



(12) 发明专利申请

(10) 申请公布号 CN 117793127 A

(43) 申请公布日 2024. 03. 29

(21) 申请号 202311834057.7

G06F 9/30 (2018.01)

(22) 申请日 2023.12.27

(71) 申请人 伟乐视讯科技股份有限公司

地址 516006 广东省惠州市仲恺高新区惠南产业园顺昌路1号

(72) 发明人 潘家跃 汤志敏 李伟 邹伟华

(74) 专利代理机构 广州粤高专利商标代理有限公司 44102

专利代理师 练逸夫

(51) Int. Cl.

H04L 67/1087 (2022.01)

H04L 67/146 (2022.01)

H04L 67/2866 (2022.01)

H04L 67/00 (2022.01)

H04L 9/40 (2022.01)

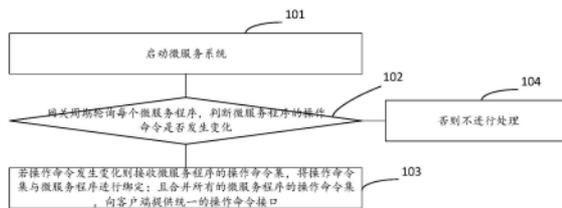
权利要求书2页 说明书8页 附图2页

(54) 发明名称

操作命令生成方法、微服务架构通讯方法及微服务系统

(57) 摘要

本发明实施例涉及通讯技术领域,具体涉及操作命令生成方法、微服务架构通讯方法及微服务系统,实现以上技术效果:网关周期轮询各个微服务程序的通讯接口标识符,确定各个微服务程序的操作命令是否有变更,如有变更,即时更新操作命令。并在网关微服务中保存了操作命令和对应微服务的通讯地址。且在微服务系统中使用网关作为所有的消息入口,统一了与客户端的通讯地址;整个服务端向外提供了统一的通讯地址,提供了统一的命令接口。客户端不用变更通讯地址,隐藏了命令和微服务之间的关系,整个服务端就像只有一个微服务。提高了系统的安全性。网关实时自动更新每个微服务操作命令。方便业务微服务的升级和管理。



1. 一种操作命令生成方法,其特征在于,应用于设有网关、及多个微服务程序的微服务系统中,所述方法包括:

启动所述微服务系统;

所述网关周期轮询每个微服务程序,判断所述微服务程序的操作命令是否发生变化;

若操作命令发生变化则接收所述微服务程序的操作命令集,将所述操作命令集与微服务程序进行绑定;且合并所有的微服务程序的操作命令集,向客户端提供统一的操作命令接口;

否则不进行处理。

2. 根据权利要求1所述的一种操作命令生成方法,其特征在于,所述网关周期轮询每个微服务程序,判断所述微服务程序的操作命令是否发生变化,具体包括:

所述网关启动;

所述网关向各个所述微服务程序发送通信接口标识符获取指令,并从所述微服务程序获取通信接口标识符;

解析所述通信接口标识符,判断通信接口标识符是否发生变化。

3. 根据权利要求2所述的一种操作命令生成方法,其特征在于,所述网关轮询每个微服务程序的周期为1秒。

4. 根据权利要求2所述的一种操作命令生成方法,其特征在于,所述微服务程序生成操作命令集,具体包括:

所述微服务程序启动;

生成所述微服务程序的操作命令集,并根据设备特征码和系统时间生成通讯接口标识符;

若所述微服务程序接收到通信接口标识符获取指令,则向网关返回通信接口标识符;

若所述微服务程序接收到获取通讯接口请求,则向网关返回通讯接口内容。

5. 根据权利要求1所述的一种操作命令生成方法,其特征在于,将所述操作命令集与微服务程序进行绑定,包括:

所述网关通过LoadedEndpoint数组,保存微服务程序的通信地址、通信接口标识符、操作命令集。

6. 根据权利要求1所述的一种操作命令生成方法,其特征在于,所述合并所有的微服务程序的操作命令集,包括:

定义所述操作命令集格式,并将所述微服务程序的操作命令集通过Query类型和Mutation类型进行通讯格式合并。

7. 根据权利要求6所述的一种操作命令生成方法,其特征在于,所述操作命令集通过SDL语言实现,且所述操作命令集的格式:sdl:schema {query:Query;mutation:Mutation},其中,query为查询模式,mutation为修改模式。

8. 一种微服务架构通讯方法,其特征在于,所述方法包括:

所述网关接收并解析客户端的请求信息,获取微服务程序的通信地址,并根据所述通信地址将请求信息转发到微服务程序中,其中,所述请求信息包括权利要求1-7任一项所述一种操作命令生成方法中的任一操作命令。

9. 根据权利要求8所述的一种微服务架构通讯方法,其特征在于,当所述微服务程序根

据所述请求信息执行操作命令后,将执行结果发送到所述网关中,再通过所述网关发送到客户端。

10. 一种微服务系统,其特征在于,包括:

所述客户端,用于向所述服务端发送请求信息;

所述服务端,包括网关、以及多个微服务程序,用于接收并解析客户端的请求信息,获取微服务程序的通信地址,并根据所述通信地址将请求信息转发到微服务程序中,其中,所述请求信息包括权利要求1-7任一项所述一种操作命令生成方法中的任一操作命令。

## 操作命令生成方法、微服务架构通讯方法及微服务系统

### 技术领域

[0001] 本发明实施例涉及通讯技术领域,具体涉及操作命令生成方法、微服务架构通讯方法及微服务系统。

### 背景技术

[0002] 微服务架构(Microservice Architecture)是一种架构概念,旨在通过将功能分解到各个离散的服务中以实现了解决方案的解耦。微服务架构把一个大型的单个应用程序和服务拆分为数个甚至数十个的支持微服务,它可扩展单个组件而不是整个的应用程序堆栈,从而满足服务等级协议。围绕业务领域组件来创建应用,这些应用可独立地进行开发、管理和迭代。在分散的组件中使用云架构和平台式部署、管理和服务功能,使产品交付变得更加简单。

[0003] 在现代微服务架构中,后端的多个微服务程序如何与前端进行通讯。其实现方式有多种多样,大致划分为如下三种:

[0004] (1) 多个微服务通过不同IP与前端通讯;

[0005] (2) 多个微服务通过相同的IP和不同的端口与前端通讯;

[0006] (3) 多个微服务通过网关,使用相同的IP和不同的下级路径与前端通讯。

[0007] 通过网关根据下级路径的不同,自动转发到对应的微服务中。不同的微服务,使用了不同的本地IP地址。

[0008] 但上述方法,在前端来看,事实上都是不同的网址路径。如果后端相关接口变更时,前端也需要同步做网址变更。另外每个微服务的通讯接口是独立的,无法向前端提供统一的通讯接口。同时容易向外暴露了后端微服务的工作情况,造成了信息安全隐患。

### 发明内容

[0009] 鉴于上述问题,本发明实施例提供了操作命令生成方法、微服务架构通讯方法及微服务系统,解决了微服务架构一致性差、信息安全差的问题。

[0010] 根据本发明实施例的一个方面,提供了一种操作命令生成方法,应用于设有网关、及多个微服务程序的微服务系统中,所述方法包括:

[0011] 启动所述微服务系统;

[0012] 所述网关周期轮询每个微服务程序,判断所述微服务程序的操作命令是否发生变化;

[0013] 若操作命令发生变化则接收所述微服务程序的操作命令集,将所述操作命令集与微服务程序进行绑定;且合并所有的微服务程序的操作命令集,向客户端提供统一的操作命令接口;

[0014] 否则不进行处理。

[0015] 在一种可选的方式中,所述网关周期轮询每个微服务程序,判断所述微服务程序的操作命令是否发生变化,具体包括:

- [0016] 所述网关启动；
- [0017] 所述网关向各个所述微服务程序发送通信接口标识符获取指令,并从所述微服务程序获取通信接口标识符；
- [0018] 解析所述通信接口标识符,判断通信接口标识符是否发生变化。
- [0019] 在一种可选的方式中,所述网关轮询每个微服务程序的周期为1秒。
- [0020] 在一种可选的方式中,所述微服务程序生成操作命令集,具体包括:
- [0021] 所述微服务程序启动；
- [0022] 生成所述微服务程序的操作命令集,并根据设备特征码和系统时间生成通讯接口标识符；
- [0023] 若所述微服务程序接收到通信接口标识符获取指令,则向网关返回通信接口标识符；
- [0024] 若所述微服务程序接收到获取通讯接口请求,则向网关返回通讯接口内容。
- [0025] 在一种可选的方式中,将所述操作命令集与微服务程序进行绑定,包括:
- [0026] 所述网关通过LoadedEndpoint数组,保存微服务程序的通信地址、通信接口标识符、操作命令集。
- [0027] 在一种可选的方式中,所述合并所有的微服务程序的操作命令集,包括:
- [0028] 定义所述操作命令集格式,并将所述微服务程序的操作命令集通过Query类型和Mutation类型进行通讯格式合并。
- [0029] 在一种可选的方式中,所述操作命令集通过SDL语言实现,且所述操作命令集的格式:sdl:schema {query:Query;mutation:Mutation},其中,query为查询模式,mutation为修改模式。
- [0030] 根据本发明实施例的另一方面,提供了一种微服务架构通讯方法,所述方法包括:
- [0031] 所述网关接收并解析客户端的请求信息,获取微服务程序的通信地址,并根据所述通信地址将请求信息转发到微服务程序中,其中,所述请求信息包括上述一种操作命令生成方法中的任一操作命令。
- [0032] 在一种可选的方式中,当所述微服务程序根据所述请求信息执行操作命令后,将执行结果发送到所述网关中,再通过所述网关发送到客户端。
- [0033] 根据本发明实施例的又一方面,提供了一种微服务系统,包括:
- [0034] 所述客户端,用于向所述服务端发送请求信息；
- [0035] 所述服务端,包括网关、以及多个微服务程序,用于接收并解析客户端的请求信息,获取微服务程序的通信地址,并根据所述通信地址将请求信息转发到微服务程序中,其中,所述请求信息包括上述一种操作命令生成方法中的任一操作命令。
- [0036] 本发明实施例设计了操作命令生成方法、微服务架构通讯方法及微服务系统,实现以上技术效果:网关周期轮询各个微服务程序的通讯接口标识符,确定各个微服务程序的操作命令是否有变更,如有变更,即时更新操作命令。并在网关微服务中保存了操作命令和对应微服务的通讯地址。且在微服务系统中使用网关作为所有的消息入口,统一了与客户端的通讯地址;整个服务端向外提供了统一的通讯地址,提供了统一的命令接口。客户端不用变更通讯地址。隐藏了命令和微服务之间的关系,整个服务端就像只有一个微服务。提高了系统的安全性。网关实时自动更新每个微服务操作命令。方便业务微服务的升级和管

理。

[0037] 上述说明仅是本发明实施例技术方案的概述,为了能够更清楚了解本发明实施例的技术手段,而可依照说明书的内容予以实施,并且为了让本发明实施例的上述和其它目的、特征和优点能够更明显易懂,以下特举本发明的具体实施方式。

### 附图说明

[0038] 附图仅用于示出实施方式,而并不认为是对本发明的限制。而且在整个附图中,用相同的参考符号表示相同的部件。在附图中:

[0039] 图1示出了本发明提供的操作命令生成方法的流程示意图;

[0040] 图2示出了本发明提供的判断微服务程序的操作命令是否发生变化的流程示意图;

[0041] 图3示出了本发明提供的微服务程序生成操作命令集的流程示意图;

[0042] 图4示出了本发明提供的微服务系统的结构示意图。

### 具体实施方式

[0043] 下面将参照附图更详细地描述本发明的示例性实施例。虽然附图中显示了本发明的示例性实施例,然而应当理解,可以以各种形式实现本发明而不应被这里阐述的实施例所限制。

