(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 111427766 A
(43)申请公布日 2020.07.17

(21)申请号 202010104636.6

(22)申请日 2020.02.20

(71)申请人 北京齐尔布莱特科技有限公司
地址 100080 北京市海淀区丹棱街3号B座10层1010室

(72)发明人 郝丹 任晓伟

(74)专利代理机构 北京思睿峰知识产权代理有限公司 11396
代理人 史小娟 赵爱军

(51)Int.Cl.
G06F 11/36(2006.01)
G06F 8/35(2018.01)
H04L 29/08(2006.01)

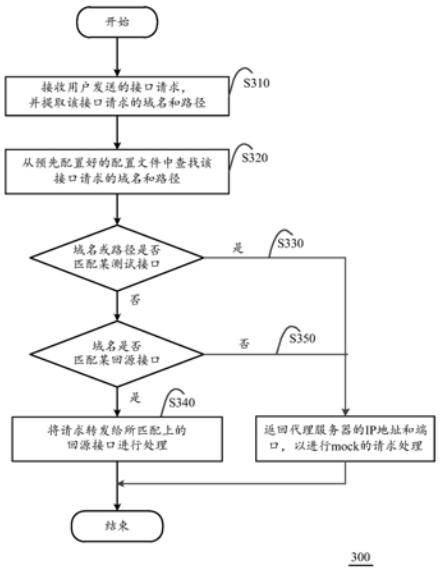
权利要求书2页 说明书12页 附图5页

(54)发明名称

一种请求处理方法、装置和代理服务器

(57)摘要

本发明公开了一种请求处理方法，在代理服务器中执行，包括：接收用户发送的接口请求，并提取该接口请求的域名和路径；从预先配置好的配置文件中查找接口请求的域名和路径，该配置文件存储有多个回源接口的域名，以及接口管理平台中多个测试接口的域名、路径和对应的响应内容；若接口请求的域名或路径匹配某测试接口，或者，接口请求的域名和路径均不匹配任何测试接口、且域名也不匹配任何回源接口，则返回代理服务器的IP地址和端口，以进行mock的请求处理；若接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口，则将该请求转发给所匹配上的回源接口进行请求处理。本发明还一并公开了对应的请求处理装置和系统。



CN 111427766 A

1.一种请求处理方法,适于在代理服务器中执行,所述代理服务器为接口管理平台的上游服务器,所述方法包括步骤:

接收用户发送的接口请求,并提取该接口请求的请求参数,所述请求参数包括域名和路径;

从预先配置好的配置文件中查找所述接口请求的域名和路径,所述配置文件存储有多个回源接口的域名,以及接口管理平台中多个测试接口的域名、路径和对应的响应内容;

若所述接口请求的域名或路径匹配某测试接口,则返回所述代理服务器的IP地址和端口,以进行mock的请求处理;

若所述接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口,则将该请求转发给所匹配上的回源接口进行请求处理;

若所述接口请求的域名和路径均不匹配任何测试接口、且该域名也不匹配任何回源接口,则返回所述代理服务器的IP地址和端口,以进行mock的请求处理。

2.如权利要求1所述的方法,其中,所述进行mock的请求处理的步骤包括:

判断所述接口请求的域名和路径是否与某测试接口匹配;

若是,则向用户返回该测试接口的响应内容;

若否,则判断所述接口请求的路径是否有匹配的测试接口,若有,则向用户返回该测试接口的响应内容,若没有,则返回结果为空。

3.如权利要求1或2所述的方法,其中,在所述提取该接口请求的域名和路径之后,还包括步骤:

将所述接口请求的域名对应的IP地址设置为所述代理服务器的IP地址。

4.如权利要求1-3中任一项所述的方法,其中,

所述配置文件包括请求参数、请求匹配规则、回源域名配置和响应配置中的至少一种;

所述请求参数包括域名、路径和参数中的至少一种;

所述请求匹配规则是为动态请求参数而配置的规则机;

所述响应配置包括固定结果响应、动态请求匹配规则的响应和基于动态编码的响应中的至少一种。

5.如权利要求4所述的方法,其中,

所述固定结果响应为固定的字符串,所述动态编码的响应通过python代码方式配置;

所述动态请求匹配规则的响应通过对接口请求中的请求参数进行正则匹配和占位符替换来配置。

6.如权利要求1-5中任一项所述的方法,其中,

所述基于动态编码的响应包括广告请求的响应,所述广告请求的请求参数中只有广告位标识可变,多个广告位标识可共用同一个广告模板,所述配置文件中存储有每个广告模板的响应数据。

