



(12) 发明专利

(10) 授权公告号 CN 116708552 B

(45) 授权公告日 2024.06.18

(21) 申请号 202211215417.0

(22) 申请日 2022.09.30

(65) 同一申请的已公布的文献号
申请公布号 CN 116708552 A

(43) 申请公布日 2023.09.05

(73) 专利权人 腾讯云计算(北京)有限责任公司
地址 100190 北京市海淀区知春路49号3层
西部309

专利权人 广州地铁集团有限公司

(72) 发明人 陈志平 黄朝晖 艾义 丁建隆
刘智成 蔡昌俊 祝唯 熊帅
方特 彭伟泽 叶红霞 温辛妍
肖莹

(74) 专利代理机构 广州三环专利商标代理有限公司 44202

专利代理师 杜维

(51) Int.Cl.

H04L 67/56 (2022.01)

H04L 67/63 (2022.01)

H04L 67/51 (2022.01)

(56) 对比文件

CN 111831269 A, 2020.10.27

CN 113992738 A, 2022.01.28

审查员 曹荣珍

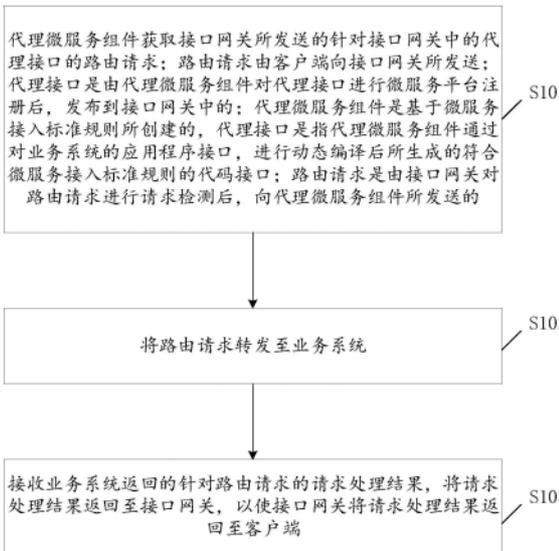
权利要求书4页 说明书24页 附图9页

(54) 发明名称

一种地铁微服务注册方法、装置、设备及可读存储介质

(57) 摘要

本申请公开了一种地铁微服务注册方法、装置、设备及可读存储介质,方法包括:代理微服务组件获取接口网关所发送的针对接口网关中的代理接口的路由请求;代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;将路由请求转发至业务系统;接收业务系统返回的针对路由请求的请求处理结果,将请求处理结果返回至接口网关,以使接口网关将请求处理结果返回至客户端。采用本申请,可以提高微服务的业务适配度,进而提高微服务的利用率。本申请可应用于云技术、人工智能、智慧交通、辅助驾驶等各种场景。



1. 一种地铁微服务注册方法,其特征在于,包括:

代理微服务组件获取接口网关所发送的针对所述接口网关中的代理接口的路由请求;所述路由请求由客户端向所述接口网关所发送;所述代理接口是由所述代理微服务组件对所述代理接口进行微服务平台注册后,发布到所述接口网关中的;所述对所述代理接口进行微服务平台注册是指将所述代理接口在微服务平台中进行注册,所述代理微服务组件是基于微服务接入标准规则所创建的,所述代理接口是对代理请求代码进行动态编译后所得到的,所述代理请求代码由所述代理微服务组件基于接口配置信息中的传入参数所生成,所述接口配置信息是指业务系统的应用程序接口的配置信息;所述路由请求是由所述接口网关对所述路由请求进行请求检测后,向所述代理微服务组件所发送的;

将所述路由请求转发至所述业务系统;

接收所述业务系统返回的针对所述路由请求的请求处理结果,将所述请求处理结果返回至所述接口网关,以使所述接口网关将所述请求处理结果返回至所述客户端。

2. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

从所述微服务平台对应的程序接口管理平台中,拉取所述应用程序接口的接口配置信息;

在所述代理微服务组件中,基于所述接口配置信息,生成针对所述应用程序接口的所述代理接口;

将所述代理接口在所述微服务平台中进行注册;

当在所述微服务平台中完成对所述代理接口的注册时,将所述代理接口发布于所述接口网关中。

3. 根据权利要求2所述的方法,其特征在于,所述接口配置信息包括所述应用程序接口的传入参数;

所述在所述代理微服务组件中,基于所述接口配置信息,生成针对所述应用程序接口的所述代理接口,包括:

在所述代理微服务组件中,基于所述传入参数,生成针对所述应用程序接口的代理请求代码;

将所述代理请求代码传入至虚拟机中,通过所述虚拟机中的动态编译工具对所述代理请求代码进行动态编译,得到动态编译字节流;

将所述动态编译字节流加载至所述虚拟机中,得到动态接口代码,将所述动态接口代码确定为所述代理接口。

4. 根据权利要求2所述的方法,其特征在于,所述方法还包括:

接收所述程序接口管理平台发送的针对所述应用程序接口的接口更新通知;

基于所述接口更新通知,从所述程序接口管理平台中,拉取所述应用程序接口的更新接口配置信息;

在所述代理微服务组件中,基于所述更新接口配置信息,生成针对所述应用程序接口的更新代理接口;

将所述更新代理接口在所述微服务平台中进行注册;

当在所述微服务平台中完成对所述更新代理接口的注册时,将所述更新代理接口发布于所述接口网关中。

5. 根据权利要求4所述的方法,其特征在于,所述方法还包括:

生成针对所述代理接口的接口卸载通知,将所述接口卸载通知发送至所述接口网关,以使所述接口网关基于所述接口卸载通知将所述接口网关中的代理接口进行卸载;

将所述代理接口从所述微服务平台中进行注销;

当在所述微服务平台中完成对所述代理接口的注销,且在接收到所述接口网关发送的针对所述代理接口的卸载完成通知消息时,执行所述在所述代理微服务组件中,基于所述更新接口配置信息,生成针对所述应用程序接口的更新代理接口的步骤。

6. 根据权利要求2所述的方法,其特征在于,所述接口配置信息包括所述应用程序接口的接口限流阈值以及接口有效时长参数;所述路由请求是由所述接口网关基于所述接口限流阈值与所述接口有效时长参数,对所述路由请求进行检测得到检测结果,且在所述检测结果为检测通过结果时,向所述代理微服务组件所发送的。

7. 根据权利要求1所述的方法,其特征在于,所述对所述代理接口进行微服务平台注册是指将所述代理接口在微服务平台中进行注册;当在所述微服务平台中完成对所述代理接口的注册时,所述微服务平台为所述业务系统提供微服务;

所述方法还包括:

基于所述代理接口,接收所述微服务平台发送的针对所述业务系统的异常处理提示信息;所述异常处理提示信息包括针对所述业务系统的异常对象,以及所述异常对象在所述业务系统中所处的异常位置;所述异常对象是指所述微服务平台基于微服务中的全链路检测服务,对所述业务系统进行全链路实时检测所检测到的发生异常的对象;

将所述异常处理提示信息发送至所述业务系统,以使所述业务系统在所述异常位置处获取到所述异常对象,且对所述异常对象进行修正处理。

8. 根据权利要求1所述的方法,其特征在于,所述对所述代理接口进行微服务平台注册是指将所述代理接口在微服务平台中进行注册;当在所述微服务平台中完成对所述代理接口的注册时,所述微服务平台为所述业务系统提供微服务;

所述方法还包括:

基于所述代理接口,接收所述微服务平台发送的针对所述业务系统的设备性能信息;所述设备性能信息是由所述微服务平台基于微服务中的全链路性能分析服务,在第一目标时间段内对业务系统中的系统设备进行全链路实时性能分析所得到的;所述设备性能信息包括所述业务系统中所述系统设备的实时响应时长以及设备错误频率;

根据所述实时响应时长以及所述设备错误频率,确定所述系统设备的设备状态;

若所述系统设备的设备状态为异常运行状态,则生成针对所述系统设备的暂停运行提示信息;

将所述暂停运行提示信息发送至所述业务系统,以使所述业务系统在第二目标时间段内暂停运行所述系统设备;所述第二目标时间段晚于所述第一目标时间段。

9. 根据权利要求8所述的方法,其特征在于,所述根据所述实时响应时长以及所述设备错误频率,确定所述系统设备的设备状态,包括:

将所述实时响应时长与响应时长阈值进行比较,将所述设备错误频率与错误频率阈值进行比较;

若所述实时响应时长小于所述响应时长阈值,且所述设备错误频率小于所述错误频率

阈值,则将所述系统的设备状态确定为正常运行状态;

若所述实时响应时长小于所述响应时长阈值,且所述设备错误频率大于所述错误频率阈值,则将所述系统的设备状态确定为异常运行状态;

若所述实时响应时长大于所述响应时长阈值,且所述设备错误频率小于所述错误频率阈值,则将所述系统的设备状态确定为异常运行状态;

若所述实时响应时长大于所述响应时长阈值,且所述设备错误频率大于所述错误频率阈值,则将所述系统的设备状态确定为异常运行状态。

10. 根据权利要求1所述的方法,其特征在于,所述代理接口的接口属性为灰度测试属性,所述代理接口的灰度发布对象的数量为至少两个,每个灰度发布对象是指具备所述代理接口的请求权限的对象,至少两个灰度发布对象包括所述客户端;

所述方法还包括:

接收至少两个灰度发布对象分别针对所述代理接口的使用反馈信息;

基于至少两个使用反馈信息,确定所述代理接口的接口发布状态;

若所述接口发布状态为合格发布状态,则将所述代理接口的接口属性从所述灰度测试属性切换为全量发布属性,且将所述代理接口发布至全量发布对象;被发布所述代理接口的全量发布对象具备所述代理接口的请求权限;

若所述接口发布状态为待定发布状态,则将所述代理接口的接口属性从所述灰度测试属性切换为发布待定属性,且将所述代理接口进行微服务平台注销,向所述接口网关发送针对所述代理接口的接口卸载通知,以使所述接口网关基于所述接口卸载通知将所述接口网关中的所述代理接口进行卸载。

11. 根据权利要求10所述的方法,其特征在于,所述基于至少两个使用反馈信息,确定所述代理接口的接口发布状态,包括:

将所述至少两个使用反馈信息中的为正向反馈信息的使用反馈信息,确定为待统计反馈信息;

统计所述待统计反馈信息的正向反馈数量,以及所述至少两个使用反馈信息的反馈总数量;

确定所述正向反馈数量与所述反馈总数量之间的数量比值;

若所述数量比值大于比值阈值,则将所述接口发布状态确定为合格发布状态;

若所述数量比值小于所述比值阈值,则将所述接口发布状态确定为待定发布状态。

12. 一种地铁微服务注册装置,其特征在于,包括:

请求获取模块,用于获取接口网关所发送的针对所述接口网关中的代理接口的路由请求;所述路由请求由客户端向所述接口网关所发送;所述代理接口是由代理微服务组件对所述代理接口进行微服务平台注册后,发布到所述接口网关中的;所述对所述代理接口进行微服务平台注册是指将所述代理接口在微服务平台中进行注册,所述代理微服务组件是基于微服务接入标准规则所创建的,所述代理接口是对代理请求代码进行动态编译后所得到的,所述代理请求代码由所述代理微服务组件基于接口配置信息中的传入参数所生成,所述接口配置信息是指业务系统的应用程序接口的配置信息;所述路由请求是由所述接口网关对所述路由请求进行请求检测后,向所述代理微服务组件所发送的;

请求转发模块,用于将所述路由请求转发至所述业务系统;

结果转发模块,用于接收所述业务系统返回的针对所述路由请求的请求处理结果,将所述请求处理结果返回至所述接口网关,以使所述接口网关将所述请求处理结果返回至所述客户端。

13. 一种计算机设备,其特征在于,包括:处理器、存储器以及网络接口;

所述处理器与所述存储器、所述网络接口相连,其中,所述网络接口用于提供网络通信功能,所述存储器用于存储计算机程序,所述处理器用于调用所述计算机程序,以使所述计算机设备执行权利要求1-11任一项所述的方法。

14. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质中存储有计算机程序,所述计算机程序适于由处理器加载并执行权利要求1-11任一项所述的方法。

15. 一种计算机程序产品,其特征在于,所述计算机程序产品包括计算机程序,所述计算机程序存储在计算机可读存储介质中,所述计算机程序适于由处理器读取并执行,以使得具有所述处理器的计算机设备执行权利要求1-11任一项所述的方法。

一种地铁微服务注册方法、装置、设备及可读存储介质

技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种地铁微服务注册方法、装置、设备及可读存储介质。

背景技术

[0002] 微服务平台是一个围绕着应用和微服务的平台,其提供应用全生命周期管理、数据化运营、立体化检测和服务治理等功能。

[0003] 目前,较多业务场景下的第三方专业系统(如地铁轨交业务中的电力专业系统、检修专业系统、电梯专业系统等),与微服务平台不适配,这些第三方专业系统无法部署在微服务平台中,无法有效利用到微服务。也就是说,在部分业务中,微服务的适配性并不高,利用率较低。

发明内容

[0004] 本申请实施例提供一种地铁微服务注册方法、装置、设备及可读存储介质,可以提高微服务的业务适配度,进而提高微服务的利用率。

[0005] 本申请实施例一方面提供了一种地铁微服务注册方法,方法包括:

[0006] 代理微服务组件获取接口网关所发送的针对接口网关中的代理接口的路由请求;路由请求由客户端向接口网关所发送;代理接口是由代理微服务组件对代理接口进行微服务平台注册后,发布到接口网关中的;代理微服务组件是基于微服务接入标准规则所创建的,代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;路由请求是由接口网关对路由请求进行请求检测后,向代理微服务组件所发送的;

[0007] 将路由请求转发至业务系统;

[0008] 接收业务系统返回的针对路由请求的请求处理结果,将请求处理结果返回至接口网关,以使接口网关将请求处理结果返回至客户端。

[0009] 本申请实施例一方面提供了一种地铁微服务注册装置,地铁微服务注册装置包括:

[0010] 请求获取模块,用于获取接口网关所发送的针对接口网关中的代理接口的路由请求;路由请求由客户端向接口网关所发送;代理接口是由代理微服务组件对代理接口进行微服务平台注册后,发布到接口网关中的;代理微服务组件是基于微服务接入标准规则所创建的,代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;路由请求是由接口网关对路由请求进行请求检测后,向代理微服务组件所发送的;

[0011] 请求转发模块,用于将路由请求转发至业务系统;