[0044] 实施例1,

[0045] 图1示出了本发明一种操作命令生成方法,应用于设有网关、及多个微服务程序的微服务系统中,本方法包括:

[0046] 101,启动微服务系统;在步骤101中,启动微服务系统,即将网关、微服务程序进行启动。

[0047] 102,网关周期轮询每个微服务程序,判断微服务程序的操作命令是否发生变化;在步骤102中,网关周期轮询每个微服务程序,判断微服务程序的操作命令是否发生变化,具体包括:网关启动;网关向各个微服务程序发送通信接口标识符获取指令,并从微服务程序获取通信接口标识符;解析通信接口标识符,判断通信接口标识符是否发生变化。

[0048] 103,若操作命令发生变化则接收微服务程序的操作命令集,将操作命令集与微服务程序进行绑定;且合并所有的微服务程序的操作命令集,向客户端提供统一的操作命令接口;在步骤103中,微服务程序生成操作命令集,具体包括:微服务程序启动;生成微服务程序的操作命令集,并根据设备特征码和系统时间生成通讯接口标识符;若微服务程序接收到通信接口标识符获取指令,则向网关返回通信接口标识符;若微服务程序接收到获取通讯接口请求,则向网关返回通讯接口内容。将操作命令集与微服务程序进行绑定,包括:网关通过LoadedEndpoint数组,保存微服务程序的通信地址、通信接口标识符、操作命令集。合并所有的微服务程序的操作命令集,包括:定义操作命令集格式,并将微服务程序的操作命令集通过Query类型和Mutation类型进行通讯格式合并。操作命令集通过SDL语言实现,且操作命令集的格式:sdl:schema{query:Query;mutation:Mutation},其中,query为查询模式,mutation为修改模式。

[0049] 104,否则不进行处理。

[0050] 在本实施例中,网关周期轮询各个微服务程序的通讯接口标识符,确定各个微服务程序的操作命令是否有变更,如有变更,即时更新操作命令。并在网关微服务中保存了操作命令和对应微服务的通讯地址。且在微服务系统中使用网关作为所有的消息入口,统一了与客户端的通讯地址;整个服务端向外提供了统一的通讯地址,提供了统一的命令接口。客户端不用变更通讯地址。隐藏了命令和微服务之间的关系,整个服务端就像只有一个微服务。提高了系统的安全性。网关实时自动更新每个微服务操作命令。方便业务微服务的升级和管理。

[0051] 在一种可选的方式中,参见图2,网关周期轮询每个微服务程序,判断微服务程序的操作命令是否发生变化,具体包括:

[0052] 201,网关启动;

[0053] 202,网关向各个微服务程序发送通信接口标识符获取指令,并从微服务程序获取通信接口标识符;

[0054] 203,解析通信接口标识符,判断通信接口标识符是否发生变化。

[0055] 在本实施例中,网关微服务的处置流程如下,网关微服务在启动时或者启动后周期轮询各个微服务,确定相关微服务的操作命令集是否有了改变,如果有改变,那么就更新相关操作命令集。通信接口标识符包括设备特征码和系统时间;如当前的微服务程序的通讯接口标识与上一次系统启动的微服务程序的通讯接口标识不同,则说明操作命令集发生变化。在上述实施例中,网关轮询每个微服务程序的周期为1秒。

[0056] 在一种可选的方式中,参见图3,微服务程序生成操作命令集,具体包括:

[0057] 301,微服务程序启动;

[0058] 302,生成微服务程序的操作命令集,并根据设备特征码和系统时间生成通信接口标识符;

[0059] 303,若微服务程序接收到通信接口标识符获取指令,则向网关返回通信接口标识符;

[0060] 304,若微服务程序接收到获取通讯接口请求,则向网关返回通讯接口内容。

[0061] 在本实施例中,业务微服务程序在每次启动时,会根据生成自身的操作命令集,并且更新自身的通信接口标识符。收到网关发过来的标识符或者操作命令请求时,就相应地回传相应的内容。

