



(12)发明专利申请

(10)申请公布号 CN 111355784 A

(43)申请公布日 2020.06.30

(21)申请号 202010104015.8

(22)申请日 2020.02.20

(71)申请人 北京字节跳动网络技术有限公司  
地址 100041 北京市石景山区实兴大街30  
号院3号楼2层B-0035房间

(72)发明人 邬亮

(74)专利代理机构 北京睿驰通程知识产权代理  
事务所(普通合伙) 11604  
代理人 唐华 张文平

(51)Int.Cl.

H04L 29/08(2006.01)

H04L 12/66(2006.01)

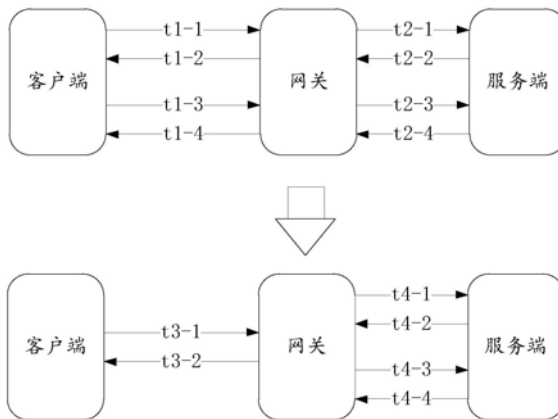
权利要求书2页 说明书11页 附图6页

(54)发明名称

一种处理请求信息的方法、装置、介质和电子设备

(57)摘要

本公开提供了一种处理请求信息的方法、装置、介质和电子设备。所述方法包括：获取多个第一请求信息；所述第一请求信息间包括相互关联的第一请求关系；基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息，生成第二请求信息；根据所述第二请求信息生成第二请求消息，并向服务端的应用程序接口网关发送所述第二请求消息。在一次信息交互过程中将发送的多个请求消息减少到一个，有效的减少了耗时，提升了用户体验；且减少了API网关负载，提升API网关吞吐量。



1. 一种处理请求信息的方法,应用于客户端,其特征在于,包括:  
获取多个第一请求信息;所述第一请求信息间包括相互关联的第一请求关系;  
基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息;  
根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。
2. 根据权利要求1所述的方法,其特征在于,所述基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息,包括:  
根据所述第一请求数据合并规则合并所述第一请求信息,生成第一合并请求信息;  
根据所述第一请求关系描述规则合并所述第一请求关系,生成第一合并请求关系信息;  
合并所述第一合并请求信息和所述第一合并请求关系信息,生成第二请求信息。
3. 根据权利要求2所述的方法,其特征在于,所述第一请求信息包括第一请求映射数据结构描述的信息;  
所述根据所述第一请求数据合并规则合并所述第一请求信息,生成第一合并请求信息,包括:  
根据所述第一请求数据合并规则合并所述第一请求映射数据结构,生成第一合并请求信息;所述第一合并请求信息包括多个串行排列的第一请求映射数据结构描述的信息。
4. 根据权利要求3所述的方法,其特征在于,所述第一请求关系包括第一请求关系结构体描述的信息;  
所述根据所述第一请求关系描述规则合并所述第一请求关系,生成第一合并请求关系信息,包括:  
根据所述第一请求关系描述规则合并所述第一请求关系结构体,生成第一合并请求关系信息;所述第一合并请求关系信息包括多个串行排列的第一请求关系结构体描述的信息。
5. 根据权利要求4所述的方法,其特征在于,所述第二请求信息包括一个第二请求合并映射数据结构描述的信息;  
所述第二请求合并映射数据结构包括:所述第一请求映射数据结构和第一请求映射数据结构的名称,以及所述第一请求关系结构体和第一请求关系结构体的名称。
6. 一种处理请求信息的装置,应用于客户端,其特征在于,包括:  
获取第一请求信息单元,用于获取多个第一请求信息;所述第一请求信息间包括相互关联的第一请求关系;  
生成第二请求信息单元,用于基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息;  
生成第二请求消息单元,用于根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。
7. 一种处理请求信息的方法,应用于服务端的应用程序接口网关,其特征在于,包括:  
接收第二请求消息;  
基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息,获取多

个第一请求信息和多个第一请求关系；

分析每个第一请求信息，获取对应的应用程序微服务；

依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

8. 根据权利要求7所述的方法，其特征在于，所述方法还包括：

依次获取响应所述第一请求信息生成的第一响应信息；所述第一响应信息间包括相互关联的第一响应关系；

基于所述第一响应数据合并规则和所述第一响应关系描述规则合并所述第一响应信息，生成第二响应信息；

根据所述第二响应信息生成第二响应消息，并向客户端返回所述第二响应消息。

9. 一种处理请求信息的装置，应用于服务端的应用程序接口网关，其特征在于，包括：

接收第二请求消息单元，用于接收第二请求消息；

分析第二请求消息单元，用于基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息，获取多个第一请求信息和多个第一请求关系；

获取应用程序微服务单元，用于分析每个第一请求信息，获取对应的应用程序微服务；

发送第一请求信息单元，用于依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

10. 一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述程序被处理器执行时实现如权利要求1至5、7至8中任一项所述的方法。

11. 一种电子设备，其特征在于，包括：

一个或多个处理器；

存储装置，用于存储一个或多个程序，当所述一个或多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器实现如权利要求1至5、7至8中任一项所述的方法。

## 一种处理请求信息的方法、装置、介质和电子设备

### 技术领域

[0001] 本公开涉及计算机技术领域,具体而言,涉及一种处理请求信息的方法、装置、介质和电子设备。

### 背景技术

[0002] 应用程序一般由客户端和服务端两部分组成。客户端向服务端发送请求;服务端接收请求后从数据库获取数据,并将数据进行处理后返回给客户端;客户端根据返回的数据更新用户界面。