[0012] 结果转发模块,用于接收业务系统返回的针对路由请求的请求处理结果,将请求处理结果返回至接口网关,以使接口网关将请求处理结果返回至客户端。

- [0013] 在一个实施例中,对代理接口进行微服务平台注册是指将代理接口在微服务平台中进行注册;
- [0014] 地铁微服务注册装置还可以包括:
- [0015] 配置信息拉取模块,用于从微服务平台对应的程序接口管理平台中,拉取应用程序接口的接口配置信息;
- [0016] 接口生成模块,用于在代理微服务组件中,基于接口配置信息,生成针对应用程序接口的代理接口;
- [0017] 接口注册模块,用于将代理接口在微服务平台中进行注册;
- [0018] 接口发布模块,用于当在微服务平台中完成对代理接口的注册时,将代理接口发布于接口网关中。
- [0019] 在一个实施例中,接口配置信息包括应用程序接口的传入参数;
- [0020] 接口生成模块可以包括:
- [0021] 请求代码生成单元,用于在代理微服务组件中,基于传入参数,生成针对应用程序接口的代理请求代码;
- [0022] 代码编译单元,用于将代理请求代码传入至虚拟机中,通过虚拟机中的动态编译工具对代理请求代码进行动态编译,得到动态编译字节流;
- [0023] 代码加载单元,用于将动态编译字节流加载至虚拟机中,得到动态接口代码,将动态接口代码确定为代理接口。
- [0024] 在一个实施例中,地铁微服务注册装置还可以包括:
- [0025] 更新信息拉取模块,用于接收程序接口管理平台发送的针对应用程序接口的接口更新通知;
- [0026] 更新信息拉取模块,还用于基于接口更新通知,从程序接口管理平台中,拉取应用程序接口的更新接口配置信息;
- [0027] 更新接口生成模块,用于在代理微服务组件中,基于更新接口配置信息,生成针对应用程序接口的更新代理接口;
- [0028] 更新接口注册模块,用于将更新代理接口在微服务平台中进行注册;
- [0029] 更新接口发布模块,用于当在微服务平台中完成对更新代理接口的注册时,将更新代理接口发布于接口网关中。
- [0030] 在一个实施例中,地铁微服务注册装置还可以包括:
- [0031] 接口卸载模块,用于生成针对代理接口的接口卸载通知,将接口卸载通知发送至接口网关,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载;
- [0032] 接口注销模块,用于将代理接口从微服务平台中进行注销;
- [0033] 步骤执行模块,用于当在微服务平台中完成对代理接口的注销,且在接收到接口网关发送的针对代理接口的卸载完成通知消息时,执行在代理微服务组件中,基于更新接口配置信息,生成针对应用程序接口的更新代理接口的步骤。
- [0034] 在一个实施例中,接口配置信息包括应用程序接口的接口限流阈值以及接口有效时长参数;路由请求是由接口网关基于接口限流阈值与接口有效时长参数,对路由请求进行检测得到检测结果,且在检测结果为检测通过结果时,向代理微服务组件所发送的。
- [0035] 在一个实施例中,对代理接口进行微服务平台注册是指将代理接口在微服务平台

中进行注册；当在微服务平台中完成对代理接口的注册时，微服务平台为业务系统提供微服务；

[0036] 地铁微服务注册装置还可以包括：

[0037] 系统检测模块，用于基于代理接口，接收微服务平台发送的针对业务系统的异常处理提示信息；异常处理提示信息包括针对业务系统的异常对象，以及异常对象在业务系统中所处的异常位置；异常对象是指微服务平台基于微服务中的全链路检测服务，对业务系统进行全链路实时检测所检测到的发生异常的对象；

[0038] 系统检测模块，还用于将异常处理提示信息发送至业务系统，以使业务系统在异常位置处获取到异常对象，且对异常对象进行修正处理。

[0039] 在一个实施例中，对代理接口进行微服务平台注册是指将代理接口在微服务平台中进行注册；当在微服务平台中完成对代理接口的注册时，微服务平台为业务系统提供微服务；

[0040] 地铁微服务注册装置还可以包括：

[0041] 系统分析模块，用于基于代理接口，接收微服务平台发送的针对业务系统的设备性能信息；设备性能信息是由微服务平台基于微服务中的全链路性能分析服务，在第一目标时间段内对业务系统中的系统设备进行全链路实时性能分析所得到的；设备性能信息包括业务系统中系统设备的实时响应时长以及设备错误频率；

[0042] 设备状态确定模块，用于根据实时响应时长以及设备错误频率，确定系统设备的设备状态；

[0043] 暂停提示模块，用于若系统设备的设备状态为异常运行状态，则生成针对系统设备的暂停运行提示信息；

[0044] 暂停提示模块，还用于将暂停运行提示信息发送至业务系统，以使业务系统在第二目标时间段内暂停运行系统设备；第二目标时间段晚于第一目标时间段。

[0045] 在一个实施例中，设备状态确定模块可以包括：

[0046] 时长比较单元，用于将实时响应时长与响应时长阈值进行比较，将设备错误频率与错误频率阈值进行比较；

[0047] 状态确定单元，用于若实时响应时长小于响应时长阈值，且设备错误频率小于错误频率阈值，则将系统的设备状态确定为正常运行状态；

[0048] 状态确定单元，还用于若实时响应时长小于响应时长阈值，且设备错误频率大于错误频率阈值，则将系统的设备状态确定为异常运行状态；

[0049] 状态确定单元，还用于若实时响应时长大于响应时长阈值，且设备错误频率小于错误频率阈值，则将系统的设备状态确定为异常运行状态；

[0050] 状态确定单元，还用于若实时响应时长大于响应时长阈值，且设备错误频率大于错误频率阈值，则将系统的设备状态确定为异常运行状态。

[0051] 在一个实施例中，代理接口的接口属性为灰度测试属性，代理接口的灰度发布对象的数量为至少两个，每个灰度发布对象是指具备代理接口的请求权限的对象，至少两个灰度发布对象包括客户端；

[0052] 地铁微服务注册装置还可以包括：

[0053] 反馈接收模块，用于接收至少两个灰度发布对象分别针对代理接口的使用反馈信

息；

[0054] 接口状态确定模块,用于基于至少两个使用反馈信息,确定代理接口的接口发布状态;

[0055] 接口处理模块,用于若接口发布状态为合格发布状态,则将代理接口的接口属性从灰度测试属性切换为全量发布属性,且将代理接口发布至全量发布对象;被发布代理接口的全量发布对象具备代理接口的请求权限;

[0056] 接口处理模块,还用于若接口发布状态为待定发布状态,则将代理接口的接口属性从灰度测试属性切换为发布待定属性,且将代理接口进行微服务平台注销,向接口网关发送针对代理接口的接口卸载通知,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载。

[0057] 在一个实施例中,接口状态确定模块可以包括:

[0058] 反馈统计单元,用于将至少两个使用反馈信息中的为正向反馈信息的使用反馈信息,确定为待统计反馈信息;

[0059] 反馈统计单元,还用于统计待统计反馈信息的正向反馈数量,以及至少两个使用反馈信息的反馈总数量;

[0060] 反馈统计单元,还用于确定正向反馈数量与反馈总数量之间的数量比值;

[0061] 发布状态确定单元,用于若数量比值大于比值阈值,则将接口发布状态确定为合格发布状态;

[0062] 发布状态确定单元,还用于若数量比值小于比值阈值,则将接口发布状态确定为待定发布状态。

[0063] 本申请实施例一方面提供了一种计算机设备,包括:处理器和存储器;

[0064] 存储器存储有计算机程序,计算机程序被处理器执行时,使得处理器执行本申请实施例中的方法。

[0065] 本申请实施例一方面提供了一种计算机可读存储介质,计算机可读存储介质存储有计算机程序,计算机程序包括程序指令,程序指令当被处理器执行时,执行本申请实施例中的方法。

[0066] 本申请的一个方面,提供了一种计算机程序产品,该计算机程序产品包括计算机程序,该计算机程序存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机程序,处理器执行该计算机程序,使得该计算机设备执行本申请实施例中一方面提供的方法。

[0067] 在本申请实施例中,可以基于微服务接入标准规则,创建一个代理微服务组件,该代理微服务组件可以对业务系统的应用程序接口进行动态代理形成代理接口,将该代理接口进行微服务平台注册,且将代理接口发布到接口网关中后,该业务系统即可有效利用到微服务(如,接口网关提供的服务等)。例如,接口网关提供的服务可为对代理接口的路由请求进行检测的服务,当某个客户端期望访问业务系统时,则该客户端可以向接口网关发送针对该代理接口的路由请求;而接口网关可对路由请求进行请求检测,并在检测后将该路由请求发送至代理微服务组件;随后,代理微服务组件可以将该路由请求转发至业务系统,而业务系统在接收到路由请求后即可进行请求处理得到请求处理结果。应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接

口进行动态编译,在将代理接口进行微服务注册并注册到接口网关后,可以使得业务系统能够有效利用到微服务。通过代理微服务组件,可以支持不能有效利用微服务的业务系统(如地铁轨交业务中的检修专业系统、电力专业系统等),很好地利用到微服务,由此可以很好地提升微服务的适配度,从而提高微服务的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

附图说明

[0068] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0069] 图1是本申请实施例提供的一种网络架构图;

[0070] 图2是本申请实施例提供的一种应用场景示意图;

[0071] 图3是本申请实施例提供的一种地铁微服务注册方法的流程示意图;

[0072] 图4是本申请实施例提供的一种系统架构图;

[0073] 图5是本申请实施例提供的一种对代理接口进行微服务注册的流程示意图;

[0074] 图6是本申请实施例提供的一种对更新代理接口进行微服务注册的流程示意图;

[0075] 图7是本申请实施例提供的一种系统交互流程图;

[0076] 图8是本申请实施例提供的一种地铁微服务注册装置的结构示意图;

[0077] 图9是本申请实施例提供的一种计算机设备的结构示意图。

具体实施方式

[0078] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范围。

[0079] 本申请涉及微服务以及相关概念,为便于理解,以下将优先对微服务以及相关概念进行阐述。

[0080] 1、微服务:微服务是指开发多个独立小型、但有完整业务功能的服务,例如系统拆分成的很多小应用。其中,每个服务都有自己的处理和通讯机制,可以部署在不同的服务器上,也可以部署在相同的服务器不同的容器上。微服务需要运行在微服务框架下,如spring cloud(春云,是一系列框架的有序集合)、K8S+Istio(Kubernetes+Istio,开源微服务框架)等,因而spring cloud、K8S+Istio等都可以属于本申请中的微服务运行环境。微服务可以理解为是一种云原生架构方法,其中单个应用程序由许多松散耦合且可独立部署的较小组件或服务组成。这些服务通常有自己的堆栈,包括数据库和数据模型等;这些服务可以通过REST API、事件流和消息代理的组合进行相互通信;这些服务是按业务能力组织的,分隔服务的线通常称为有界上下文。

[0081] 2、微服务平台:微服务平台是一个围绕着应用和微服务的PaaS平台,提供应用全生命周期管理、数据化运营、立体化检测和服务治理等功能。TSF拥抱Spring Cloud、

Service Mesh微服务框架,帮助企业或其他对象解决传统集中式架构转型的困难,打造大规模高可用的分布式系统架构,实现业务、产品的快速落地。

[0082] 3、API (Application Programming Interface) 网关 (API的中文名为应用程序编程的接口,可简称为应用程序接口,其相当于一个网络平台的身份地址,目的是为了供给应用程序与开发人员根据某个软件或某个硬件才得以访问一组例程的能力):API网关是一个服务器,是业务系统的唯一入口。从面向对象设计的角度看,它与外观模式类似。API网关封装了系统内部架构,为每个客户端提供一个定制的API。它可能还具有其他职责,如身份验证、负载均衡、缓存、请求分片与管理、静态响应处理。本申请实施例中的API网关可以为微服务网关。微服务网关主要是针对微服务运行环境而言,有微服务API网关时,该网关主要用于客户端或消费端通过统一入口接入微服务。

[0083] 4、组件:组件把应用系统需要用到的算法、数据、流程、通用、可视化、物联等能力,以标准定义封装为组件并沉淀下来。

[0084] 5、API管理平台:API管理平台主要用于定义API接口的传入参数(可简称为入参)、传出参数(可简称为出参)以及鉴权方式等API的相关信息。

[0085] 6、云技术(Cloud technology):云技术是指在广域网或局域网内将硬件、软件、网络等系列资源统一起来,实现数据的计算、储存、处理和共享的一种托管技术。

[0086] 云技术是基于云计算商业模式应用的网络技术、信息技术、整合技术、管理平台技术、应用技术等的总称,可以组成资源池,按需所用,灵活便利。云计算技术将变成重要支撑。技术网络系统的后台服务需要大量的计算、存储资源,如视频网站、图片类网站和更多的门户网站。伴随着互联网行业的高度发展和应用,将来每个物品都有可能存在自己的识别标志,都需要传输到后台系统进行逻辑处理,不同程度级别的数据将会分开处理,各类行业数据皆需要强大的系统后盾支撑,只能通过云计算来实现。

[0087] 为便于理解,下面将对本申请实施例的设计思想进行简要介绍:

[0088] 目前,对于微服务平台而言,很多场景下的第三方业务系统(如城规行业场景下的电力专业系统、检修专业系统等等),由于不适配或其他原因,无法迁移部署在微服务平台上,导致这些第三方业务系统无法通过微服务平台所提供的微服务功能进行系统管理。

[0089] 鉴于此,本申请实施例提供一种地铁微服务注册方法,实际上可以理解为一种提高微服务平台的适配能力的数据处理方法,通过本申请所提供的方法,可以提高微服务平台的适配度,使得如地铁轨交等场景中的不适配微服务平台的第三方业务系统,也能够有效利用到微服务平台所提供的微服务功能。具体的,本申请可以基于微服务平台的接入标准规则(实际上,该接入标准规则可以是指微服务平台的开发标准规范,可简称为微服务接入标准规则)创建一个代理微服务组件,对于某个不适配微服务平台的第三方业务系统(可以理解为不支持微服务改造的第三方业务系统,以下将简称为业务系统)的API接口(也就是应用程序接口),可以通过代理微服务组件对其进行动态编译,将应用程序接口转换为代码接口(可称之为代理接口),由于代理微服务组件是基于微服务平台的接入标准规则所创建的,那么通过该代理微服务组件所动态编译得到的代理接口,也是符合微服务平台的接入标准规则的,那么该代理接口可以注册到微服务平台中(即可以进行微服务注册),通过注册到微服务平台中,可以使得该代理接口所对应的业务系统,能够有效利用到微服务;同时,在对代理接口进行微服务注册后,也可以将该代理接口发布到微服务平台对应的接口