[0062] 在一种可选的方式中,将操作命令集与微服务程序进行绑定,包括:网关通过LoadedEndpoint数组,保存微服务程序的通信地址、通信接口标识符、操作命令集。具体地,网关会保存所有微服务以下三种信息:

```
LoadedEndpoint {  
    url: http://ferris-auth:8003/graphql,           // 微服务通信地址  
    sdlId: bb8a212aa3afb397014223e5d2650944, // 通信接口标识符  
[0063]    sdl: schema {                                   // 操作命令集  
        query: Query  
        mutation: Mutation}};
```

[0064] 通过LoadedEndpoint数组可以将操作命令与微服务程序进行绑定。在客户端发生请求信号,通过相应的操作指令,可以直接查询到相应的微服务程序。

[0065] 在一种可选的方式中,合并所有的微服务程序的操作命令集,包括:

[0066] 定义操作命令集格式,并将微服务程序的操作命令集通过Query类型和Mutation类型进行通讯格式合并。其中,操作命令集通过SDL语言实现,且操作命令集的格式:sdl:schema {query:Query;mutation:Mutation},其中,query为查询模式,mutation为修改模式,修改模式包括增加、删除、修改等操作。

[0067] 具体地,定义了用户管理微服务的操作命令接口。包括支持哪些查询和哪些修改操作,以及相关操作需要用到的自定义类成员结构类型。

[0068] A、定义了UserPermission,UserGroup,User三种数据类型通过以下方式进行定义:

```

type UserPermission {
    id: ID!
    permission: String!}

```

```

type UserGroup {
    id: ID!
    permissions: [String!]
    groupName: String! }

```

[0069]

```

type User {
    id: ID!
    password: String
    email: String
    groups: [UserGroup!]
    name: String! }

```

[0070] B、定义查询模式的内容,支持查询user,getUserGroup,和userPermission等三种对象。

```
[0071] type Query {
```

```
[0072]   user(name:String!):User!//输入参数为字符串name,返回User类对象。
```

```
[0073]   getUserGroup(name:String):UserGroup//输入参数为字符串name,返回UserGroup类对象。
```

```
[0074]   userPermission:UserPermission//无输入参数,返回UserPermission类对象。}
```

[0075] C、定义修改模式的内容,支持创建用户createUser,deleteUser,addUserGroup,deleteUserGroup,createUserPermission和deleteUserPermission等六种操作。

```
[0076] type Mutation{
```

```
[0077]   createUser(email:String,password:String,name:String!):User!
```

```
[0078]   deleteUser(name:String!):User!//输入用户名字符串,返回User类对象
```

```
[0079]   addUserGroup(permissions:[String!],name:String!):UserGroup!
```

```
[0080]   deleteUserGroup(name:String!):UserGroup!
```

```
[0081]   createUserPermission(name:String!):UserPermission!
```

```
[0082]   deleteUserPermission(name:String!):UserPermission!}。
```

[0083] 每个微服务使用的都是上述格式进行定义,分别定义了Query和Mutation类型。由于不同的微服务的业务功能不一致,向对应的查询命令和修改命令都是不一样的。网关微服务从各个微服务获取到所有的操作命令也就是不一样的。如果相关接口都合理定义,合

并起来不会产生冲突。

[0084] 用户管理微服务和生产微服务的通讯格式合并结果如下：

```

    type Query {
        user(name: String!): User!           //用户管理微服务
        getUserGroup(name: String): UserGroup //用户管理微服务
    [0085] userPermission: UserPermission     //用户管理微服务
        product(type: String!): Product[]!   //产品微服务
    }
    type Mutation {
        createUser(email: String, password: String, name: String!): User!
        deleteUser(name: String!): User!
        addUserGroup(permissions: [String!], name: String!): UserGroup!
        deleteUserGroup(name: String!): UserGroup!
    [0086] createUserPermission(name: String!): UserPermission!
        deleteUserPermission(name: String!): UserPermission! //用户管理微服
        createProduct(type: String, weight: Number): Product! //产品微服务
        deleteProduct(id: ID): boolean! //产品微服务 }
  
