# PR #39835 完整报告

vllm-project/vllm

[ROCm][P/D][MORI][BugFix] Ensure correct api is used when making requests to prefill / decode nodes

合并时间：2026-04-22 08:48

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39835

## 执行摘要

- 一句话：修复 MORI IO KV 连接器 API 路由错误，确保预填充和解码节点使用正确端点。
- 推荐动作：该 PR 值得精读，重点关注代理服务器的路由重构设计，以及如何通过参数化 API 路径来避免硬编码，这对于构建灵活的服务端点有借鉴意义。

## 功能与动机

根据 PR 描述，当前 MORI IO KV 连接器已损坏，因为请求 URL 错误地使用了 /v1/completions 后缀，而正确应为 /v1。此外，处理器未区分 /v1/completions 和 /v1/chat/completions 路由，导致问题。此 PR 旨在修复这些问题，确保连接器能正确工作，并通过 lm_eval 测试验证。

## 实现拆解

1. 重构代理服务器路由处理：在 examples/online_serving/disaggregated_serving/moriio_toy_proxy_server.py 中，将原单一 handle_request 函数拆分为 handle_completions_request 和 handle_chat_completions_request 两个独立路由函数，它们调用新的 handle_request 函数并传入 API 路径参数（/completions 或 /chat/completions）。
2. 动态构建请求端点 URL：在 handle_request 函数中，使用传入的 api 参数动态拼接预填充和解码实例的端点 URL（例如 prefill_request_url = prefill_instance_endpoint["request_address"] + api），确保使用正确的 API 路径。
3. 更新 KV 连接器注册地址：在 vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py 的 _ping 方法中，将 http_request_address 从 f"http://{self.request_address}/v1/completions" 改为 f"http://{self.request_address}/v1"，以匹配代理服务器期望的基地址。
4. 改进错误处理和修复语法错误：在代理服务器中，增强 send_request_to_prefill 和 stream_decode_response 函数的错误消息，包含更多响应详情；并修复了 review 中发现的语法错误（多余括号）。

关键文件：

- examples/online_serving/disaggregated_serving/moriio_toy_proxy_server.py（模块 代理服务器；类别 source；类型 core-logic；符号 handle_completions_request,

handle_request, handle_chat_completions_request）：代理服务器的主逻辑文件，重构了路由处理以支持正确的 API 端点，是关键变更所在。

- vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py（模块 KV 连接器；类别 source；类型 configuration）：KV 连接器实现，更新了注册时的请求地址，确保与代理服务器匹配。

关键符号：handle_completions_request, handle_request, handle_chat_completions_request, _ping

## 关键源码片段

### examples/online_serving/disaggregated_serving/moriio_toy_proxy_server.py

代理服务器的主逻辑文件，重构了路由处理以支持正确的 API 端点，是关键变更所在。

```
@app.route("/v1/completions", methods=["POST"])
async def handle_completions_request():
    # 处理标准补全请求，调用通用处理器并传入 API 路径
    return await handle_request("/completions", request)

@app.route("/v1/chat/completions", methods=["POST"])
async def handle_chat_completions_request():
    # 处理聊天补全请求，区分路由以避免混淆
    return await handle_request("/chat/completions", request)

async def handle_request(api: str, request: Request):
    # 通用请求处理器，接收 API 路径参数
    try:
        # ... 原有逻辑 ...
        # 动态构建预填充请求 URL，避免硬编码
        prefill_request_url = prefill_instance_endpoint["request_address"] + api
        send_prefill_task = asyncio.create_task(
            send_request_to_prefill(
                prefill_request_url, # 使用拼接后的 URL
                req_data_to_prefill,
                request_id,
                decode_instance_endpoint,
                selected_prefill_dp_rank,
            )
        )
        # 类似地处理解码请求 URL
        decode_request_url = decode_instance_endpoint["request_address"] + api
        decode_request_task = asyncio.create_task(
            start_decode_request(decode_request_url, req_data, request_id)
        )
        # ... 其余逻辑 ...
    except Exception as e:
        # 错误处理
        return await make_response(...)
```

### vllm/distributed/kv_transfer/kv_connector/v1/moriio/moriio_connector.py

KV 连接器实现，更新了注册时的请求地址，确保与代理服务器匹配。

```
def _ping(self, zmq_context):
    # 构建 HTTP 请求地址，用于向代理服务器注册实例
    # 修复前：http_request_address = f"http://{self.request_address}/v1/completions"
    http_request_address = f"http://{self.request_address}/v1" # 移除 /completions
    后缀，使用基地址
    role = "P" if self.is_producer else "D"
    retry_count = 0
    index = 1
    with zmq_context.socket(zmq.DEALER) as sock:
        sock.connect(f"tcp://{self.proxy_ip}:{self.proxy_ping_port}")
        while True:
            try:
                data = {
                    "type": "register",
                    "role": role,
                    "index": str(index),
                    "request_address": http_request_address, # 使用修正后的地址
                    "handshake_port": self.handshake_port,
                    "notify_port": self.notify_port,
                    "dp_size": self.moriio_config.dp_size,
                    "tp_size": self.moriio_config.tp_size,
                    "transfer_mode": self.mode.name,
                }
                # ... 发送注册数据 ...
            except Exception as e:
                # 错误处理
                pass
```

## 评论区精华

review 中主要讨论了代码正确性：

- gemini-code-assist[bot] 指出语法错误：在 moriio_toy_proxy_server.py 第 135 行有一个多余的闭合括号，会导致 SyntaxError。
- rasmith 确认修复：作者回复“Fixed”，表示已修复该问题。
- 其他 reviewer 批准：gshtras 和 inkcherry 均批准了 PR，未提出进一步争议。
- 语法错误修复 (correctness): 作者 rasmith 确认修复了该问题。

## 风险与影响

- 风险：技术风险较低：
  - 回归风险：变更集中在 MORI IO 连接器的代理服务器和注册逻辑，属于特定功能模块，但若其他组件依赖原 URL 格式，可能引入兼容性问题。
  - 性能风险：无显著性能影响，主要是路由和 URL 构建逻辑调整。

    - 安全风险：无新增安全漏洞，错误信息改进可能泄露内部 URL 细节，但属于调试信息，风险可控。
    - 兼容性风险：连接器注册地址变更可能影响已部署实例，需确保所有相关组件同步更新。
- 影响：影响范围有限但关键：
    - 用户影响：修复后，使用 MORI IO KV 连接器的用户将能正常进行预填充和解码请求，避免因 API 路由错误导致的失败。
    - 系统影响：仅影响分布式 KV 传输中的 MORI IO 连接器模块，不涉及核心推理路径。
    - 团队影响：提供了更清晰的错误信息和模块化路由，便于未来维护和扩展。
    - 风险标记：API 路由变更，缺少测试覆盖

# 关联脉络

- PR #34907 [ROCm][P/D][MORI] Add MORI IO KV connector: PR body 中引用了该 PR 的 Justfile 和测试指令，表明当前 PR 是基于早期 MORI IO 连接器实现的修复。