网关(也可称之为微服务网关)中,通过该接口网关,也可以对该代理接口进行管理,例如,可以对发布到接口网关中的应用程序接口(针对通过代理微服务组件进行动态代理的接口而言,接口网关中的接口实际上是代理接口)进行限流、熔断、审计、超时等服务处理。

[0090] 例如,当代理接口被发布于接口网关后,若有某个客户端发起针对该代理接口的路由请求,那么该接口网关可以接收到该路由请求,在转发该路由请求前,该接口网关可以基于该代理接口的检测策略(如限流策略)对该路由请求进行检测,在通过检测得到检测结果后,若该检测结果为检测通过结果,则该接口网关可以将该路由请求转发至代理微服务组件;该代理微服务组件可以将该路由请求转发至业务系统;随后,该业务系统可以对该路由请求进行请求处理,得到请求处理结果后,可以将该请求处理结果返回至代理微服务组件,而该代理微服务组件可以将该请求处理结果返回至接口网关,该接口网关再将该请求处理结果返回至该客户端。

[0091] 也就是说,本申请实施例中的代理微服务组件,主要用于对无法部署到微服务平台中的应用程序接口(API),进行动态代理,使其能够部署到微服务平台中,提高微服务平台的适配性,从而可以使得无法部署到微服务平台中的业务系统能够接入微服务平台,从而可以使得这些业务系统有效利用到微服务平台提供的微服务,由此可以提高微服务平台的利用率;同时,代理微服务组件也可以提供数据转发的功能(例如,提供请求转发、请求处理结果转发的功能)。

[0092] 以下将结合说明书附图对本申请的优选实施例进行说明,应当理解,此处所描述的优选实施例仅用于说明和解释本申请,并不用于限定本申请,并且在不冲突的情况下,本申请中的实施例以及实施例中的特征可以相互结合。

[0093] 为便于理解,请参见图1,图1是本申请实施例提供的一种网络架构图。如图1所示,该网络架构可以包括服务器1000、终端设备集群以及第三方业务系统。该终端设备集群可以包括一个或者多个终端设备。如图1所示,该终端设备集群具体可以包括终端设备100a、终端设备100b、终端设备100c、…、终端设备100n。如图1所示,终端设备100a、终端设备100b、终端设备100c、…、终端设备100n可以分别与上述服务器1000进行网络连接,以便于每个终端设备可以通过该网络连接与服务器1000进行数据交互。其中,这里的网络连接不限定连接方式,可以通过有线通信方式进行直接或间接地连接,也可以通过无线通信方式进行直接或间接地连接,还可以通过其他方式,本申请在此不做限制。

[0094] 其中,该终端设备集群中的每个终端设备均可以包括:智能手机、平板电脑、笔记本电脑、台式计算机、智能音箱、智能手表、车载终端、智能电视、智能车载等具有数据处理功能(如图像处理、视频处理等等)的智能终端。应当理解,如图1所示的终端设备集群中的每个终端设备均可以安装有目标应用(即应用客户端),当该应用客户端运行于各终端设备中时,可以分别与上述图1所示的服务器1000之间进行数据交互。其中,该应用客户端可以包括社交客户端、多媒体客户端(例如,视频客户端)、娱乐客户端(例如,游戏客户端)、教育客户端、直播客户端等应用客户端。其中,该应用客户端可以为独立的客户端,也可以为集成在某客户端(例如,社交客户端、教育客户端以及多媒体客户端等)中的嵌入式子客户端,在此不做限定。

[0095] 如图1所示,本申请实施例中的服务器1000可以为微服务平台设备(微服务平台设备可以是指微服务平台对应的设备,该微服务平台设备中可以集成有API网关(接口网关,

也可称之为微服务网关)、微服务平台、代理微服务组件、程序接口管理平台(API管理平台)等等),其可以负责为第三方业务系统提供微服务。该服务器1000可以是独立的物理服务器,也可以是多个物理服务器构成的服务器集群或者分布式系统,还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。其中,本申请实施例将不对终端设备和服务器的数量进行限制。当然,微服务平台设备中的某个组件(如API网关)也可以是独立的物理服务器。

[0096] 为便于理解,本申请实施例可以在图1所示的多个终端设备中选择一个终端设备作为目标终端设备。例如,本申请实施例可以将图1所示的终端设备100a作为目标终端设备,该目标终端设备中可以集成有目标应用(即应用客户端,如社交客户端)。此时,该目标终端设备可以通过该应用客户端对应的业务数据平台与服务器1000之间实现数据交互。以目标应用为社交客户端为例,当目标终端设备上运行社交客户端时,目标终端设备的应用数据(如访问请求)可以通过网络连接发送至服务器1000,而服务器1000可以将该应用数据转发至第三方业务系统,由该第三方业务系统对该应用数据进行相应数据处理。

[0097] 可以理解的是,本申请实施例提供的方法可以由计算机设备执行,计算机设备包括但不限于终端设备或服务器。其中,终端设备以及服务器可以通过有线或无线通信方式进行直接或间接地连接,本申请在此不做限制。

[0098] 需要说明的是,在本申请的具体实施方式中,涉及到用户信息、用户数据(如应用数据中所包含的用户数据等)等相关的数据,需要经过用户授权许可才能进行获取。也就是说,当本申请以上实施例运用到具体产品或技术中时,需要获得用户许可或者同意,且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。

[0099] 为便于理解,请参见图2,图2是本申请实施例提供的一种应用场景示意图。如图2所示,首先,本申请可以基于微服务平台的开发标准规范,创建对应的代理服务(也就是代理微服务组件),由于创建后的代理服务部署于微服务平台设备中,所以可以将其称之为代理微服务组件。其中,如图2所示,在创建代理微服务时,可以配置服务名称(如是为电梯专业系统创建的代理服务,则该服务名称可以配置为电梯专业)、部署类型(如图所示的中央类型)、服务类型(如图所示的组件服务)、业务类型(如图所示的非TSF服务)、资源访问(也就是访问的环境,即访问context,如图所示的elevator)、服务描述(如图所示的电梯专业服务)、环境名称(如图所示的dev)、服务连接(可以理解为服务Host)等信息,这些信息可以作为代理微服务组件的描述信息(基本信息)。

[0100] 进一步地,在创建完成电梯专业系统对应的代理微服务后,可以在该代理微服务下配置电梯专业系统的应用程序接口(API接口)的信息。例如,可以配置应用程序接口的名称(如接口名称,如图所示的电梯状态查询接口)、接口路径、请求方式(如图所示的GET方式)、域名类型(如图所示的HTTP)等信息。应当理解,该应用程序接口的信息可以用于应用方查看,以便于应用方基于这些信息进行查看。当然,该应用程序接口的信息也可以用于代理微服务,生成相应的代理接口。

[0101] 进一步地,在配置完成应用程序接口的信息(可称之为API信息)后,代理微服务可以生成该应用程序接口的代理接口,对于该代理接口可以发布到对应的网关(接口网关)中。其中,由于该代理接口实际上为该应用程序接口的代码接口,那么发布该代理接口实际

上也就是发布该应用程序接口,可以将该代理接口发布到对应的网关中。例如,如图所示,在选择网关名称以及网关分组后,即可将该代理接口发布到对应的网关中。

[0102] 可以理解的是,在发布代理接口到对应的网关中后,可以对应用程序接口配置相应的处理策略。例如,可以配置限流策略,即可以配置该应用程序接口的最大请求次数(可称之为接口限流阈值),那么对于该应用程序接口的路由请求的请求频率(请求次数),不能超过该最大请求次数。而接口网关可以基于该最大请求次数对客户端发送的路由请求进行检测。

[0103] 应当理解,本申请实施例通过创建代理微服务,可以通过代理微服务对不能部署到微服务品台中的应用程序接口进行动态代理,将代理的代理接口注册到接口网关中,客户端请求的时候可以通过接口网关的策略对其进行检测,并可以通过代理微服务转发到业务系统,将代理微服务部署到微服务平台中,可以使得业务系统能够有效利用到微服务。

[0104] 进一步地,请参见图3,图3是本申请实施例提供的一种地铁微服务注册方法的流程示意图。本申请实施例所提供的方法可应用于各种场景,包括但不限于音视频、云技术、人工智能、智慧交通、辅助驾驶等。如图3所示,该方法可以由终端设备(例如,上述图1的终端设备)执行,也可以由服务器(例如,上述图1的服务器)执行,还可以由终端设备和服务器共同执行。其中,该计算机设备可以是指部署有代理微服务组件的微服务平台设备,该方法具体可以由微服务平台设备中的代理微服务组件执行。为便于理解,本实施例以该方法由上述代理微服务组件执行为例进行说明,以阐述在代理微服务组件中进行微服务注册的具体过程。其中,该方法至少可以包括以下步骤S101-步骤S103:

[0105] 步骤S101,代理微服务组件获取接口网关所发送的针对接口网关中的代理接口的路由请求;路由请求由客户端向接口网关所发送;代理接口是由代理微服务组件对代理接口进行微服务平台注册后,发布到接口网关中的;代理微服务组件是基于微服务接入标准规则所创建的,代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;路由请求是由接口网关对路由请求进行请求检测后,向代理微服务组件所发送的。

[0106] 本申请中,代理微服务组件可以是指基于微服务平台所创建的代理服务,其可以部署于微服务平台对应的微服务平台设备中,则该代理服务该微服务平台设备中的一个组件,用于执行相应的业务,则该代理服务可以称之为代理微服务组件。为了提高微服务平台的适配性,使得无法进行微服务改造的第三方专业系统(具体如,地铁轨交场景中无法进行微服务改造的电力专业系统、电梯专业系统等等。也可称之为第三方业务系统,以下将简称为业务系统),能够接入微服务平台,从而有效利用到微服务平台所提供的微服务,本申请可以基于微服务平台的开发标准规范(本申请可称之为接入标准规则),创建一个代理微服务,并将其部署于微服务平台设备中,该代理微服务可以用于对这些业务系统的API接口(应用程序接口)进行动态代理,得到代码接口(可称之为代理接口),而这些代理接口可以部署到微服务平台中,由此,业务系统可以接入微服务平台。代理微服务可以作为微服务平台设备中的一个组件(以下将称之为代理微服务组件),主要用于进行API接口的动态代理。

[0107] 其中,可以理解的是,本申请中的对代理接口进行微服务平台注册,实际上可以是指将代理接口在微服务平台中进行注册,也就是说,对于某个业务系统的某个应用程序接口(API接口),代理微服务组件可以对其进行动态编译,得到一个代码接口作为代理接口,

随后,代理微服务组件可以将该代理接口注册到微服务平台中,并将其发布于接口网关中。其具体过程主要为:代理微服务组件可以从微服务平台对应的程序接口管理平台(API管理平台)中,拉取应用程序接口的接口配置信息;随后,可以在代理微服务组件中,基于接口配置信息,生成针对应用程序接口的代理接口;代理微服务组件可以将代理接口在微服务平台中进行注册;当在微服务平台中完成对代理接口的注册时,即可将代理接口发布于接口网关中。

[0108] 应当理解的是,程序接口管理平台可以是指用于对应用程序接口的接口配置信息进行存储管理的平台,其可以与微服务平台、接口网关等一起部署到微服务平台设备中。而用户可以在程序接口管理平台中,定义配置某个应用程序接口的传入参数(可简称为入参)、传出参数(可简称为出参)、接口请求路径、接口请求方式、接口描述信息、接口限流阈值以及接口有效时长参数等相关信息,这些信息可以称之为接口配置信息。而代理微服务组件可以动态定时从程序接口管理平台拉取增量变化的程序应用接口的接口配置信息,基于该接口配置信息,代理微服务组件可以生成针对应用程序接口的代理接口。随后,代理微服务组件可以将该代理接口注册到微服务平台中,并在完成注册时,将该代理接口发布到接口网关中。

[0109] 需要说明的是,这些接口配置信息也可以用于应用方(如客户端)进行查看,应用方可以基于接口配置信息来请求相应的接口。例如,可以基于接口配置信息中的传出参数(可简称为出参)、接口请求路径、接口请求方式、接口描述信息、接口限流阈值以及接口有效时长参数等,来发起相应的接口请求。

[0110] 其中,可以理解的是,代理微服务组件具体可以是通过接口配置信息中的传入参数,来生成针对应用程序接口的代理接口,其具体实现过程可以为:在代理微服务组件中,可以基于传入参数,生成针对应用程序接口的代理请求代码;随后,可以将代理请求代码传入至虚拟机中,通过虚拟机中的动态编译工具可以对代理请求代码进行动态编译,得到动态编译字节流;随后,可以将动态编译字节流加载至虚拟机中,得到动态接口代码,可以将动态接口代码确定为代理接口。

[0111] 可以理解的是,这里的代理请求代码可以是指某个程序编程语言(如java)的接口代码,代理微服务组件可以解析传入参数动态生成java接口代码,该java接口代码即可称之为代理请求代码;随后,可以通过虚拟机提供的动态编译工具,将java接口代码在运行时编译为字节流(称之为动态编译字节流);编译得到字节流后,可以再通过虚拟机将字节流加载到虚拟机的执行环境中(如加载到classpath中)。加载完成的字节流可以称之为动态接口代码,该动态接口代码即为该代理接口。

[0112] 可选的,应当理解的是,对于程序接口管理平台中的API接口的接口配置信息,是动态更新的,而一旦接口配置信息存在更新,那么代理微服务组件可以重新基于新的接口配置信息(可称之为更新接口配置信息),生成新的代理接口(更新代理接口),此外,代理微服务组件可以将微服务平台以及接口网关中的原有的代理接口进行卸载,并将该更新代理接口注册到微服务平台中,将更新代理接口发布到接口网关中。

[0113] 对于发布更新代理接口的具体实现方式可为:当应用程序接口的接口配置信息发生更新时,程序接口管理平台可以向代理微服务组件发送接口更新通知,而代理微服务组件可以接收程序接口管理平台发送的针对应用程序接口的接口更新通知;基于接口更新通

知,代理微服务组件可以从程序接口管理平台中,拉取应用程序接口的更新接口配置信息;随后,在代理微服务组件中,可以基于更新接口配置信息,生成针对应用程序接口的更新代理接口;随后,可以将更新代理接口在微服务平台中进行注册;当在微服务平台中完成对更新代理接口的注册时,可以将更新代理接口发布于接口网关中。