```

[0087] 这个合并后的操作命令集会向客户端统一提供。

[0088] 实施例2，

[0089] 根据本发明实施例的另一方面，提供了一种微服务架构通讯方法，方法包括：网关接收并解析客户端的请求信息，获取微服务程序的通信地址，并根据通信地址将请求信息转发到微服务程序中，其中，请求信息包括实施例1所述一种操作命令生成方法中的任一操作命令。当网关收到前端发来的请求时，会分析请求内容中的命令，然后找出微服务通讯地址，然后把相关请求转发到对应微服务中。请求信息中的操作命令为实施例1的操作命令集任一条。当微服务程序根据请求信息执行操作命令后，将执行结果发送到网关中，再通过网关发送到客户端。

[0090] 实施例3，

[0091] 根据本发明实施例的又一方面，提供了一种微服务系统，参见图4，包括：客户端，用于向服务端发送请求信息；服务端，包括网关、以及多个微服务程序，用于接收并解析客户端的请求信息，获取微服务程序的通信地址，并根据通信地址将请求信息转发到微服务程序中，其中，请求信息包括上述一种操作命令生成方法中的任一操作命令。本发明实施例设计了操作命令生成方法、微服务架构通讯方法及微服务系统，实现以上技术效果：网关周

期轮询各个微服务程序的通讯接口标识符,确定各个微服务程序的操作命令是否有变更,如有变更,即时更新操作命令。并在网关微服务中保存了操作命令和对应微服务的通讯地址。且在微服务系统中使用网关作为所有的消息入口,统一了与客户端的通讯地址;整个服务端向外提供了统一的通讯地址,提供了统一的命令接口。客户端不用变更通讯地址。隐藏了命令和微服务之间的关系,整个服务端就像只有一个微服务。提高了系统的安全性。网关实时自动更新每个微服务操作命令。方便业务微服务的升级和管理。

[0092] 在此提供的算法或显示不与任何特定计算机、虚拟系统或者其它设备固有相关。此外,本发明实施例也不针对任何特定编程语言。

[0093] 在此处所提供的说明书中,说明了大量具体细节。然而能够理解,本发明的实施例可以在没有这些具体细节的情况下实践。类似地,为了精简本发明并帮助理解各个发明方面中的一个或多个,在上面对本发明的示例性实施例的描述中,本发明实施例的各个特征有时被一起分组到单个实施例、图、或者对其的描述中。其中,遵循具体实施方式的权利要求书由此明确地并入该具体实施方式,其中每个权利要求本身都作为本发明的单独实施例。

[0094] 本领域技术人员可以理解,可以对实施例中的设备中的模块进行自适应性地改变并且把它们设置在与该实施例不同的一个或多个设备中。可以把实施例中的模块或单元或组件组合成一个模块或单元或组件,以及此外可以把它分成多个子模块或子单元或子组件。除了这样的特征和/或过程或者单元中的至少一些是相互排斥之外。

[0095] 应该注意的是上述实施例对本发明进行说明而不是对本发明进行限制,并且本领域技术人员在不脱离所附权利要求的范围的情况下可设计出替换实施例。在权利要求中,不应将位于括号之间的任何参考符号构造成对权利要求的限制。单词“包含”不排除存在未列在权利要求中的元件或步骤。位于元件之前的单词“一”或“一个”不排除存在多个这样的元件。本发明可以借助于包括有若干不同元件的硬件以及借助于适当编程的计算机来实现。在列举了若干装置的单元权利要求中,这些装置中的若干个可以是通过同一个硬件项来具体体现。单词第一、第二、以及第三等的使用不表示任何顺序。可将这些单词解释为名称。上述实施例中的步骤,除有特殊说明外,不应理解为对执行顺序的限定。