[0003] 当前,客户端功能变得越来越复杂,一个用户界面可能包含多个模块。为了获取多个模块的数据,在一次信息交互过程中,客户端需要向服务端发送多请求消息。但是,客户端对并发请求消息的数量是有限制的。当同时发送请求消息的数目超过限制的数量时,超出的请求消息就会被暂时阻塞,等已经发送的请求消息返回之后才会被发送。另外,多个请求消息之间可能存在依赖关系。当一个请求消息的请求数据依赖于另一个请求消息的响应数据时,需要等另外一个请求返回之后才能发送。也就是说,存在依赖关系的请求消息必需串行发送。对于超出并发请求消息数量限制或存在依赖关系的多个请求消息,需要耗费多个请求时间,相应的用户界面刷新也需要等待多个请求时间。如果首屏渲染时发送的多个请求消息超出并发请求数量限制或存在依赖关系,会造成比较长的白屏时间,严重损害用户体验。而服务端一般采用微服务架构,即服务端的服务由多个功能单一的微服务组成。多个微服务通过一个应用程序接口(英文全称Application Programming Interface,简称API)网关对客户端提供服务。

[0004] 客户端发送的请求消息先经过API网关,API网关再分发给各个微服务。由于当前的硬件功能非常强,数据库的读写和业务逻辑运算耗时非常少。而且API网关和微服务通常部署在同一个机房或比较近的多个机房,从API网关到微服务的请求时间也非常短。因此,请求消息的耗时瓶颈主要集中在客户端与API网关之间的信息传输过程。

### 发明内容

[0005] 提供该发明内容部分以便以简要的形式介绍构思,这些构思将在后面的具体实施方式部分被详细描述。该发明内容部分并不旨在标识要求保护的技术方案的关键特征或必要特征,也不旨在用于限制所要求的保护的技术方案的范围。

[0006] 本公开的目的在于提供一种处理请求信息的方法、装置、介质和电子设备,能够解决上述提到的至少一个技术问题。具体方案如下:

[0007] 根据本公开的具体实施方式,第一方面,本公开提供一种处理请求信息的方法,应用于客户端,包括:

[0008] 获取多个第一请求信息;所述第一请求信息间包括相互关联的第一请求关系;

[0009] 基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息;

[0010] 根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。

[0011] 根据本公开的具体实施方式,第二方面,本公开提供一种处理请求信息的装置,应用于客户端,包括:

[0012] 获取第一请求信息单元,用于获取多个第一请求信息;所述第一请求信息间包括相互关联的第一请求关系;

[0013] 生成第二请求信息单元,用于基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息;

[0014] 生成第二请求消息单元,用于根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。

[0015] 根据本公开的具体实施方式,第三方面,本公开提供一种处理请求信息的方法,应用于服务端的应用程序接口网关,包括:

[0016] 接收第二请求消息;

[0017] 基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息,获取多个第一请求信息和多个第一请求关系;

[0018] 分析每个第一请求信息,获取对应的应用程序微服务;

[0019] 依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

[0020] 根据本公开的具体实施方式,第四方面,本公开提供一种处理请求信息的装置,应用于服务端的应用程序接口网关,包括:

[0021] 接收第二请求消息单元,用于接收第二请求消息;

[0022] 分析第二请求消息单元,用于基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息,获取多个第一请求信息和多个第一请求关系;

[0023] 获取应用程序微服务单元,用于分析每个第一请求信息,获取对应的应用程序微服务;

[0024] 发送第一请求信息单元,用于依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

[0025] 根据本公开的具体实施方式,第五方面,本公开提供一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现如第一方面任一项所述处理请求信息的方法。根据本公开的具体实施方式,第六方面,本公开提供一种电子设备,包括:一个或多个处理器;存储装置,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现如第一方面任一项所述处理请求信息的方法。

[0026] 根据本公开的具体实施方式,第七方面,本公开提供一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现如第三方面任一项所述处理请求信息的方法。

[0027] 根据本公开的具体实施方式,第八方面,本公开提供一种电子设备,包括:一个或多个处理器;存储装置,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行时,使得所述一个或多个处理器实现如第三方面任一项所述处理请求信息的方法。

[0028] 本公开实施例的上述方案与现有技术相比,至少具有以下有益效果:

[0029] 本公开提供了一种处理请求信息的方法、装置、介质和电子设备。在一次信息交互过程中将发送的多个请求消息减少到一个,有效的减少了耗时,提升了用户体验;且减少了API网关负载,提升API网关吞吐量。

### 附图说明

[0030] 结合附图并参考以下具体实施方式,本公开各实施例的上述和其他特征、优点及方面将变得更加明显。贯穿附图中,相同或相似的附图标记表示相同或相似的元素。应当理解附图是示意性的,原件和元素不一定按照比例绘制。在附图中:

[0031] 图1示出了根据本公开实施例的处理请求信息的方法的流程图;

[0032] 图2示出了根据本公开实施例的处理请求信息的方法的时间示意图;

[0033] 图3示出了根据本公开实施例的处理请求信息的方法的第一合并请求信息;

[0034] 图4示出了根据本公开实施例的处理请求信息的方法的第一合并请求关系信息;

[0035] 图5示出了根据本公开实施例的处理请求信息的方法的第二请求信息;

[0036] 图6示出了根据本公开实施例的处理请求信息的方法的第一响应结构体;

[0037] 图7示出了根据本公开实施例的处理请求信息的装置的单元框图;

[0038] 图8示出了根据本公开实施例的处理请求信息的方法的流程图;

[0039] 图9示出了根据本公开实施例的处理请求信息的装置的单元框图;