[0114] 其中,对于基于更新接口配置信息,生成针对应用程序接口的更新代理接口的具体实现方式,与上述对于基于接口配置信息,生成针对应用程序接口的代理接口的实现方式相同,具体可以为:在代理微服务组件中,可以基于更新接口配置信息中的传入参数(为便于理解,这里将之称为更新传入参数),生成针对应用程序接口的更新代理请求代码;随后,可以将更新代理请求代码传入至虚拟机中,通过虚拟机中的动态编译工具可以对更新代理请求代码进行动态编译,得到更新动态编译字节流;随后,可以将更新动态编译字节流加载至虚拟机中,得到更新动态接口代码,可以将更新动态接口代码确定为更新代理接口。也就是说,在生成更新代理接口时,代理微服务组件会重新执行一次生成代理接口的逻辑(先生成代理请求代码、再进行动态编译、最后再进行加载),以生成新的代理接口。

[0115] 其中,可以理解的是,在发布更新代理接口之前,需要先将原有的代理接口进行注销以及卸载。当然,也可以是先将原有的代理接口进行注销以及卸载后,再生成更新代理接口并发布。其具体实现方式可为:可以生成针对代理接口的接口卸载通知,并将接口卸载通知发送至接口网关,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载;随后,可以将代理接口从微服务平台中进行注销;当在微服务平台中完成对代理接口的注销,且在接收到接口网关发送的针对代理接口的卸载完成通知消息时,可以执行在代理微服务组件中,基于更新接口配置信息,生成针对应用程序接口的更新代理接口的步骤。

[0116] 步骤S102,将路由请求转发至业务系统。

[0117] 本申请中,通过上述可知,应用程序接口的接口配置信息中可以包括应用程序接口的接口限流阈值与接口有效时长参数,而接口网关在接收到客户端发送的路由请求后,可以基于接口限流阈值以及接口有效时长参数,对路由请求进行检测得到检测结果。其具体实现方式可为:在接口网关中,可以基于接口限流阈值获取路由请求对应的请求频率,基于接口有效时长参数获取路由请求的请求发起时间;随后,可以获取接口有效时长参数所指示的结束时间;若请求频率小于接口限流阈值,且请求发起时间早于结束时间,则可以将路由请求的检测结果显示为检测通过结果;而若请求频率小于接口限流阈值,且请求发起时间晚于结束时间,则可以将路由请求的检测结果显示为检测未通过结果;若请求频率大于接口限流阈值,且请求发起时间晚于结束时间,则可以将路由请求的检测结果显示为检测未通过结果;若请求频率大于接口限流阈值,且请求发起时间早于结束时间,则可以将路由请求的检测结果显示为检测未通过结果。

[0118] 其中,上述请求频率可以是指路由请求对应的请求次数,接口限流阈值可以是指上述最大请求次数,接口有效时长参数可以是指接口的维持时长(从发布时间开始算起),也就是说,只有在请求频率与请求发起时间均满足条件的情况下,才将检测结果确定为检测通过结果;若任一个因素(请求频率或请求发起时间)未满足条件,则将检测结果确定为检测未通过结果。

[0119] 当接口网关对路由请求进行请求检测得到检测结果后,若该检测结果为检测通过结果,则接口网关可以将该路由请求转发至代理微服务组件;而若该检测结果为检测未通

过结果,则接口网关可以不进行转发,接口网关此时可以向客户端返回一个请求未能成功转发的提示消息,用以提示客户端再次发起请求或进行其他处理。也就是说,对于代理微服务组件所接收到的路由请求,是由接口网关基于接口限流阈值与接口有效时长参数,对路由请求进行检测得到检测结果,且在检测结果为检测通过结果时,才向代理微服务组件所发送的。

[0120] 在接口网关将路由转发至代理微服务组件后,代理微服务组件可以将路由请求转发至业务系统,其中,代理微服务组件可以执行代理微服务组件中的该代理接口对应的动态接口代码,用以将该路由请求转发至业务系统。

[0121] 步骤S103,接收所述业务系统返回的针对所述路由请求的请求处理结果,将所述请求处理结果返回至所述接口网关,以使所述接口网关将所述请求处理结果返回至所述客户端。

[0122] 本申请中,业务系统可以对上述路由请求进行请求处理得到请求处理结果,业务系统可以将该请求处理结果返回至代理微服务组件,而该代理微服务组件可以将该请求处理结果返回至接口网关,接口网关可以将该请求处理结果再返回至客户端。

[0123] 应当理解的是,对代理接口进行微服务平台注册是指将代理接口注册到微服务平台中进行注册,在微服务平台中完成对代理接口的注册时,微服务平台可以向业务系统提供相应的微服务。例如,可以在微服务平台对业务系统的服务进行全链路监测、调用日志收集、对接口进行灰度发布等等。

[0124] 具体的,微服务平台提供的微服务中可以包括全链路实时检测的服务,那么代理微服务组件可以基于微服务平台对业务系统进行全链路实时检测,从而可以及时检测到业务系统中发生异常的对象(可称之为异常对象,例如某个设备、某个零件等),进而可以使得业务系统能够及时对该异常对象进行处理(例如,修正处理)。其具体实现方式可为:基于代理接口,接收微服务平台发送的针对业务系统的异常处理提示信息;异常处理提示信息包括针对业务系统的异常对象,以及异常对象在业务系统中所处的异常位置;异常对象是指微服务平台基于微服务中的全链路检测服务,对业务系统进行全链路实时检测所检测到的发生异常的对象;代理微服务组件可以将异常处理提示信息发送至业务系统,以使业务系统在异常位置处获取到异常对象,且对异常对象进行修正处理。

[0125] 应当理解,代理微服务组件可以通过代理接口,向微服务平台发送一个进行全链路实时检测的通知信息,而微服务平台可以基于代理接口及时地对业务系统进行全链路实时检测,以检测业务系统中发生异常的对象,若检测到某个对象出现异常,则该微服务平台可以将该对象确定为异常对象,并获取到该异常对象所处的位置(可称之为异常位置),微服务平台可以向代理微服务组件返回该异常对象以及该异常位置。当然,微服务平台可以直接将该针对业务系统的异常对象以及该异常位置返回至代理微服务组件,由该代理微服务组件来生成包括该异常对象以及该异常位置的异常处理提示信息,再发送至业务系统,以使该业务系统快速从该异常位置处获取到该异常对象,并对该异常对象进行修正处理;微服务平台也可以先基于异常对象以及该异常位置生成一个异常处理提示信息,将该异常处理提示信息发送至代理微服务组件后,代理微服务组件可以不用进行任何处理,而是直接将该异常处理提示信息转发至业务系统,以使该业务系统快速从该异常位置处获取到该异常对象,并对该异常对象进行修正处理。

[0126] 同样,具体的,微服务平台提供的微服务中可以包括全链路性能分析的服务,那么代理微服务组件可以基于微服务平台对业务系统的系统设备进行全链路实时性能分析,从而可以分析确定业务系统中哪一个系统设备不再支持运行,进而可以使得业务系统不再运行该系统设备。其具体实现方式可为:基于代理接口,可以接收微服务平台发送的针对业务系统的设备性能信息;其中,设备性能信息是由微服务平台基于微服务中的全链路性能分析服务,在第一目标时间段内对业务系统中的系统设备进行全链路实时性能分析所得;设备性能信息可以包括业务系统中系统设备的实时响应时长以及设备错误频率;根据实时响应时长以及设备错误频率,代理微服务组件可以确定系统设备的设备状态;若系统设备的设备状态为异常运行状态,则可以生成针对系统设备的暂停运行提示信息;代理微服务组件可以将暂停运行提示信息发送至业务系统,以使业务系统在第二目标时间段内暂停运行系统设备;其中,第二目标时间段晚于第一目标时间段。

[0127] 其中,对于根据实时响应时长以及设备错误频率,确定系统设备的设备状态的具体实现方式可为:可以将实时响应时长与响应时长阈值进行比较,并将设备错误频率与错误频率阈值进行比较;若实时响应时长小于响应时长阈值,且设备错误频率小于错误频率阈值,则可以将系统的设备状态确定为正常运行状态;若实时响应时长小于响应时长阈值,且设备错误频率大于错误频率阈值,则可以将系统的设备状态确定为异常运行状态;若实时响应时长大于响应时长阈值,且设备错误频率小于错误频率阈值,则可以将系统的设备状态确定为异常运行状态;若实时响应时长大于响应时长阈值,且设备错误频率大于错误频率阈值,则可以将系统的设备状态确定为异常运行状态。

[0128] 可以理解的是,将代理接口注册到微服务平台后,微服务平台可以为业务系统提供相应的微服务。例如,通过微服务平台,可以对业务系统的系统设备进行全链路实时性能分析,例如,可以统计每个系统设备在某个时间段(如第一目标时间段)内的实时响应时长以及设备错误频率,其中实时响应时长可以是指该时间段内的平均响应时长,例如,可以在第一目标时间段内每隔一定时间(如3s)获取一次系统设备的响应时长(从系统设备接收到处理通知到系统设备返回处理结果的持续时长),再求取这些响应时长的均值得到平均响应时长后,该平均响应时长即可作为该系统设备在该第一目标时间段内的实时响应时长。设备错误频率可以是指系统设备在该第一目标时间段内发生错误的频率(次数)。根据该实时响应时长与设备错误频率,即可共同确定出系统设备的设备状态。例如,只有在实时响应时长较小、设备错误频率较小时,才可以确定系统设备的处理性能足够优异,此时可以在下一个时间段(如第二目标时间段)内,继续运行该系统设备。而若实时响应时长较大、或是设备错误频率较大时,可以确定系统设备的处理性能并不足以继续运行,此时可以在下一个时间段(如第二目标时间段)内,暂停运行该系统设备。

[0129] 同样,具体的,微服务平台提供的微服务中可以包括对接口进行灰度发布的服务,那么代理微服务组件可以基于微服务平台对业务系统的应用程序接口进行灰度发布,以代理接口为例,可以先将代理接口发布至较少的对象(如客户端,可称之为灰度发布对象),在这些灰度发布对象对该代理接口的反馈大多数为正向反馈时,再将该代理接口进行全量发布(也就是将代理接口发布至所有的对象)。以上述代理接口的接口属性为灰度测试属性(即代理接口为灰度发布),代理接口的灰度发布对象的数量为至少两个,每个灰度发布对象是指具备代理接口的请求权限的对象(即每个灰度发布对象具备发起针对代理接口的路

由请求的权限),至少两个灰度发布对象包括客户端为例,其具体实现方式可为:可以接收至少两个灰度发布对象分别针对代理接口的使用反馈信息;基于至少两个使用反馈信息,可以确定代理接口的接口发布状态;若接口发布状态为合格发布状态,则可以将代理接口的接口属性从灰度测试属性切换为全量发布属性,且将代理接口发布至全量发布对象;被发布代理接口的全量发布对象具备代理接口的请求权限;而若接口发布状态为待定发布状态,则可以将代理接口的接口属性从灰度测试属性切换为发布待定属性,且将代理接口进行微服务平台注销,向接口网关发送针对代理接口的接口卸载通知,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载。

[0130] 其中,对于基于至少两个使用反馈信息,确定代理接口的接口发布状态的具体实现方式可为:可以将至少两个使用反馈信息中的为正向反馈信息的使用反馈信息,确定为待统计反馈信息;随后,可以统计待统计反馈信息的正向反馈数量,以及至少两个使用反馈信息的反馈总数量;可以确定正向反馈数量与反馈总数量之间的数量比值;若数量比值大于比值阈值,则可以将接口发布状态确定为合格发布状态;而若数量比值小于比值阈值,则可以将接口发布状态确定为待定发布状态。

[0131] 也就是说,若至少两个使用反馈信息中,存在超过一定数值的使用反馈信息均为正向反馈信息,则可以将该代理接口的接口发布状态确定为合格发布状态,可以将该代理接口的接口属性切换为全量发布属性,同时将该代理接口发布至业务系统对应的所有的对象(可能应用到业务系统的对象,可称之为全量发布对象)。

[0132] 应当理解,通过上述可知,本申请实施例通过代理微服务组件,可以使得无法进行微服务改造的业务系统,能够接入微服务平台,从而有效利用到微服务的治理功能。可以很好地提升微服务平台的适配性,进而提升微服务平台的利用率。

[0133] 在本申请实施例中,可以基于微服务接入标准规则,创建一个代理微服务组件,该代理微服务组件可以对业务系统的应用程序接口进行动态代理形成代理接口,将该代理接口进行微服务平台注册,且将代理接口发布到接口网关中后,该业务系统即可有效利用到微服务(如,接口网关提供的服务等)。例如,接口网关提供的服务可为对代理接口的路由请求进行检测的服务,当某个客户端期望访问业务系统时,则该客户端可以向接口网关发送针对该代理接口的路由请求;而接口网关可对路由请求进行请求检测,并在检测后将该路由请求发送至代理微服务组件;随后,代理微服务组件可以将该路由请求转发至业务系统,而业务系统在接收到路由请求后即可进行请求处理得到请求处理结果。应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在将代理接口进行微服务注册并注册到接口网关后,可以使得业务系统能够有效利用到微服务。通过代理微服务组件,可以支持不能有效利用微服务的业务系统(如地铁轨交场景下的检修专业系统、电力专业系统等等),很好地利用到微服务,由此可以很好地提升微服务的适配度,从而提高微服务的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0134] 进一步地,请一并参见图4,图4是本申请实施例提供的一种系统架构图。其中,如图4所示的系统架构可以包括代理微服务设备的架构。如图4所示,该系统架构可以包括客户端、接口网关、代理微服务组件、微服务平台、程序接口管理平台、未做微服务改造的业务系统以及已做微服务改造的业务系统。其中,接口网关可以是指微服务网关,该接口网关、

代理微服务组件、微服务平台、程序接口管理平台可以共同集成于一个设备中,该设备可以称之为微服务平台设备(也可以称之为代理微服务设备)。也就是说,对于代理微服务设备的架构,可以包括接口网关、代理微服务组件、微服务平台以及程序接口管理平台。为便于理解,以下将对各个模块进行阐述。

[0135] 未做微服务改造的业务系统:未做微服务改造的业务系统可以包括专业系统A、…专业系统M(M可为正整数)。每个未做微服务改造的业务系统(例如,如图所示的专业系统A)具体可以是指与微服务平台不适配,从而无法部署到微服务平台的老旧的专业系统(第三方专业系统),这些业务系统无法有效利用到微服务平台所提供的微服务。

[0136] 已做微服务改造的业务系统:已做微服务改造的业务系统可以包括专业系统1、专业系统2、…、专业系统n(n可为正整数)。每个已做微服务改造的业务系统(例如,如图所示的专业系统1)具体可以是指与微服务平台相适配,从而可以部署到且已经部署到微服务平台的专业系统,当部署到微服务平台后,这些业务系统可以有效利用到微服务平台提供的微服务。