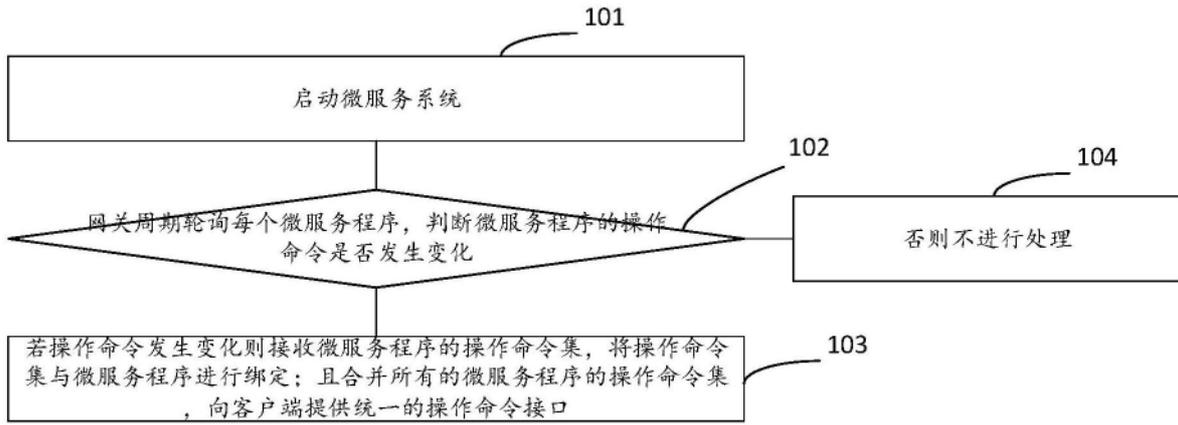


图1

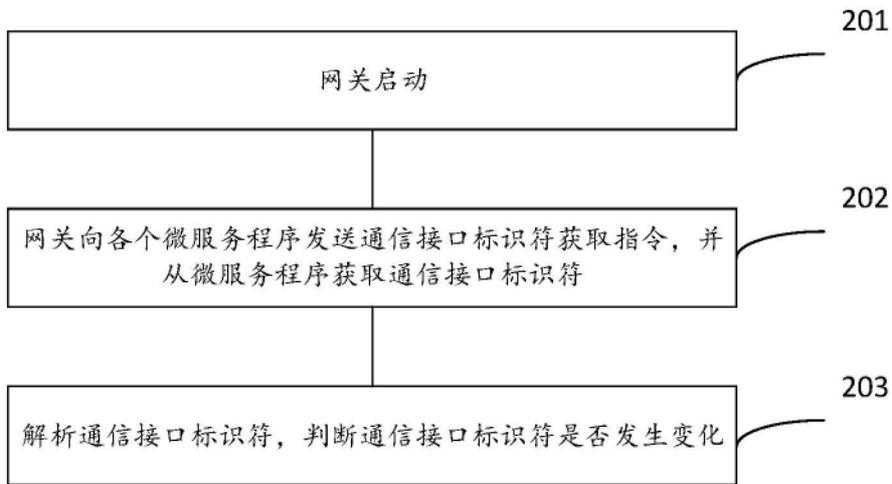


图2