7.如权利要求6所述的方法,其中,所述广告模板的响应数据通过以下过程配置:

从线上抓取该广告模板的广告数据,并将该广告数据作为响应内容,以代码方式存入该广告模板的各广告位的响应配置中。

8.一种请求处理装置,适于驻留在代理服务器中,所述代理服务器为接口管理平台的上游服务器,所述装置包括:

请求接收模块，适于接收用户发送的接口请求，并提取该接口请求的请求参数，所述请求参数包括域名和路径；

数据查找模块，适于从预先配置好的配置文件中查找所述接口请求的域名和路径，所述配置文件存储有多个回源接口的域名，以及接口管理平台中多个测试接口的域名、路径和对应的响应内容；

第一处理模块，适于当所述接口请求的域名或路径匹配某测试接口时，返回所述代理服务器的IP地址和端口，以进行mock的请求处理；

第二处理模块，适于当所述接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口时，将该请求转发给所匹配上的回源接口进行请求处理；

第三处理模块，适于当所述接口请求的域名和路径均不匹配任何测试接口的配置信息、且该域名也不匹配任何回源接口的配置信息时，返回所述代理服务器的IP地址和端口，以进行mock的请求处理。

9.一种代理服务器，包括：

一个或多个处理器；

存储器；以及

一个或多个程序，其中所述一个或多个程序存储在所述存储器中并被配置为由所述一个或多个处理器执行，所述一个或多个程序被处理器执行时实现权利要求1-7任一项所述的方法的步骤。

10.一种存储一个或多个程序的可读存储介质，所述一个或多个程序包括指令，所述指令当由代理服务器执行时实现权利要求1-7任一项所述的方法的步骤。

一种请求处理方法、装置和代理服务器

技术领域

[0001] 本发明涉及自动化测试领域，尤其涉及一种请求处理方法、装置和代理服务器。

背景技术

[0002] mock测试就是在测试过程中，对于某些不容易构造或者不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法。mock技术目前有大量的应用场景，且在不同的领域有不同的技术出现，比如前端开发用到的mock.js技术，以及后端用到的postman mock等。

[0003] mock.js的mock方式是通过在前端代码里面进行js的代码编写进行mock，但其使用场景局限，例如在黑盒测试和联调的场景中不可使用。而且，mock的数据通过开发来维护，对于表格等复杂的数据进行mock时，需要对复杂的规则进行理解并适配才可以解析。mockjs依赖js代码，支持度不高。

[0004] 基于postman进行mock的技术，可以在postman上面创建mock，生成一个新的url，然后使用mock服务器返回mock数据。但在联调的过程中，需要修改配置才可以使用。对于广告模版的mock测试，需要频繁的构造重复的数据，效率低且容易数据冲突。

[0005] 因此，需要提供一种更便捷灵活的mock测试方法。

发明内容

[0006] 鉴于上述问题，本发明提出了一种请求处理方法、装置和代理服务器，以力图解决或者至少解决上面存在的问题。

[0007] 根据本发明的一个方面，提供了一种请求处理方法，适于在代理服务器中执行，代理服务器为接口管理平台的上游服务器，该方法包括步骤：接收用户发送的接口请求，并提取该接口请求的请求参数，请求参数包括域名和路径；从预先配置好的配置文件中查找接口请求的域名和路径，配置文件存储有多个回源接口的域名，以及接口管理平台中多个测试接口的域名、路径和对应的响应内容；若接口请求的域名或路径匹配某测试接口，则返回代理服务器的IP地址和端口，以进行mock的请求处理；若接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口，则将该请求转发给所匹配上的回源接口进行请求处理；若接口请求的域名和路径均不匹配任何测试接口、且该域名也不匹配任何回源接口，则返回代理服务器的IP地址和端口，以进行mock的请求处理。

[0008] 可选地，在根据本发明的请求处理方法中，进行mock的请求处理的步骤包括：判断接口请求的域名和路径是否与某测试接口匹配；若是，则向用户返回该测试接口的响应内容；若否，则判断所述接口请求的路径是否有匹配的测试接口，若有，则向用户返回该测试接口的响应内容，若没有，则返回结果为空。

[0009] 可选地，在根据本发明的请求处理方法中，在提取该接口请求的域名和路径之后，还包括步骤：将接口请求的域名对应的IP地址设置为所述代理服务器的IP地址。

[0010] 可选地，在根据本发明的请求处理方法中，配置文件包括请求参数、请求匹配规

则、回源域名配置和响应配置中的至少一种;请求参数包括域名、路径和参数中的至少一种;请求匹配规则是为动态请求参数而配置的规则机;响应配置包括固定结果响应、动态请求匹配规则的响应和基于动态编码的响应中的至少一种。

