日志建议未完全实现，且设计上明确此为 GLM 扩展而非 OpenAI 标准。

- Anthropic 端点中 tool_reference 字段名错误 (correctness): Bug 被修复，确保 tool_reference 能正确解锁延迟工具。
- 测试覆盖和日志建议 (testing): 测试已部分添加，但日志建议未实现，可能影响可调试性。
- tool_choice 处理逻辑改进 (design): 逻辑已调整，但具体实现细节未在 review 中完全确认。

风险与影响

- 风险：

1. 协议兼容性风险：在 protocol.py 中新增 defer_loading 字段和序列化逻辑可能影响现有 API 客户端，特别是 strict 字段的默认值处理需保持向后兼容。
2. Anthropic 集成错误：serving.py 中的 tool_reference 解锁逻辑若错误使用字段名，会导致延迟工具无法解锁，影响功能可用性。
3. 测试覆盖不足：测试文件虽新增，但未覆盖 Anthropic 端点的过滤和解锁场景，可能引入回归问题。
4. 设计混淆风险：defer_loading 为 GLM 特定扩展，与 OpenAI API 标准不一致，可能导致用户误解或集成困难。 - 影响：对用户：GLM 模型用户现在可以利用 defer_loading 功能实现工具的动态隐藏和通过 tool_reference 解锁，增强多步工具调用的灵活性。对系统：扩展了 Chat Completions API 协议，新增字段和内容类型，影响 OpenAI 和 Anthropic 端点的请求处理流程。对团队：需要维护新的协议逻辑和确保与下游聊天模板（如 GLM-5.1）的集成，增加了代码复杂性和测试负担。 - 风险标记：协议扩展风险，测试覆盖不足，Anthropic 集成错误

关联脉络

- 暂无明显关联 PR