[0040] 图10示出了根据本公开的实施例的电子设备连接结构示意图。

### 具体实施方式

[0041] 下面将参照附图更详细地描述本公开的实施例。虽然附图中显示了本公开的某些实施例,然而应当理解的是,本公开可以通过各种形式来实现,而且不应该被解释为限于这里阐述的实施例,相反提供这些实施例是为了更加透彻和完整地理解本公开。应当理解的是,本公开的附图及实施例仅用于示例性作用,并非用于限制本公开的保护范围。

[0042] 应当理解,本公开的方法实施方式中记载的各个步骤可以按照不同的顺序执行,和/或并行执行。此外,方法实施方式可以包括附加的步骤和/或省略执行示出的步骤。本公开的范围在此方面不受限制。

[0043] 本文使用的术语“包括”及其变形是开放性包括,即“包括但不限于”。术语“基于”是“至少部分地基于”。术语“一个实施例”表示“至少一个实施例”;术语“另一实施例”表示“至少一个另外的实施例”;术语“一些实施例”表示“至少一些实施例”。其他术语的相关定义将在下文描述中给出。

[0044] 需要注意,本公开中提及的“第一”、“第二”等概念仅用于对不同的装置、模块或单元进行区分,并非用于限定这些装置、模块或单元所执行的功能的顺序或者相互依存关系。

[0045] 需要注意,本公开中提及的“一个”、“多个”的修饰是示意性而非限制性的,本领域技术人员应当理解,除非在上下文另有明确指出,否则应该理解为“一个或多个”。

[0046] 本公开实施方式中的多个装置之间所交互的消息或者信息的名称仅用于说明性的目的,而并不是用于对这些消息或信息的范围进行限制。

[0047] 下面结合附图详细说明本公开的可选实施例。

[0048] 对本公开提供的第一实施例,即一种处理请求信息的方法的实施例。所述方法应用于客户端。

[0049] 下面结合图1至图6对本公开实施例进行详细说明。请参见图1所示。

[0050] 步骤S101,获取多个第一请求信息。

[0051] 通常,在一个用户界面可能包含多个模块。为了获取多个模块的数据,在一次信息交互过程中,客户端需要向服务端发送多个请求消息。而每个第一请求消息的内容可能不同。所述第一请求信息间包括相互关联的第一请求关系。例如,串联关系或依赖关系。

[0052] 步骤S102,基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息。

[0053] 为了客户端将一次信息交互过程中多个请求消息合并成一个请求消息,且与服务端相关联的API网关能够解析该请求消息,本公开实施例预先设计了第一请求数据合并规则和第一请求关系描述规则。

[0054] 第一请求数据合并规则用于定义一次信息交互过程中第二请求信息描述多个第一请求信息的请求数据的规则。

[0055] 由于客户端需要向服务端发送多个请求消息时,多个请求消息之间可能存在依赖关系。当一个请求消息的请求数据依赖于另一个请求消息的响应数据时,需要等另外一个请求返回之后才能发送。因此,本公开实施例针对第一请求关系预先设计了第一请求关系描述规则。第一请求关系描述规则用于定义一次信息交互过程中第二请求信息描述多个第一请求关系的规则。

[0056] 所述基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息,包括以下步骤:

[0057] 步骤S102-1,根据所述第一请求数据合并规则合并所述第一请求信息,生成第一合并请求信息。

[0058] 所述第一请求信息包括第一请求映射数据结构描述的信息。

[0059] 第一请求映射数据结构包括第一请求名称和第一请求结构体。

[0060] 第一请求结构体包括第一请求方法和第一请求路径。例如,第一请求名称为“api1”,第一请求结构体为“{method:’POST’,path:’/api\_path’}”。

[0061] 进一步的,所述根据所述第一请求数据合并规则合并所述第一请求信息,生成第一合并请求信息,包括以下步骤:

[0062] 步骤S102-1-1,根据所述第一请求数据合并规则合并所述第一请求映射数据结构,生成第一合并请求信息。

[0063] 请参见图3所示,所述第一合并请求信息包括多个串行排列的第一请求映射数据结构描述的信息。

[0064] 步骤S102-2,根据所述第一请求关系描述规则合并所述第一请求关系,生成第一合并请求关系信息。

[0065] 所述第一请求关系包括第一请求关系结构体描述的信息。

[0066] 第一请求关系结构体,包括:第一请求管道、第一请求信息和第一请求数据映射。

[0067] 第一请求管道是一个用符号“|”表示串行关系且用符号“,”表示并行关系的字符串。管道必须以符号“|”开头和结尾。例如,“pipeline:’|api1|api2|’”表示依次发送请求

api1和api2;“pipeline:’|api1,api2|’”表示同时请求api1和api2;“pipeline:’|api1,api2|api3|’”表示先同时请求api1和api2,然后再请求api3。

[0068] 第一请求数据映射用于描述第一请求信息之间的数据依赖关系的映射数据结构。

[0069] 第一请求数据映射,包括:第一请求映射键值和第一请求映射值。

[0070] 其中,第一请求映射键值为依赖请求的请求数据的依赖字段的访问路径;第一请求映射值为被依赖请求的响应数据的被依赖字段的访问路径,访问路径用符号‘.’表示。例如,“{‘api2.cname’:’api1.name’}”表示发送请求api2时,把请求api1的响应数据中“name”赋值给请求api2的请求数据“cname”。

[0071] 进一步的,所述根据所述第一请求关系描述规则合并所述第一请求关系,生成第一合并请求关系信息,包括以下步骤:

[0072] 步骤S102-2-1,根据所述第一请求关系描述规则合并所述第一请求关系结构体,生成第一合并请求关系信息。

[0073] 请参见图4所示,所述第一合并请求关系信息包括多个串行排列的第一请求关系结构体描述的信息。

[0074] 步骤S102-3,合并所述第一合并请求信息和所述第一合并请求关系信息,生成第二请求信息。

[0075] 所述第二请求信息包括一个第二请求合并映射数据结构描述的信息。

[0076] 所述第二请求合并映射数据结构包括:所述第一请求映射数据结构和第一请求映射数据结构的名称,以及所述第一请求关系结构体和第一请求关系结构体的名称。

[0077] 请参见图5所示,例如,第一请求映射数据结构的名称为“reqDataMap”;第一请求关系结构体的名称为“apiRelation”;第一请求数据映射为“dataMaps”。

[0078] 步骤S103,根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。

[0079] 消息是指软件对象之间进行交互作用和通讯的一种方式。请求消息是发送方为了从响应方获取特定数据而发送的特定消息。通常请求消息是根据应用环境中的通讯协议封装而成。例如,在互联网环境中,根据HTTP协议封装而成。

[0080] 请参见图2所示,现有技术中,完成一次信息交互过程,客户端和网关之间需要消耗的时间是t1-1、t1-2、t1-3和t1-4的和;网关和服务端之间需要消耗的时间是t2-1、t2-2、t2-3和t2-4的和。本公开实施例提供了在一次信息交互过程中将发送的多个请求消息减少到一个,客户端和网关之间需要消耗的时间是t3-1和t3-2的和;网关和服务端之间需要消耗的时间是t4-1、t4-2、t4-3和t4-4的和。在客户端和网关之间有效的减少了耗时,提升了用户体验;且减少了API网关负载,提升API网关吞吐量。

[0081] 与本公开提供的第一实施例相对应,本公开还提供了第二实施例,即一种处理请求信息的装置。由于第二实施例基本相似于第一实施例,所以描述得比较简单,相关的部分请参见第一实施例的对应说明即可。下述描述的装置实施例仅仅是示意性的。

[0082] 图7示出了本公开提供的一种处理请求信息的装置的实施例。图7为本公开实施例提供的处理请求信息的装置的单元框图。

[0083] 请参见图7所示,本公开提供一种处理请求信息的装置,应用于客户端,包括:获取第一请求信息单元701,生成第二请求信息单元702,生成第二请求消息单元703。



- [0084] 获取第一请求信息单元701,用于客户端获取多个第一请求信息;所述第一请求信息间包括相互关联的第一请求关系;
- [0085] 生成第二请求信息单元702,用于基于第一请求数据合并规则和第一请求关系描述规则合并所述第一请求信息,生成第二请求信息;
- [0086] 生成第二请求消息单元703,用于根据所述第二请求信息生成第二请求消息,并向服务端的应用程序接口网关发送所述第二请求消息。
- [0087] 可选的,在所述生成第二请求信息单元702中,包括:
- [0088] 生成第一合并请求信息子单元,用于根据所述第一请求数据合并规则合并所述第一请求信息,生成第一合并请求信息;
- [0089] 生成第一合并请求关系信息子单元,用于根据所述第一请求关系描述规则合并所述第一请求关系,生成第一合并请求关系信息;
- [0090] 生成第二请求信息子单元,用于合并所述第一合并请求信息和所述第一合并请求关系信息,生成第二请求信息。
- [0091] 可选的,所述第一请求信息包括第一请求映射数据结构描述的信息;
- [0092] 在所述生成第一合并请求信息子单元中,包括:
- [0093] 合并第一请求映射数据结构子单元,用于根据所述第一请求数据合并规则合并所述第一请求映射数据结构,生成第一合并请求信息;所述第一合并请求信息包括多个串行排列的第一请求映射数据结构描述的信息。
- [0094] 可选的,所述第一请求关系包括第一请求关系结构体描述的信息;
- [0095] 在所述生成第一合并请求关系信息子单元中,包括:
- [0096] 合并第一请求关系结构体子单元,用于根据所述第一请求关系描述规则合并所述第一请求关系结构体,生成第一合并请求关系信息;所述第一合并请求关系信息包括多个串行排列的第一请求关系结构体描述的信息。
- [0097] 可选的,所述第二请求信息包括一个第二请求合并映射数据结构描述的信息;
- [0098] 所述第二请求合并映射数据结构包括:所述第一请求映射数据结构和其所属的第一请求映射数据结构的名称,以及所述第一请求关系结构体和其所属的第一请求关系结构体的名称。
- [0099] 本公开实施例提供了在一次信息交互过程中将发送的多个请求消息减少到一个,有效的减少了耗时,提升了用户体验;且减少了API网关负载,提升API网关吞吐量。
- [0100] 与本公开提供的第一实施例相关联的,本公开还提供了第三实施例,即另一种传送网络数据的方法的实施例。在服务端响应客户端发送的请求消息前,需要服务端的应用程序接口网关根据请求消息的特征将请求消息分送给对应的应用程序微服务进行响应处理。本公开实施例所述方法应用于服务端的应用程序接口网关。
- [0101] 下面结合图对本公开实施例进行详细说明,请参见图8所示。
- [0102] 步骤S801,接收第二请求消息。
- [0103] 第二请求消息是本公开第一实施例中客户端发送的一个合成信息的消息。
- [0104] 步骤S802,基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息,获取多个第一请求信息和多个第一请求关系。
- [0105] 第一请求数据合并规则和第一请求关系描述规则与本公开第一实施例中的第一