[0137] 程序接口管理平台:程序接口管理平台(API管理平台)可以用于开发者定义业务系统(如未做微服务改造的业务系统、已做微服务改造的业务系统)的应用程序接口的接口配置信息。这些接口配置信息可以动态存储在该程序接口管理平台中。也就是说,程序接口管理平台可以用于定义业务系统的应用程序接口的配置信息(可称之为接口配置信息),而程序接口管理平台可以对这些配置信息进行管理。

[0138] 代理微服务组件:代理微服务组件可以用于从程序接口管理平台动态拉取应用程序接口的接口配置信息(这里具体可以是指未做微服务改造的业务系统的应用程序接口的接口配置信息),并根据该接口配置信息进行动态编译,得到代理接口。动态编译得到代理接口的具体过程可以包括:根据该接口配置信息中的传入参数,生成代理请求代码;代理微服务组件还可以用于对该代理请求代码进行动态编译以及动态加载,形成动态接口代码,该动态接口代码即可为该代理接口。同时,代理微服务组件还可以用于将该代理接口注册到微服务框架(微服务平台)中,并将该代理接口发布到接口网关中。当代理微服务组件接收到接口网关发送的针对该代理接口的路由请求时,代理微服务组件可以执行动态生成的代码,并将该请求转发至对应的业务系统。也就是说,代理微服务组件可以对上述未做微服务改造的业务系统的应用程序接口进行动态代理,使得动态代理的代理接口能够与微服务平台相适配,从而可以将代理接口能够部署到微服务平台中,进而使其对应的业务系统能够有效利用到微服务平台提供的微服务。

[0139] 需要说明的是,对于未做微服务改造的业务系统的应用程序接口的配置信息,代理微服务组件可以基于这些配置信息生成代理接口,再由该代理微服务组件对该代理接口在微服务平台中进行注册,以及发布到接口网关;而对于已做微服务改造的业务系统,在程序接口管理平台中定义完成应用程序接口的配置信息后,可以直接将该应用程序接口注册到微服务平台,且发布到接口网关中。

[0140] 接口网关:接口网关具备限流、熔断、超时以及灰度等服务功能,当某个接口(如上述未做微服务改造的业务系统的代理接口、已做微服务改造的业务系统的API接口)被发布到接口网关中后,接口网关可以为该接口提供限流、熔断、超时以及灰度等服务。例如,接口网关可以基于限流功能来转发客户端发起的路由请求,当客户端发起针对某个接口(可以

包括接口网关中包括的所有接口,例如已做微服务改造的API接口以及未做微服务改造的代理接口)的路由请求时,该接口网关可以对其基于策略对其进行检测,例如,可以基于限流策略对其进行检测,并在检测结果为通过结果时,再将该路由请求进行转发。

[0141] 可以理解的是,对于未做微服务改造的业务系统,接口网关在对路由请求进行检测后,可以将该请求先转发至代理微服务组件,该代理微服务组件可以执行动态接口代码以将该路由请求转发至未做微服务改造的业务系统;随后,未做微服务改造的业务系统可以执行请求并得到请求执行结果,未做微服务改造的业务系统可以将该请求执行结果返回至代理微服务组件,代理微服务组件可以将该请求执行结果转发至接口网关,而接口网关可以将该请求执行结果返回至客户端。也就是说,代理微服务组件可以在接口网关与未做微服务改造的业务系统之间,进行数据转发。

[0142] 而对于已做微服务改造的业务系统,接口网关在对路由请求进行检测后,可以将该请求直接转发至已做微服务改造的业务系统,已做微服务改造的业务系统可以执行请求并得到请求执行结果,已做微服务改造的业务系统可以将该请求执行结果返回至接口网关,接口网关可以将该请求执行结果返回至客户端。也就是说,对于已做微服务改造的业务系统,可以直接与接口网关进行通信。

[0143] 微服务平台:微服务平台具备镜像管理、服务发布、服务编排以及全链路实时监测等微服务,当某个接口(如上述未做微服务改造的业务系统的代理接口、已做微服务改造的业务系统的API接口)被注册到微服务平台后,微服务平台可以为该接口提供镜像管理、服务发布、服务编排以及全链路实时监测等微服务。也就是说,微服务平台主要用于对已经进行微服务平台注册的接口(如已做微服务改造的业务系统的应用程序接口、未做微服务改造的业务系统的代理接口)所对应的业务系统,提供微服务功能。

[0144] 应当理解,对于未做微服务改造的业务系统,可以通过代理微服务组件,将代理接口部署到微服务平台中。在后续应用中,也可以由代理微服务组件来调用微服务平台中的某个微服务,例如,代理微服务组件可以向微服务平台发送针对某个微服务的微服务调用信息,而微服务平台在基于该微服务得到微服务执行结果后,也可以先向代理微服务组件发送该微服务执行结果,再由代理微服务组件将该微服务执行结果发送至未做微服务改造的业务系统。而对于已做微服务改造的业务系统,已做微服务改造的业务系统可以将应用程序接口直接部署到微服务平台中。在后续应用中,已做微服务改造的业务系统可以直接向微服务平台发送微服务通知信息,而微服务平台在得到微服务执行结果后,也可以直接将该微服务执行结果发送至已做微服务改造的业务系统。

[0145] 也就是说,在本申请中,对于未做微服务改造的业务系统,主要通过代理微服务组件来对其应用程序接口进行动态代理,使其能够部署到微服务平台中;同时,代理微服务组件也可以在未做微服务改造的业务系统与微服务平台、接口网关之间进行数据转发。即对于未做微服务改造的业务系统,主要通过代理微服务组件来与接口网关、微服务平台之间进行通信。

[0146] 通过上述可知,本申请通过在微服务平台设备中增设代理微服务组件的方式,使其能够对无法部署到微服务平台的业务系统的应用程序接口进行动态代理,从而可以使得动态代理得到的代理接口能够部署到微服务平台中,其对应的无法部署到微服务平台的业务系统能够有效利用到微服务平台提供的微服务。对于无法部署到微服务平台的业务系统

与微服务平台之间的通信,也可以由代理微服务组件进行转发,无需浪费过多资源,即可让这些业务系统能够利用到微服务平台,可以在保持较低成本的同时,很好地提高微服务平台的适配度和利用率。

[0147] 应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在注册到接口网关后,基于接口网关可以对业务系统进行限流、检测、超时等服务处理;而将代理接口进行微服务平台注册,也可以使得业务系统能够有效利用到微服务,通过代理微服务组件,可以支持不能有效利用微服务设备的业务系统,很好地利用到微服务,从而可以很好地改善微服务设备的适配度不高的问题,提高微服务设备的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0148] 进一步地,请一并参见图5,图5是本申请实施例提供的一种对代理接口进行微服务注册的流程示意图。如图5所示,该流程至少可以包括以下步骤S51-步骤S55:

[0149] 步骤S51,代理微服务组件从程序接口管理平台拉取接口配置信息。

[0150] 步骤S52,代理微服务组件根据接口配置信息中的传入参数生成代理请求代码。

[0151] 步骤S53,代理微服务组件动态编译代理请求代码,得到字节流。

[0152] 步骤S54,代理微服务组件动态加载字节流,得到代理接口。

[0153] 步骤S55,代理微服务组件注册代理接口到微服务平台。

[0154] 其中,对于步骤S51-步骤S55的具体实现方式,可以参见上述图3所对应实施例中对于代理微服务组件生成代理接口的描述,这里将不再进行赘述。

[0155] 应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在注册到接口网关后,基于接口网关可以对业务系统进行限流、检测、超时等服务处理;而将代理接口进行微服务平台注册,也可以使得业务系统能够有效利用到微服务,通过代理微服务组件,可以支持不能有效利用微服务设备的业务系统,很好地利用到微服务,从而可以很好地改善微服务设备的适配度不高的问题,提高微服务设备的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0156] 进一步地,请一并参见图6,图6是本申请实施例提供的一种对更新代理接口进行微服务注册的流程示意图。如图6所示,该流程至少可以包括以下步骤S61-步骤S66:

[0157] 步骤S61,代理微服务组件从程序接口管理平台拉取更新后的接口配置信息(更新接口配置信息)。

[0158] 步骤S62,代理微服务组件将原有的代理接口进行卸载。

[0159] 步骤S63,代理微服务组件根据更新后的接口配置信息中的传入参数生成代理请求代码。

[0160] 步骤S64,代理微服务组件动态编译代理请求代码,得到字节流。

[0161] 步骤S65,代理微服务组件动态加载字节流,得到更新后的代理接口(更新代理接口)。

[0162] 步骤S66,代理微服务组件注册更新代理接口到微服务平台。

[0163] 其中,对于步骤S61-步骤S66的具体实现方式,可以参见上述图3所对应实施例中对于代理微服务组件生成更新代理接口并注册的描述,这里将不再进行赘述。

[0164] 应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在注册到接口网关后,基于接口网关可以对业务系统进行限流、检测、超时等服务处理;而将代理接口进行微服务平台注册,也可以使得业务系统能够有效利用到微服务,通过代理微服务组件,可以支持不能有效利用微服务设备的业务系统,很好地利用到微服务,从而可以很好地改善微服务设备的适配度不高的问题,提高微服务设备的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0165] 进一步地,请参见图7,图7是本申请实施例提供的一种系统交互流程图。

[0166] 如图7所示,该交互流程至少可以包括以下步骤S71-步骤S76:

[0167] 步骤S71,客户端向接口网关请求代理接口。

[0168] 具体的,客户端可以向接口网关发送针对代理接口的路由请求。

[0169] 步骤S72,接口网关转发请求至代理微服务组件。

[0170] 步骤S73,代理微服务组件执行生成的动态代码以转发请求至业务系统。

[0171] 步骤S74,业务系统进行请求处理得到执行结果,将执行结果返回至代理微服务组件。

[0172] 步骤S75,代理微服务组件向接口网关返回执行结果。

[0173] 步骤S76,接口网关向客户端返回执行结果。

[0174] 具体的,本申请实施例仅是将交互过程进行直观呈现描述,对于步骤S71-步骤S76中每个步骤的具体实现方式,可以参见上述图3所对应实施例中的描述,这里将不再进行赘述。

[0175] 应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在注册到接口网关后,基于接口网关可以对业务系统进行限流、检测、超时等服务处理;而将代理接口进行微服务平台注册,也可以使得业务系统能够有效利用到微服务,通过代理微服务组件,可以支持不能有效利用微服务设备的业务系统,很好地利用到微服务,从而可以很好地改善微服务设备的适配度不高的问题,提高微服务设备的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0176] 进一步地,请参见图8,图8是本申请实施例提供的一种地铁微服务注册装置的结构示意图。该地铁微服务注册装置可以是运行于计算机设备(如微服务平台设备,该微服务平台设备可以包括接口网关和代理微服务组件)中的一个计算机程序(包括程序代码),例如该地铁微服务注册装置为一个应用软件;该地铁微服务注册装置可以用于执行图3所示的方法。如图8所示,该地铁微服务注册装置1可以包括:请求获取模块11、请求转发模块12以及结果转发模块13。

[0177] 请求获取模块11,用于获取接口网关所发送的针对接口网关中的代理接口的路由请求;路由请求由客户端向接口网关所发送;代理接口是由代理微服务组件对代理接口进行微服务平台注册后,发布到接口网关中的;代理微服务组件是基于微服务接入标准规则所创建的,代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;路由请求是由接口网关对路由请求进行请求检测后,向代理微服务组件所发送的;

- [0178] 请求转发模块12,用于将路由请求转发至业务系统;
- [0179] 结果转发模块13,用于接收业务系统返回的针对路由请求的请求处理结果,将请求处理结果返回至接口网关,以使接口网关将请求处理结果返回至客户端。
- [0180] 其中,请求获取模块11、请求转发模块12以及结果转发模块13的具体实现方式,可以参见上述图3所对应实施例中步骤S101-步骤S103的描述,这里将不再进行赘述。
- [0181] 在一个实施例中,对代理接口进行微服务平台注册是指将代理接口在微服务平台中进行注册;
- [0182] 地铁微服务注册装置1还可以包括:配置信息拉取模块16、接口生成模块17、接口注册模块18、接口发布模块19。
- [0183] 配置信息拉取模块16,用于从微服务平台对应的程序接口管理平台中,拉取应用程序接口的接口配置信息;
- [0184] 接口生成模块17,用于在代理微服务组件中,基于接口配置信息,生成针对应用程序接口的代理接口;
- [0185] 接口注册模块18,用于将代理接口在微服务平台中进行注册;
- [0186] 接口发布模块19,用于当在微服务平台中完成对代理接口的注册时,将代理接口发布于接口网关中。
- [0187] 其中,配置信息拉取模块16、接口生成模块17、接口注册模块18、接口发布模块19的具体实现方式,可以参见上述图3所对应实施例中步骤S101的描述,这里将不再进行赘述。
- [0188] 在一个实施例中,接口配置信息包括应用程序接口的传入参数;
- [0189] 接口生成模块17可以包括:请求代码生成单元171、代码编译单元172以及代码加载单元173。
- [0190] 请求代码生成单元171,用于在代理微服务组件中,基于传入参数,生成针对应用程序接口的代理请求代码;
- [0191] 代码编译单元172,用于将代理请求代码传入至虚拟机中,通过虚拟机中的动态编译工具对代理请求代码进行动态编译,得到动态编译字节流;
- [0192] 代码加载单元173,用于将动态编译字节流加载至虚拟机中,得到动态接口代码,将动态接口代码确定为代理接口。
- [0193] 其中,请求代码生成单元171、代码编译单元172以及代码加载单元173的具体实现方式,可以参见上述图3所对应实施例中步骤S101的描述,这里将不再进行赘述。
- [0194] 在一个实施例中,地铁微服务注册装置1还可以包括:更新信息拉取模块20、更新接口生成模块21、更新接口注册模块22以及更新接口发布模块23。
- [0195] 更新信息拉取模块20,用于接收程序接口管理平台发送的针对应用程序接口的接口更新通知;
- [0196] 更新信息拉取模块20,还用于基于接口更新通知,从程序接口管理平台中,拉取应用程序接口的更新接口配置信息;
- [0197] 更新接口生成模块21,用于在代理微服务组件中,基于更新接口配置信息,生成针对应用程序接口的更新代理接口;
- [0198] 更新接口注册模块22,用于将更新代理接口在微服务平台中进行注册;