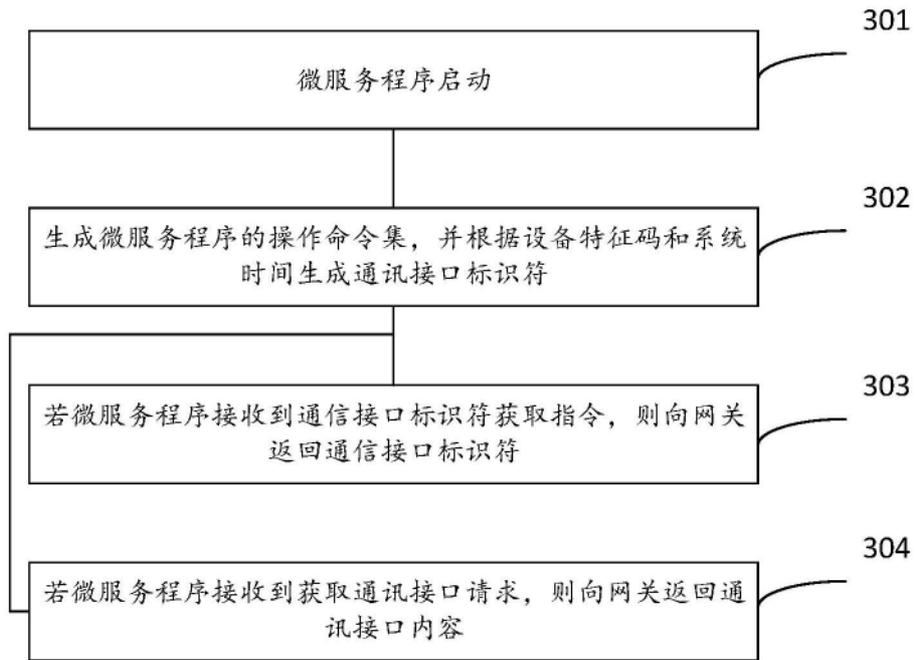


图3

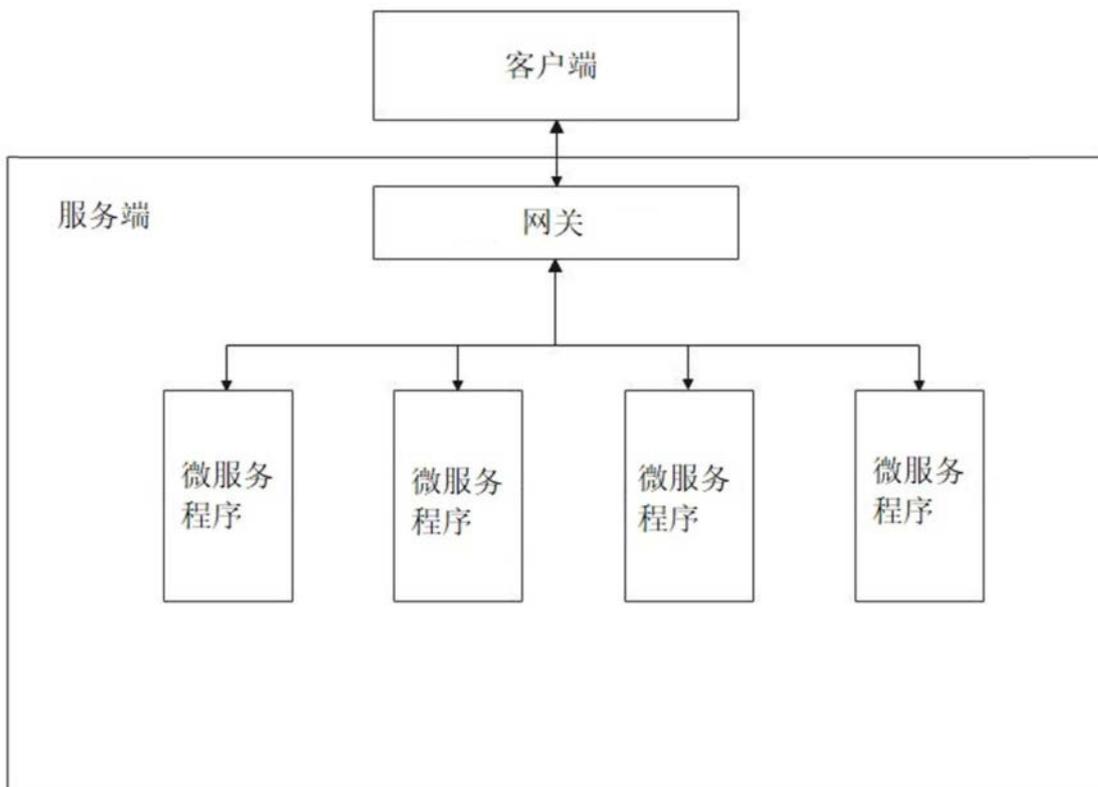


图4