请求数据合并规则和第一请求关系描述规则是同样的规则,以便根据相同规则分析客户端发送的第二请求消息。

[0106] 第一请求关系是第一请求信息间相互关联的关系。例如,串联关系或依赖关系。

[0107] 请参见第一实施例,所述第一请求关系包括第一请求关系结构体描述的信息。

[0108] 第一请求关系结构体,包括:第一请求管道、第一请求信息和第一请求数据映射。

[0109] 由于第二请求消息是根据多个第一请求关系将多个第一请求信息合并在一起的,因此,当解析第二请求消息后,能够获取与本公开第一实施例相同的多个第一请求信息和多个第一请求关系。

[0110] 步骤S803,分析每个第一请求信息,获取对应的应用程序微服务。

[0111] 由于每个第一请求信息中均包含访问的应用程序微服务的信息,因此,分析每个第一请求信息,能够获取每个第一请求信息对应的应用程序微服务。例如,第一请求信息中包括第一请求方法信息和第一请求路径信息,根据请求方法信息和请求路径信息就可以获取对应的应用程序微服务。

[0112] 可选的,请参见第一实施例,所述第一请求信息包括第一请求映射数据结构描述的信息。

[0113] 第一请求映射数据结构包括第一请求名称和第一请求结构体。

[0114] 第一请求结构体包括第一请求方法和第一请求路径。

[0115] 步骤S804,依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

[0116] 所述顺序向所述应用程序微服务发送对应的第一请求信息,也就是根据第一请求关系所约定的串行关系或依赖关系顺序向所述服务端发送对应的第一请求信息。

[0117] 可选的,所述方法还包括以下步骤:

[0118] 步骤S121,依次获取响应所述第一请求信息生成的第一响应信息。

[0119] 由于第一请求信息间包括相互关联的第一请求关系,同样,所述第一响应信息间包括相互关联的第一响应关系。

[0120] 所述第一响应信息包括第一响应映射数据结构描述的信息。

[0121] 第一响应映射数据结构包括第一响应名称和第一响应结构体。

[0122] 第一响应结构体包括第一响应数据名称和第一响应数据。请参见图6所示。

[0123] 所述第一响应关系包括第一响应关系结构体描述的信息。

[0124] 第一响应关系结构体,包括:第一响应管道、第一响应信息和第一响应数据映射。

[0125] 第一响应数据映射用于描述第一响应信息之间的数据依赖关系的映射数据结构。

[0126] 第一响应数据映射,包括:第一响应映射键值和第一响应映射值。

[0127] 步骤S122,基于所述第一响应数据合并规则和所述第一响应关系描述规则合并所述第一响应信息,生成第二响应信息。

[0128] 第一响应数据合并规则用于定义一次信息交互过程中第二响应信息描述多个第一响应信息的响应数据的规则。第一响应数据合并规则和第一响应关系描述规则均是预先设计的规则。第一响应数据合并规则可以与第一请求数据合并规则相同,也可以不同。

[0129] 第一响应关系描述规则用于定义一次信息交互过程中第二响应信息描述多个第一响应关系的规则。第一响应关系描述规则可以与第一请求关系描述规则相同,也可以不

同。

[0130] 所述第二响应信息包括一个第二响应合并映射数据结构描述的信息。

[0131] 所述第二响应合并映射数据结构包括:所述第一响应映射数据结构和第一响应映射数据结构的名称,以及所述第一响应关系结构体和第一响应关系结构体的名称。

[0132] 第二响应合并映射数据结构可以与第二请求合并映射数据结构相同,也可以不同。

[0133] 步骤S123,根据所述第二响应信息生成第二响应消息,并向客户端返回所述第二响应消息。

[0134] 请参见图2所示,本公开实施例提供了在一次信息交互过程中将发送的多个请求消息减少到一个,有效的减少了耗时,提升了用户体验;且减少了API网关负载,提升API网关吞吐量。

[0135] 与本公开提供的第三实施例相对应,本公开还提供了第四实施例,即另一种传送网络数据的装置。由于第四实施例基本相似于第三实施例,所以描述得比较简单,相关的部分请参见第三实施例的对应说明即可。下述描述的装置实施例仅仅是示意性的。

[0136] 请参见图9所示,本公开提供另一种传送网络数据的装置,应用于服务端的应用程序接口网关,包括:接收第二请求消息单元901,分析第二请求消息单元902,获取应用程序微服务单元903,发送第一请求信息单元904。

[0137] 接收第二请求消息单元901,用于接收第二请求消息;

[0138] 分析第二请求消息单元902,用于基于第一请求数据合并规则和第一请求关系描述规则分析所述第二请求消息,获取多个第一请求信息和多个第一请求关系;

[0139] 获取应用程序微服务单元903,用于分析每个第一请求信息,获取对应的应用程序微服务;

[0140] 发送第一请求信息单元904,用于依据所述第一请求关系顺序向应用程序微服务发送对应的第一请求信息。

[0141] 可选的,所述装置还包括:

[0142] 获取第一响应信息单元,用于依次获取响应所述第一请求信息生成的第一响应信息;所述第一响应信息间包括相互关联的第一响应关系;

[0143] 生成第二响应信息单元,用于基于所述第一响应数据合并规则和所述第一响应关系描述规则合并所述第一响应信息,生成第二响应信息;

[0144] 发送第二响应消息单元,用于根据所述第二响应信息生成第二响应消息,并向客户端返回所述第二响应消息。

[0145] 请参见图2所示,本公开实施例提供了在一次信息交互过程中将发送的多个请求消息减少到一个,有效的减少了耗时,提升了用户体验;且减少了API网关负载,提升API网关吞吐量。

[0146] 本公开实施例提供了第五实施例,即一种电子设备,该设备用于处理请求信息的方法,所述电子设备,包括:至少一个处理器;以及,与所述至少一个处理器通信连接的存储器;其中,

[0147] 所述存储器存储有可被所述一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行如第一实施例所述处理请求信息的方法。