[0199] 更新接口发布模块23,用于当在微服务平台中完成对更新代理接口的注册时,将更新代理接口发布于接口网关中。

[0200] 其中,更新信息拉取模块20、更新接口生成模块21、更新接口注册模块22以及更新接口发布模块23的具体实现方式,可以参见上述图3所对应实施例中步骤S101的描述,这里将不再进行赘述。

[0201] 在一个实施例中,地铁微服务注册装置1还可以包括:接口卸载模块24、接口注销模块25以及步骤执行模块26。

[0202] 接口卸载模块24,用于生成针对代理接口的接口卸载通知,将接口卸载通知发送至接口网关,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载;

[0203] 接口注销模块25,用于将代理接口从微服务平台中进行注销;

[0204] 步骤执行模块26,用于当在微服务平台中完成对代理接口的注销,且在接收到接口网关发送的针对代理接口的卸载完成通知消息时,执行在代理微服务组件中,基于更新接口配置信息,生成针对应用程序接口的更新代理接口的步骤。

[0205] 其中,接口卸载模块24、接口注销模块25以及步骤执行模块26的具体实现方式,可以参见上述图3所对应实施例中步骤S101的描述,这里将不再进行赘述。

[0206] 在一个实施例中,接口配置信息包括应用程序接口的接口限流阈值以及接口有效时长参数;路由请求是由接口网关基于接口限流阈值与接口有效时长参数,对路由请求进行检测得到检测结果,且在检测结果为检测通过结果时,向代理微服务组件所发送的。

[0207] 在一个实施例中,地铁微服务注册装置1还可以包括:系统检测模块27。

[0208] 系统检测模块27,用于基于代理接口,接收微服务平台发送的针对业务系统的异常处理提示信息;异常处理提示信息包括针对业务系统的异常对象,以及异常对象在业务系统中所处的异常位置;异常对象是指微服务平台基于微服务中的全链路检测服务,对业务系统进行全链路实时检测所检测到的发生异常的对象;

[0209] 系统检测模块27,还用于将异常处理提示信息发送至业务系统,以使业务系统在异常位置处获取到异常对象,且对异常对象进行修正处理。

[0210] 其中,系统检测模块27的具体实现方式,可以参见上述图3所对应实施例中步骤S103的描述,这里将不再进行赘述。

[0211] 在一个实施例中,对代理接口进行微服务平台注册是指将代理接口在微服务平台中进行注册;当在微服务平台中完成对代理接口的注册时,微服务平台为业务系统提供微服务;

[0212] 地铁微服务注册装置1还可以包括:系统分析模块28、设备状态确定模块29以及暂停提示模块30。

[0213] 系统分析模块28,用于基于代理接口,接收微服务平台发送的针对业务系统的设备性能信息;设备性能信息是由微服务平台基于微服务中的全链路性能分析服务,在第一目标时间段内对业务系统中的系统设备进行全链路实时性能分析所得到的;设备性能信息包括业务系统中系统设备的实时响应时长以及设备错误频率;

[0214] 设备状态确定模块29,用于根据实时响应时长以及设备错误频率,确定系统设备的设备状态;

[0215] 暂停提示模块30,用于若系统设备的设备状态为异常运行状态,则生成针对系统

设备的暂停运行提示信息；

[0216] 暂停提示模块30,还用于通过代理微服务组件将暂停运行提示信息发送至业务系统,以使业务系统在第二目标时间段内暂停运行系统设备;第二目标时间段晚于第一目标时间段。

[0217] 其中,系统分析模块28、设备状态确定模块29以及暂停提示模块30的具体实现方式,可以参见上述图3所对应实施例中步骤S103的描述,这里将不再进行赘述。

[0218] 在一个实施例中,设备状态确定模块29可以包括:时长比较单元291以及状态确定单元292。

[0219] 时长比较单元291,用于将实时响应时长与响应时长阈值进行比较,将设备错误频率与错误频率阈值进行比较;

[0220] 状态确定单元292,用于若实时响应时长小于响应时长阈值,且设备错误频率小于错误频率阈值,则将系统的设备状态确定为正常运行状态;

[0221] 状态确定单元292,还用于若实时响应时长小于响应时长阈值,且设备错误频率大于错误频率阈值,则将系统的设备状态确定为异常运行状态;

[0222] 状态确定单元292,还用于若实时响应时长大于响应时长阈值,且设备错误频率小于错误频率阈值,则将系统的设备状态确定为异常运行状态;

[0223] 状态确定单元292,还用于若实时响应时长大于响应时长阈值,且设备错误频率大于错误频率阈值,则将系统的设备状态确定为异常运行状态。

[0224] 其中,时长比较单元291以及状态确定单元292的具体实现方式,可以参见上述图3所对应实施例中步骤S103的描述,这里将不再进行赘述。

[0225] 在一个实施例中,代理接口的接口属性为灰度测试属性,代理接口的灰度发布对象的数量为至少两个,每个灰度发布对象是指具备代理接口的请求权限的对象,至少两个灰度发布对象包括客户端;

[0226] 地铁微服务注册装置1还可以包括:反馈接收模块31、接口状态确定模块32以及接口处理模块33。

[0227] 反馈接收模块31,用于接收至少两个灰度发布对象分别针对代理接口的使用反馈信息;

[0228] 接口状态确定模块32,用于基于至少两个使用反馈信息,确定代理接口的接口发布状态;

[0229] 接口处理模块33,用于若接口发布状态为合格发布状态,则将代理接口的接口属性从灰度测试属性切换为全量发布属性,且将代理接口发布至全量发布对象;被发布代理接口的全量发布对象具备代理接口的请求权限;

[0230] 接口处理模块33,还用于若接口发布状态为待定发布状态,则将代理接口的接口属性从灰度测试属性切换为发布待定属性,且将代理接口进行微服务平台注销,向接口网关发送针对代理接口的接口卸载通知,以使接口网关基于接口卸载通知将接口网关中的代理接口进行卸载。

[0231] 其中,反馈接收模块31、接口状态确定模块32以及接口处理模块33的具体实现方式,可以参见上述图3所对应实施例中步骤S103的描述,这里将不再进行赘述。

[0232] 在一个实施例中,接口状态确定模块32可以包括:反馈统计单元321以及发布状态

确定单元322。

[0233] 反馈统计单元321,用于将至少两个使用反馈信息中的为正向反馈信息的使用反馈信息,确定为待统计反馈信息;

[0234] 反馈统计单元321,还用于统计待统计反馈信息的正向反馈数量,以及至少两个使用反馈信息的反馈总数量;

[0235] 反馈统计单元321,还用于确定正向反馈数量与反馈总数量之间的数量比值;

[0236] 发布状态确定单元322,用于若数量比值大于比值阈值,则将接口发布状态确定为合格发布状态;

[0237] 发布状态确定单元322,还用于若数量比值小于比值阈值,则将接口发布状态确定为待定发布状态。

[0238] 其中,反馈统计单元321以及发布状态确定单元322的具体实现方式,可以参见上述图3所对应实施例中步骤S103的描述,这里将不再进行赘述。

[0239] 在本申请实施例中,可以基于微服务接入标准规则,创建一个代理微服务组件,该代理微服务组件可以对业务系统的应用程序接口进行动态代理形成代理接口,将该代理接口进行微服务平台注册,且将代理接口发布到接口网关中后,该业务系统即可有效利用到微服务(如,接口网关提供的服务等)。例如,接口网关提供的服务可为对代理接口的路由请求进行检测的服务,当某个客户端期望访问业务系统时,则该客户端可以向接口网关发送针对该代理接口的路由请求;而接口网关可对路由请求进行请求检测,并在检测后将该路由请求发送至代理微服务组件;随后,代理微服务组件可以将该路由请求转发至业务系统,而业务系统在接收到路由请求后即可进行请求处理得到请求处理结果。应当理解,本申请可以通过创建代理微服务组件的形式,使得代理微服务组件能够对业务系统的应用程序接口进行动态编译,在将代理接口进行微服务注册并注册到接口网关后,可以使得业务系统能够有效利用到微服务。通过代理微服务组件,可以支持不能有效利用微服务的业务系统,很好地利用到微服务,由此可以很好地提升微服务的适配度,从而提高微服务的利用率。综上,本申请可以提高微服务的业务适配度,进而提高微服务的利用率。

[0240] 进一步地,请参见图9,图9是本申请实施例提供的一种计算机设备的结构示意图。如图9所示,上述图8所对应实施例中的地铁微服务注册装置1可以应用于上述计算机设备8000,上述计算机设备8000可以包括:处理器8001,网络接口8004和存储器8005,此外,上述计算机设备8000还包括:用户接口8003,和至少一个通信总线8002。其中,通信总线8002用于实现这些组件之间的连接通信。其中,用户接口8003可以包括显示屏(Display)、键盘(Keyboard),可选用户接口8003还可以包括标准的有线接口、无线接口。网络接口8004可选的可以包括标准的有线接口、无线接口(如WI-FI接口)。存储器8005可以是高速RAM存储器,也可以是非不稳定的存储器(non-volatile memory),例如至少一个磁盘存储器。存储器8005可选的还可以是至少一个位于远离前述处理器8001的存储装置。如图9所示,作为一种计算机可读存储介质的存储器8005中可以包括操作系统、网络通信模块、用户接口模块以及设备控制应用程序。

[0241] 在图9所示的计算机设备8000中,网络接口8004可提供网络通讯功能;而用户接口8003主要用于为用户提供输入的接口;而处理器8001可以用于调用存储器8005中存储的设备控制应用程序,以实现:

[0242] 获取接口网关所发送的针对接口网关中的代理接口的路由请求;路由请求由客户端向接口网关所发送;代理接口是由代理微服务组件对代理接口进行微服务平台注册后,发布到接口网关中的;代理微服务组件是基于微服务接入标准规则所创建的,代理接口是指代理微服务组件通过对业务系统的应用程序接口,进行动态编译后所生成的符合微服务接入标准规则的代码接口;路由请求是由接口网关对路由请求进行请求检测后,向代理微服务组件所发送的;

[0243] 将路由请求转发至业务系统;

[0244] 接收业务系统返回的针对路由请求的请求处理结果,将请求处理结果返回至接口网关,以使接口网关将请求处理结果返回至客户端。

[0245] 应当理解,本申请实施例中所描述的计算机设备8000可执行前文图3所对应实施例中对该地铁微服务注册方法的描述,也可执行前文图8所对应实施例中对该地铁微服务注册装置1的描述,在此不再赘述。另外,对采用相同方法的有益效果描述,也不再赘述。

[0246] 此外,这里需要指出的是:本申请实施例还提供了一种计算机可读存储介质,且上述计算机可读存储介质中存储有前文提及的数据处理的计算机设备8000所执行的计算机程序,且上述计算机程序包括程序指令,当上述处理器执行上述程序指令时,能够执行前文图3所对应实施例中对上述地铁微服务注册方法的描述,因此,这里将不再赘述。另外,对采用相同方法的有益效果描述,也不再赘述。对于本申请所涉及的计算机可读存储介质实施例中未披露的技术细节,请参照本申请方法实施例的描述。

[0247] 上述计算机可读存储介质可以是前述任一实施例提供的地铁微服务注册装置或者上述计算机设备的内部存储单元,例如计算机设备的硬盘或内存。该计算机可读存储介质也可以是该计算机设备的外部存储设备,例如该计算机设备上配备的插接式硬盘,智能存储卡(smart media card,SMC),安全数字(secure digital,SD)卡,闪存卡(flash card)等。进一步地,该计算机可读存储介质还可以既包括该计算机设备的内部存储单元也包括外部存储设备。该计算机可读存储介质用于存储该计算机程序以及该计算机设备所需的其他程序和数据。该计算机可读存储介质还可以用于暂时地存储已经输出或者将要输出的数据。

[0248] 本申请的一个方面,提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行本申请实施例中一方面提供的方法。

[0249] 本申请实施例的说明书和权利要求书及附图中的术语“第一”、“第二”等是用于区别不同对象,而非用于描述特定顺序。此外,术语“包括”以及它们任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、装置、产品或设备没有限定于已列出的步骤或模块,而是可选地还包括没有列出的步骤或模块,或可选地还包括对于这些过程、方法、装置、产品或设备固有的其他步骤单元。

[0250] 本领域普通技术人员可以意识到,结合本文中所公开的实施例描述的各示例的单元及算法步骤,能够以电子硬件、计算机软件或者二者的结合来实现,为了清楚地说明硬件和软件的可互换性,在上述说明中已经按照功能一般性地描述了各示例的组成及步骤。这

些功能究竟以硬件还是软件方式来执行,取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能,但是这种实现不应认为超出本申请的范围。

[0251] 本申请实施例提供的方法及相关装置是参照本申请实施例提供的方法流程图和/或结构示意图来描述的,具体可由计算机程序指令实现方法流程图和/或结构示意图的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。这些计算机程序指令可提供到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或结构示意图一个方框或多个方框中指定的功能的装置。这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或结构示意图一个方框或多个方框中指定的功能。这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或结构示意图一个方框或多个方框中指定的功能的步骤。