[0148] 本公开实施例提供了第六实施例,即一种处理请求信息的计算机存储介质,所述计算机存储介质存储有计算机可执行指令,该计算机可执行指令可执行如第一实施例中所述处理请求信息的方法。

[0149] 本公开实施例提供了第七实施例,即一种电子设备,该设备用于处理请求信息的方法,所述电子设备,包括:至少一个处理器;以及,与所述至少一个处理器通信连接的存储器;其中,

[0150] 所述存储器存储有可被所述一个处理器执行的指令,所述指令被所述至少一个处理器执行,以使所述至少一个处理器能够执行如第三实施例所述处理请求信息的方法。

[0151] 本公开实施例提供了第八实施例,即一种处理请求信息的计算机存储介质,所述计算机存储介质存储有计算机可执行指令,该计算机可执行指令可执行如第三实施例中所述处理请求信息的方法。

[0152] 下面参考图10,其示出了适于用来实现本公开第五实施例或第七实施例的电子设备的结构示意图。本公开实施例中的终端设备可以包括但不限于诸如移动电话、笔记本电脑、数字广播接收器、PDA(个人数字助理)、PAD(平板电脑)、PMP(便携式多媒体播放器)、车载终端(例如车载导航终端)等等的移动终端以及诸如数字TV、台式计算机等等的固定终端。图10示出的电子设备仅仅是一个示例,不应对本公开实施例的功能和使用范围带来任何限制。

[0153] 如图10所示,电子设备可以包括处理装置(例如中央处理器、图形处理器等)1001,其可以根据存储在只读存储器(ROM)1002中的程序或者从存储装置1008加载到随机访问存储器(RAM)1003中的程序而执行各种适当的动作和处理。在RAM 1003中,还存储有电子设备操作所需的各种程序和数据。处理装置1001、ROM 1002以及RAM 1003通过总线1004彼此相连。输入/输出(I/O)接口1005也连接至总线1004。

[0154] 通常,以下装置可以连接至I/O接口1005:包括例如触摸屏、触摸板、键盘、鼠标、摄像头、麦克风、加速度计、陀螺仪等的输入装置1006;包括例如液晶显示器(LCD)、扬声器、振动器等的输出装置1007;包括例如磁带、硬盘等的存储装置1008;以及通信装置1009。通信装置1009可以允许电子设备与其他设备进行无线或有线通信以交换数据。虽然图10示出了具有各种装置的电子设备,但是应理解的是,并不要求实施或具备所有示出的装置。可以替代地实施或具备更多或更少的装置。

[0155] 特别地,根据本公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本公开的实施例包括一种计算机程序产品,其包括承载在非暂态计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信装置1009从网络上被下载和安装,或者从存储装置1008被安装,或者从ROM 1002被安装。在该计算机程序被处理装置1001执行时,执行本公开实施例的方法中限定的上述功能。

[0156] 需要说明的是,本公开上述的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储

器 (EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器 (CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本公开中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本公开中,计算机可读信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读信号介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:电线、光缆、RF (射频) 等等,或者上述的任意合适的组合。

[0157] 在一些实施方式中,客户端、服务器可以利用诸如HTTP (HyperText Transfer Protocol,超文本传输协议) 之类的任何当前已知或未来研发的网络协议进行通信,并且可以与任意形式或介质的数字数据通信 (例如,通信网络) 互连。通信网络的示例包括局域网 (“LAN”), 广域网 (“WAN”), 网际网 (例如,互联网) 以及端对端网络 (例如,ad hoc端对端网络), 以及任何当前已知或未来研发的网络。

[0158] 上述计算机可读介质可以是上述电子设备中所包含的;也可以是单独存在,而未装配入该电子设备。

[0159] 可以以一种或多种程序设计语言或其组合来编写用于执行本公开的操作的计算机程序代码,上述程序设计语言包括但不限于面向对象的程序设计语言—诸如Java、Smalltalk、C++, 还包括常规的过程式程序设计语言—诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络—包括局域网 (LAN) 或广域网 (WAN) —连接到用户计算机,或者,可以连接到外部计算机 (例如利用因特网服务提供商来通过因特网连接)。

[0160] 附图中的流程图和框图,图示了按照本公开各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0161] 描述于本公开实施例中所涉及到的单元可以通过软件的方式实现,也可以通过硬件的方式来实现。其中,单元的名称在某种情况下并不构成对该单元本身的限定。

[0162] 本文中以上描述的功能可以至少部分地由一个或多个硬件逻辑部件来执行。例如,非限制性地,可以使用的示范类型的硬件逻辑部件包括:现场可编程门阵列 (FPGA)、专用集成电路 (ASIC)、专用标准产品 (ASSP)、片上系统 (SOC)、复杂可编程逻辑设备 (CPLD) 等

等。

[0163] 在本公开的上下文中,机器可读介质可以是有形的介质,其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备,或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器 (RAM)、只读存储器 (ROM)、可擦除可编程只读存储器 (EPROM 或快闪存储器)、光纤、便捷式紧凑盘只读存储器 (CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0164] 以上描述仅为本公开的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解,本公开中所涉及的公开范围,并不限于上述技术特征的特定组合而成的技术方案,同时也应涵盖在不脱离上述公开构思的情况下,由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本公开中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

[0165] 此外,虽然采用特定次序描绘了各操作,但是这不应理解为要求这些操作以所示出的特定次序或以顺序次序执行来执行。在一定环境下,多任务和并行处理可能是有利的。同样地,虽然在上面论述中包含了若干具体实现细节,但是这些不应被解释为对本公开的范围的限制。在单独的实施例的上下文中描述的某些特征还可以组合地实现在单个实施例中。相反地,在单个实施例的上下文中描述的各种特征也可以单独地或以任何合适的子组合的方式实现在多个实施例中。

[0166] 尽管已经采用特定于结构特征和/或方法逻辑动作的语言描述了本主题,但是应当理解所附权利要求书中所限定的主题未必局限于上面描述的特定特征或动作。相反,上面所描述的特定特征和动作仅仅是实现权利要求书的示例形式。

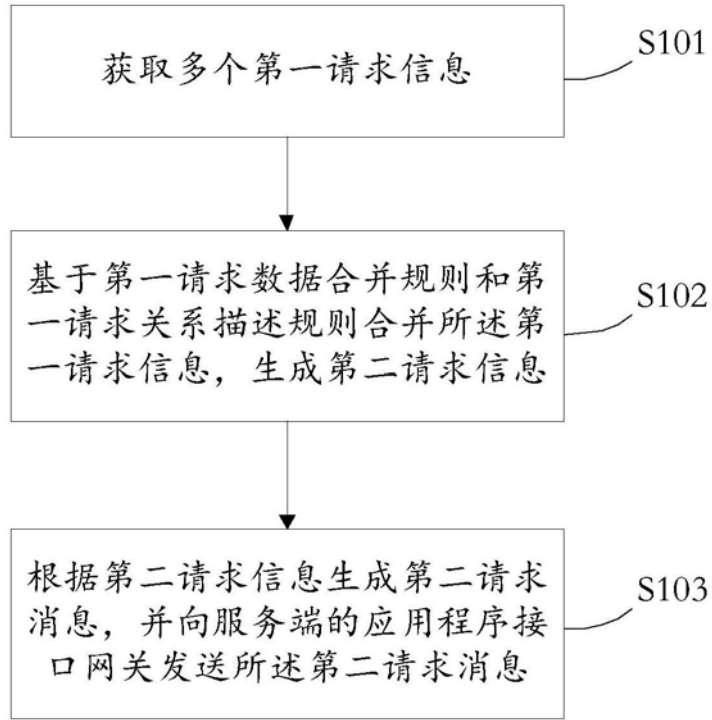


图1

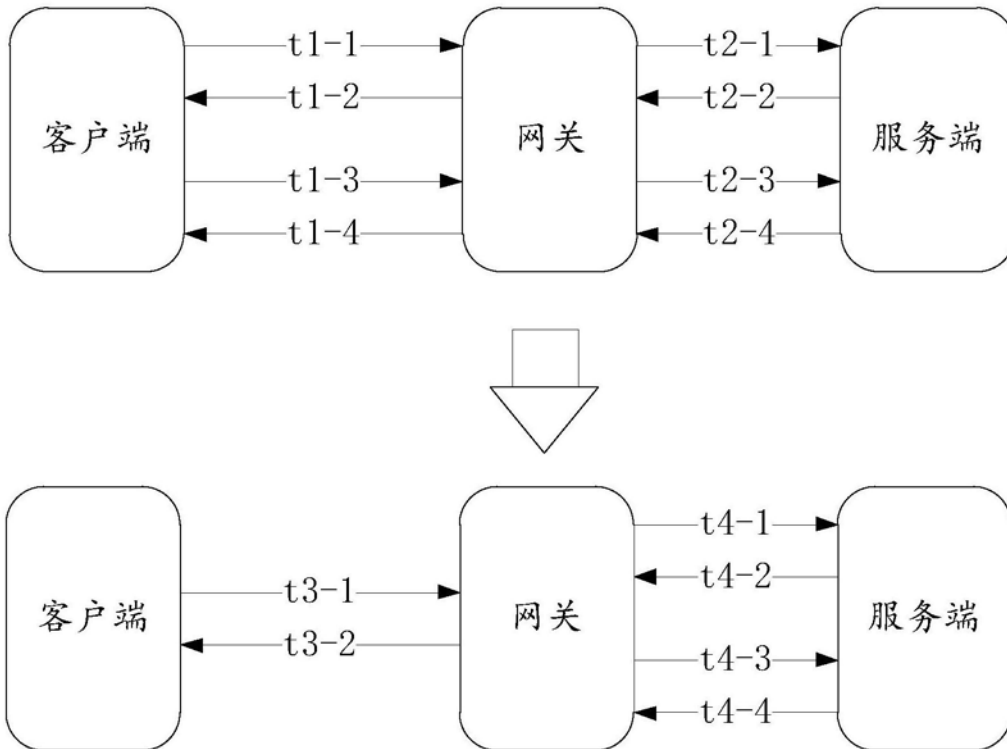


图2

```
apiInfos: {  
  api1: { method: 'POST', path: '/api/api1' },  
  api2: { method: 'POST', path: '/api/api2' },  
},
```

图3

```
{  
  pipeline: '|api1|api2|',  
  apiInfos: {  
    api1: { method: 'POST', path: '/api/api1' },  
    api2: { method: 'POST', path: '/api/api2' },  
  },  
  dataMaps: {  
    'api2.cname': 'api1.name'  
  },  
}
```

图4



```
{
  reqDataMap: {
    api1: {
      id: '1234'
    },
    api2: {
      cname: 'lance'
    }
  },
  apiRelation: {
    pipeline: '|api1|api2|',
    apiInfos: {
      api1: { method: 'POST', path: '/api/api1' },
      api2: { method: 'POST', path: '/api/api2' },
    },
    dataMaps: {
      'api2.name': 'api1.cname'
    },
  }
}
```

图5

```
{  
  apil: {  
    name: 'lance'  
  },  
  api2: {  
    age: 18  
  }  
}
```