[0252] 以上所揭露的仅为本申请较佳实施例而已,当然不能以此来限定本申请之权利范围,因此依本申请权利要求所作的等同变化,仍属本申请所涵盖的范围。

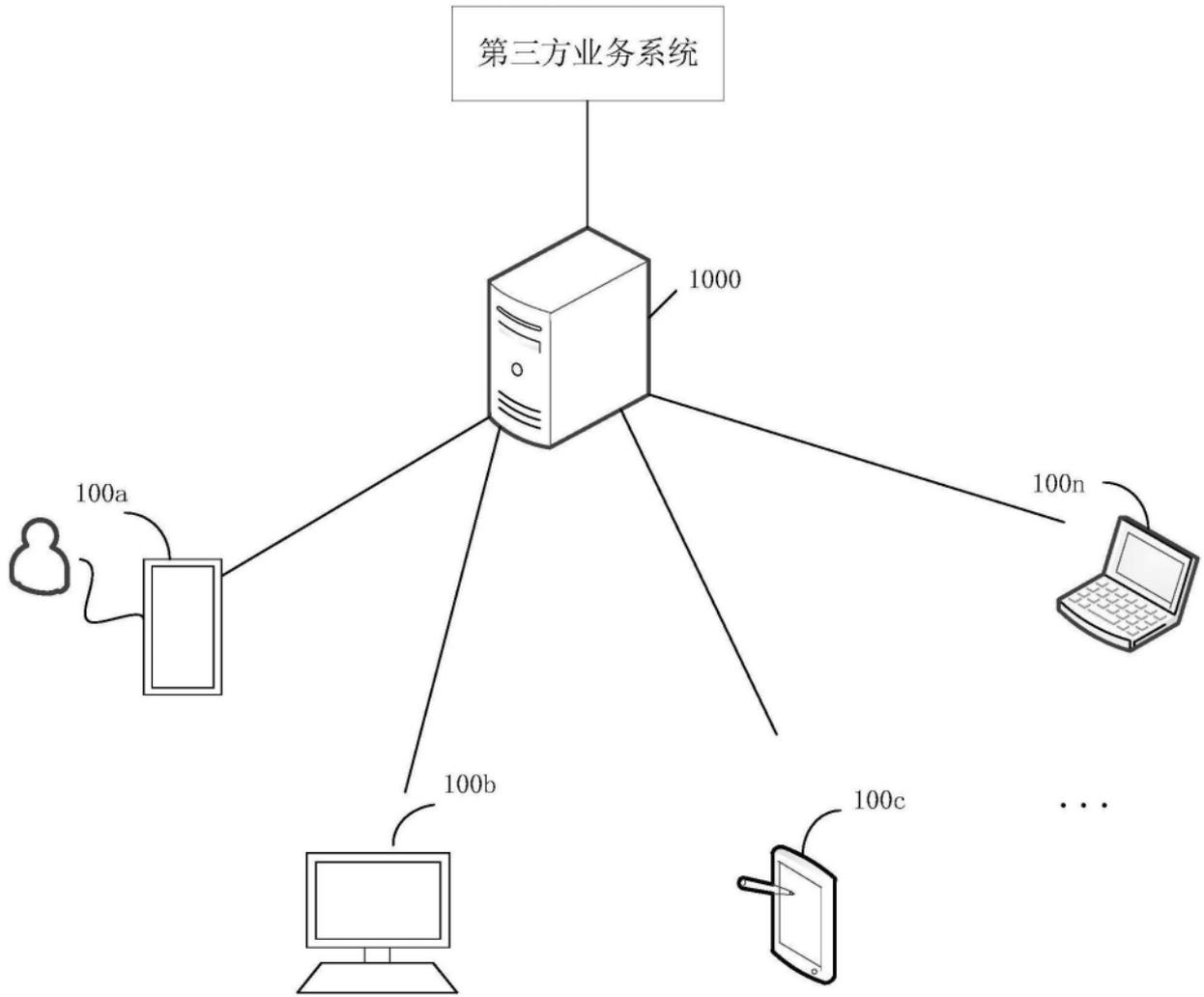


图1

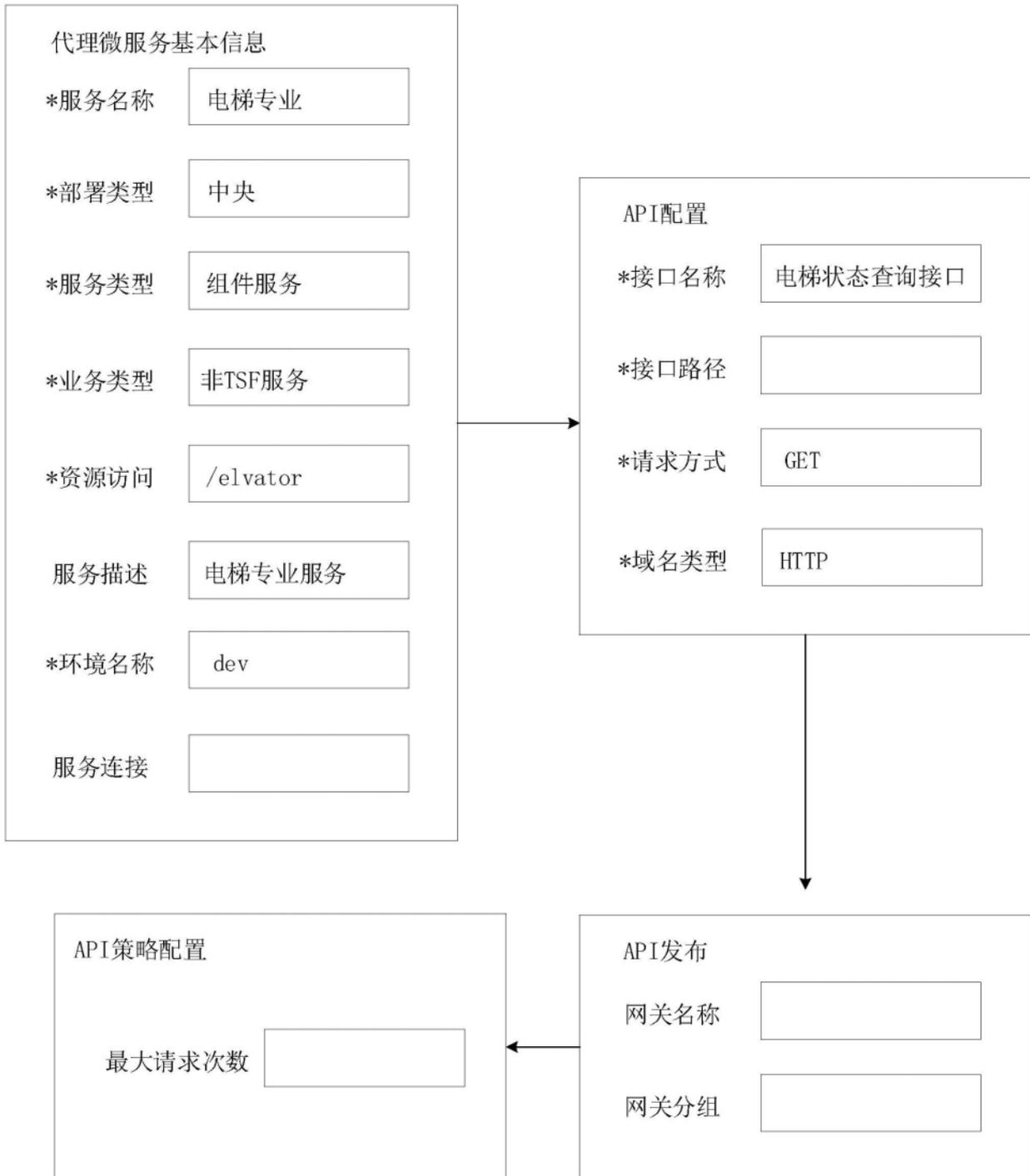


图2

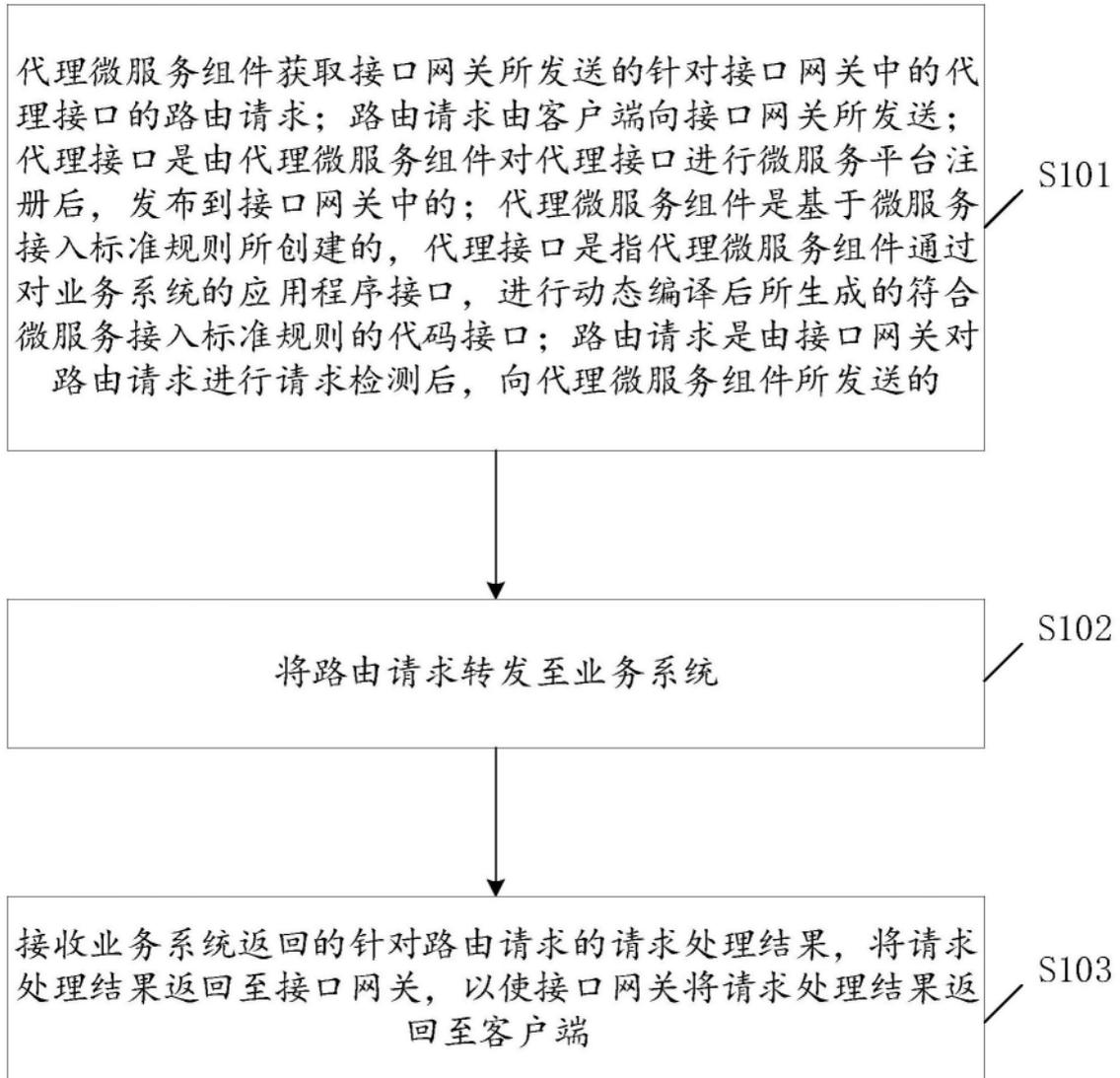


图3

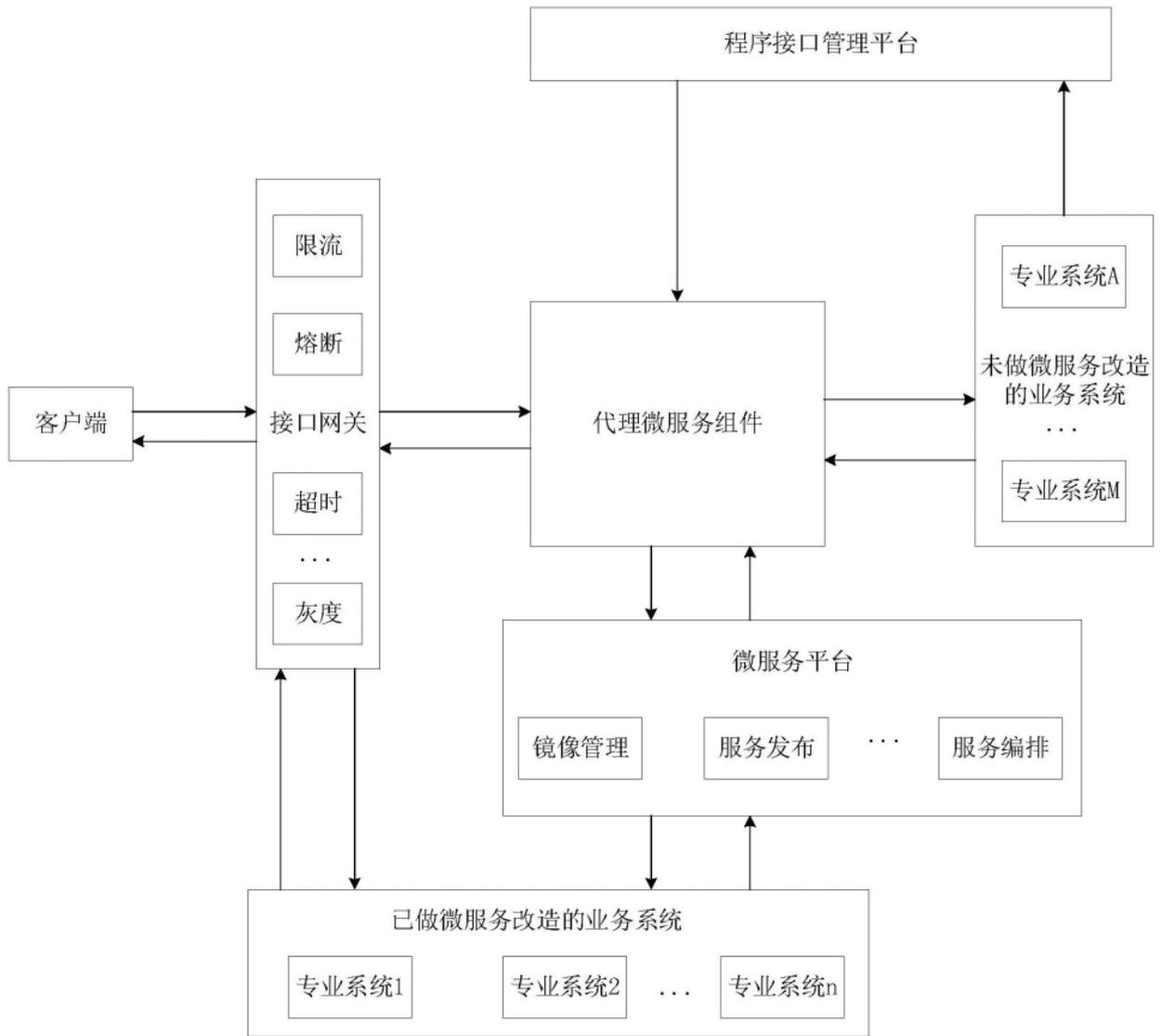