图6

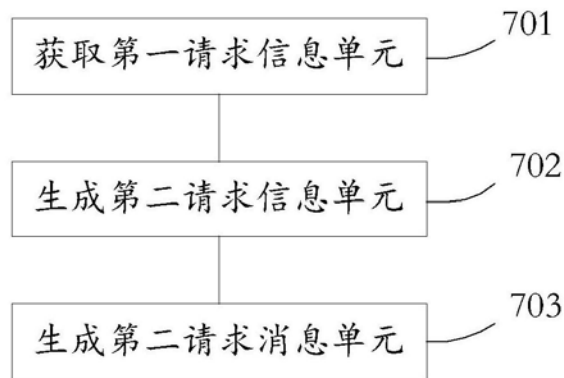


图7

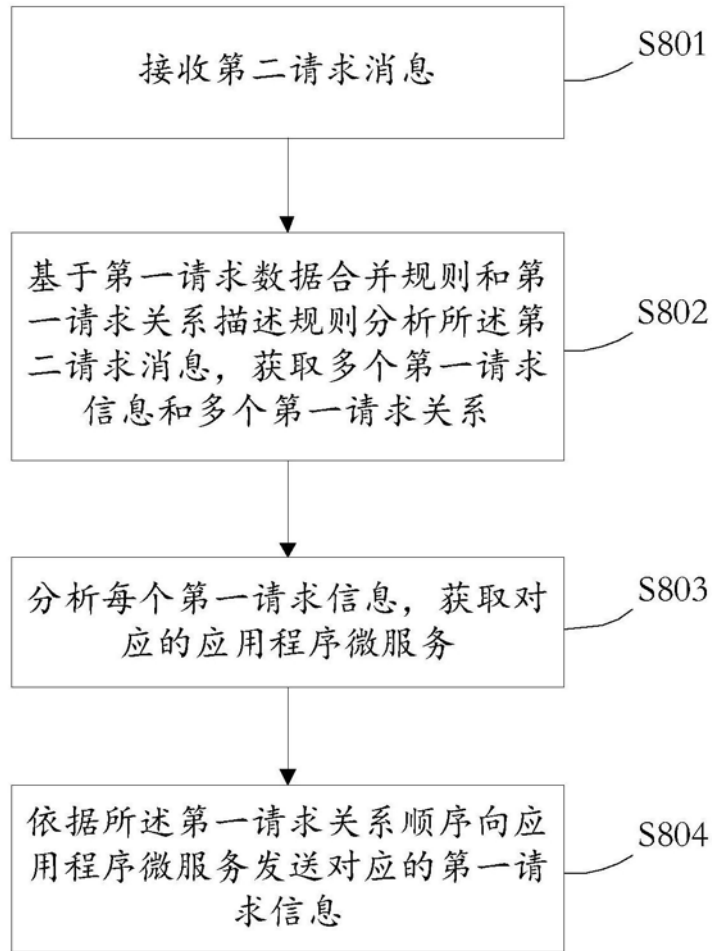


图8

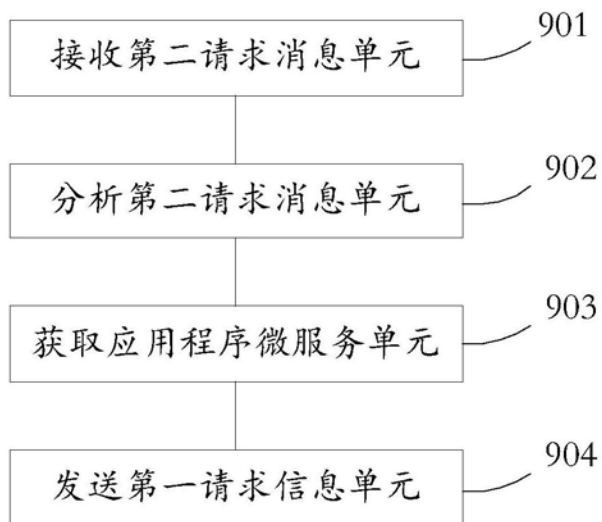


图9

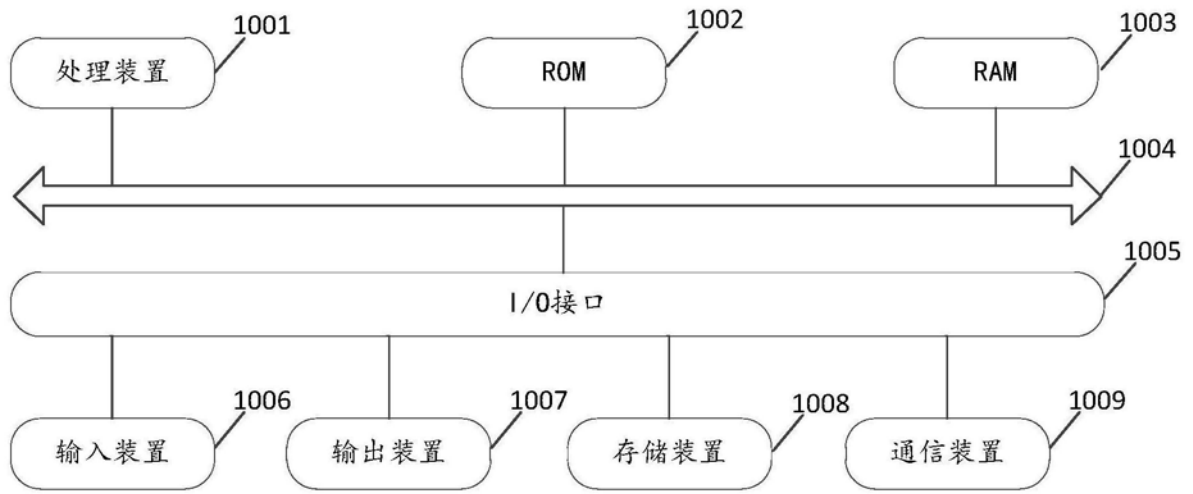


图10