图4

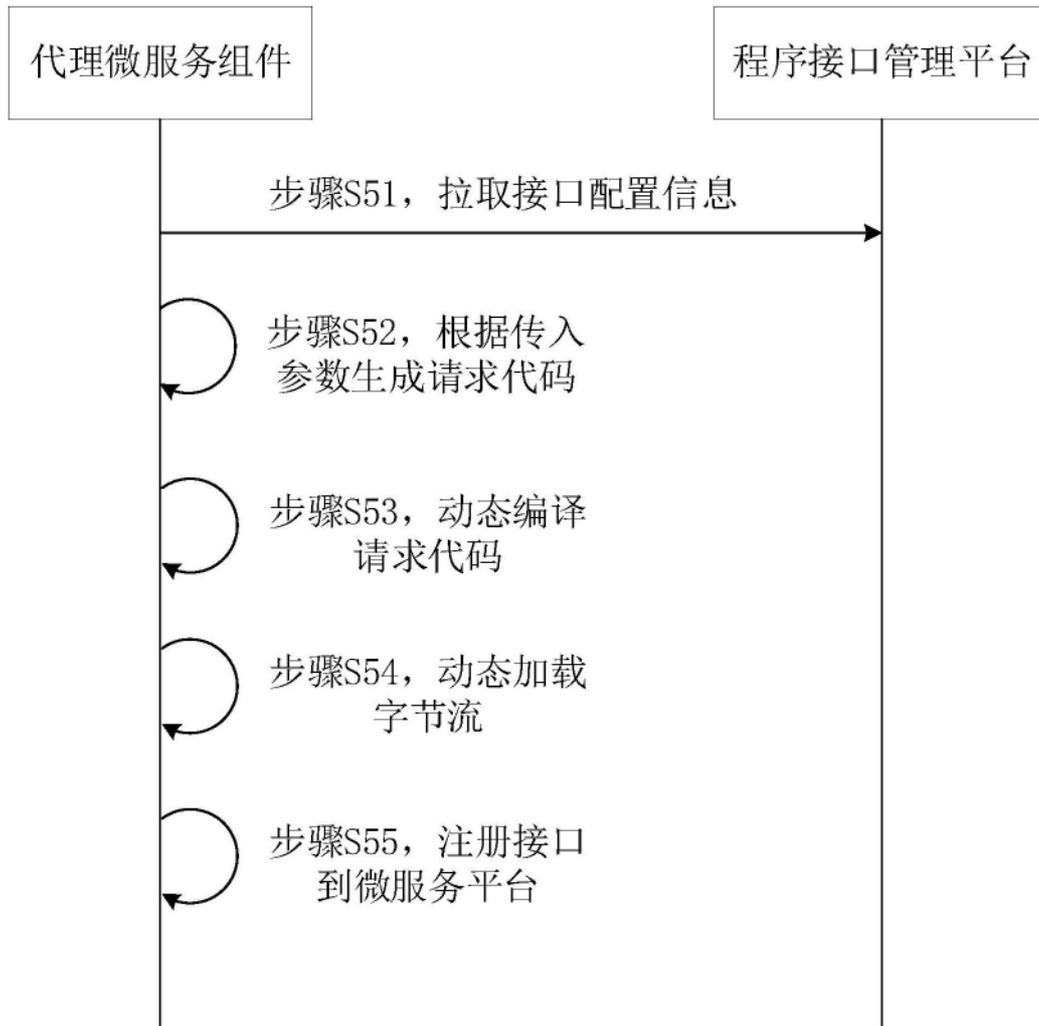


图5

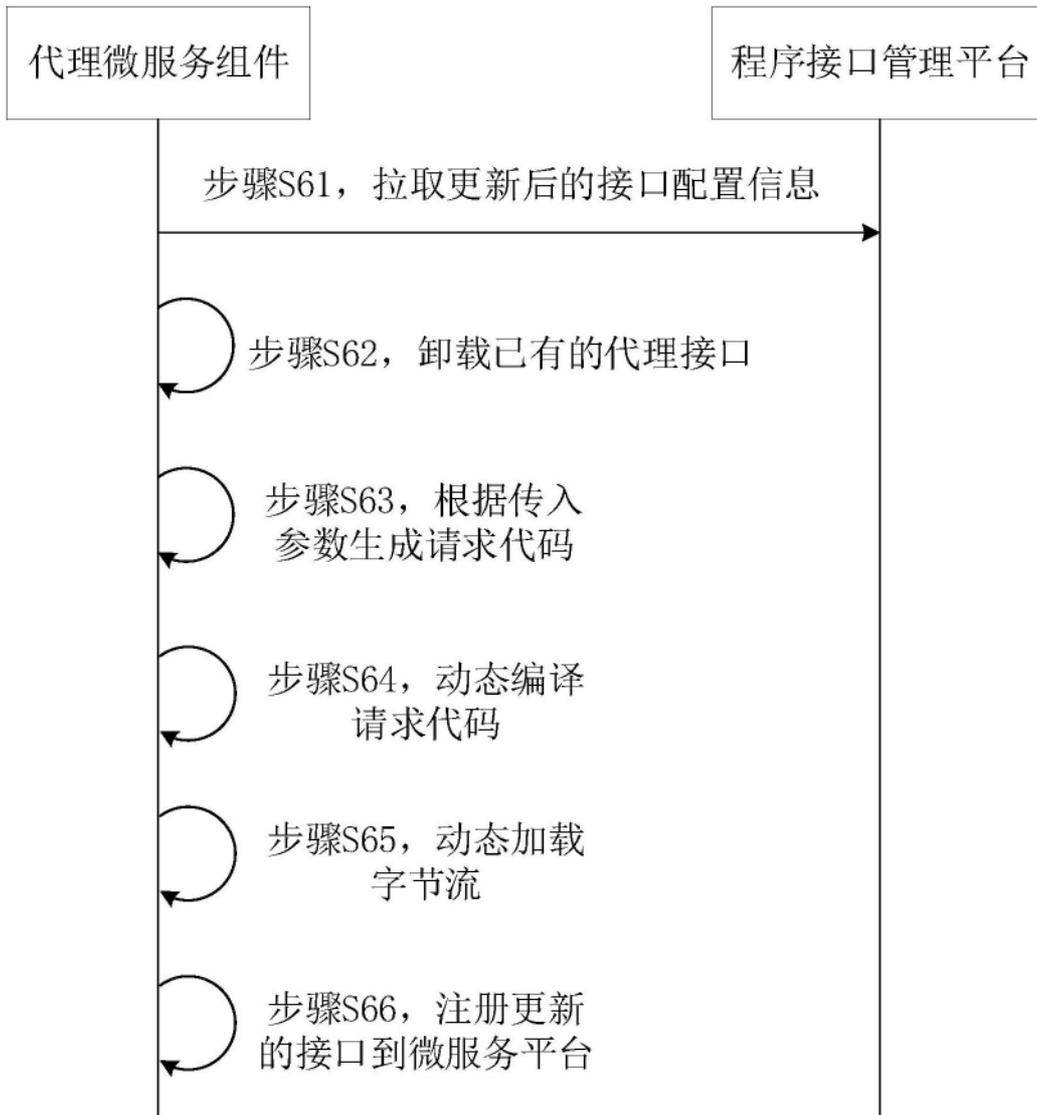


图6

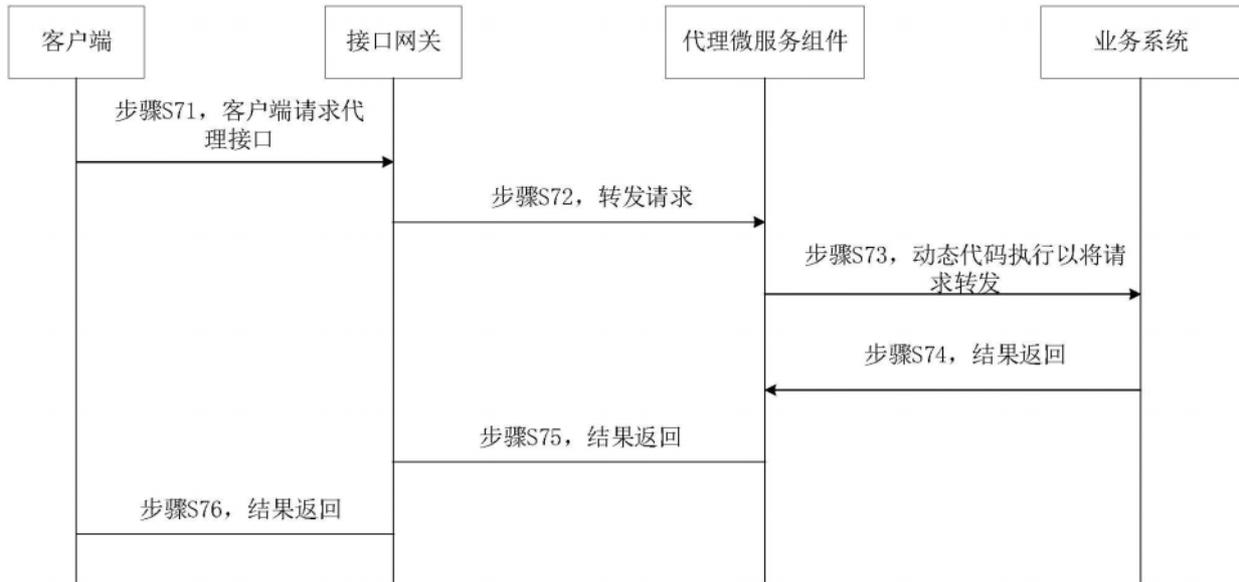


图7

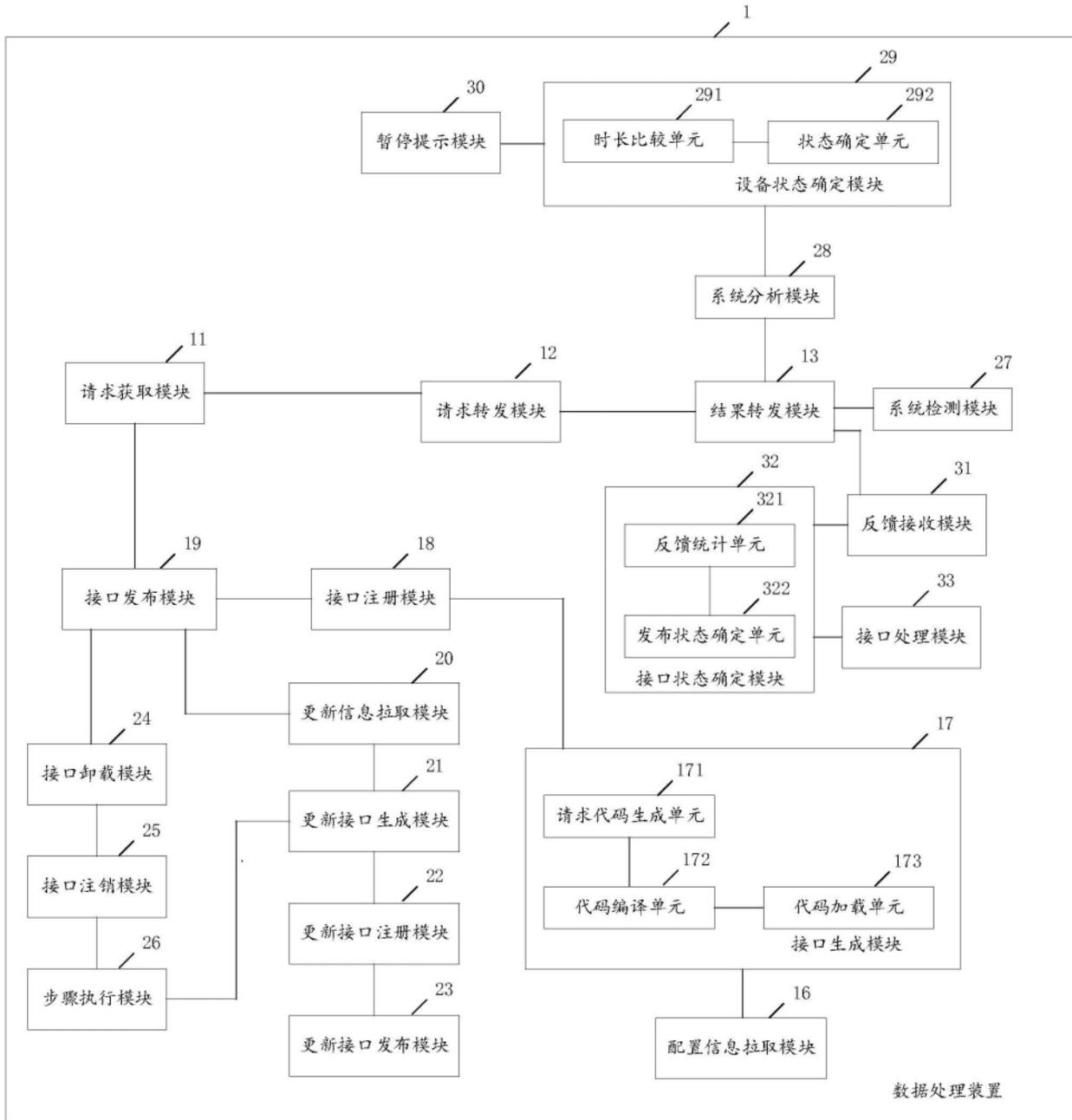


图8

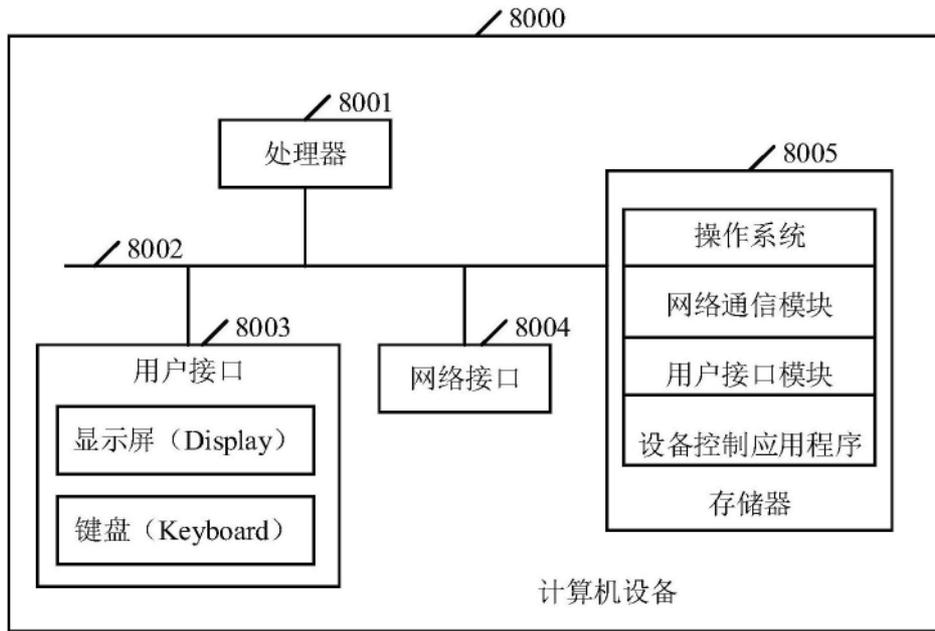


图9