[0011] 可选地,在根据本发明的请求处理方法中,固定结果响应为固定的字符串,动态编码的响应通过python代码方式配置;动态请求匹配规则的响应通过对接口请求中的请求参数进行正则匹配和占位符替换来配置。

[0012] 可选地,在根据本发明的请求处理方法中,基于动态编码的响应包括广告请求的响应,广告请求的请求参数中只有广告位标识可变,多个广告位标识可共用同一个广告模板,配置文件中存储有每个广告模板的响应数据。

[0013] 可选地,在根据本发明的请求处理方法中,广告模板的响应数据通过以下过程配置:从线上抓取该广告模板的广告数据,并将该广告数据作为响应内容,以代码方式存入该广告模板的各广告位的响应配置中。

[0014] 可选地,在根据本发明的请求处理方法中,接口请求为广告接口请求,所提取的请求参数还包括广告位标识,该广告接口请求所对应的mock响应数据为该广告位标识所对应的广告数据。

[0015] 可选地,在根据本发明的请求处理方法中,配置文件存储在redis数据库中,代理服务器通过访问redis数据库来从预先配置好的配置文件中查找接口请求的域名和路径。

[0016] 根据本发明的另一个方面,提供了一种请求处理装置,适于驻留在代理服务器中,代理服务器为接口管理平台的上游服务器,该装置包括:请求接收模块,适于接收用户发送的接口请求,并提取该接口请求的请求参数,请求参数包括域名和路径;数据查找模块,适于从预先配置好的配置文件中查找所述接口请求的域名和路径,配置文件存储有多个回源接口的域名,以及接口管理平台中多个测试接口的域名、路径和对应的响应内容;第一处理模块,适于当所述接口请求的域名或路径匹配某测试接口时,返回代理服务器的IP地址和端口,以进行mock的请求处理;第二处理模块,适于当所述接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口时,将该请求转发给所匹配上的回源接口进行请求处理;第三处理模块,适于当所述接口请求的域名和路径均不匹配任何测试接口、且该域名也不匹配任何回源接口时,返回所述代理服务器的IP地址和端口,以进行mock的请求处理。

[0017] 根据本发明的又一方面,提供一种代理服务器,包括:一个或多个处理器;存储器;以及一个或多个程序,其中一个或多个程序存储在存储器中并被配置为由一个或多个处理器执行,该一个或多个程序被处理器执行时实现如上所述的请求处理方法的步骤。

[0018] 根据本发明的又一方面,提供一种存储一个或多个程序的可读存储介质,该一个或多个程序包括指令,所述指令当由代理服务器执行时实现如上所述的请求处理方法的步骤。

[0019] 根据本发明的技术方案,提供了一种通过host配置进行mock,用代理服务器进行动态路由转发的方法,实现页面即时配置即时生效的实时效果。本发明支持用python动态mock编程,实现页面进行python编程,动态化实现用户的灵活的mock结果生成需求。而且,本发明还可以将广告线上数据倒入mock库进行一键mock,同时将其作为各业务的线下应用,能够以较低的开发和维护成本应用在大多数mock使用的场景。

[0020] 上述说明仅是本发明技术方案的概述，为了能够更清楚了解本发明的技术手段，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

[0021] 为了实现上述以及相关目的，本文结合下面的描述和附图来描述某些说明性方面，这些方面指示了可以实践本文所公开的原理的各种方式，并且所有方面及其等效方面旨在落入所要求保护的主题的范围内。通过结合附图阅读下面的详细描述，本公开的上述以及其它目的、特征和优势将变得更加明显。遍及本公开，相同的附图标记通常指代相同的部件或元素。

[0022] 图1示出了根据本发明一个实施例的请求处理系统100的结构框图；

[0023] 图2示出了根据本发明一个实施例的计算设备200的结构图；

[0024] 图3示出了根据本发明一个实施例的请求处理方法300的流程图；

[0025] 图4示出了根据本发明一个实施例的接口管理平台的配置示意图；

[0026] 图5示出了根据本发明一个实施例的动态编码响应信息返回的示意图；

[0027] 图6示出了根据本发明一个实施例的广告位mock数据的配置示意图；以及

[0028] 图7示出了根据本发明一个实施例的请求处理装置700的结构图。

具体实施方式

[0029] 下面将参照附图更详细地描述本公开的示例性实施例。虽然附图中显示了本公开的示例性实施例，然而应当理解，可以以各种形式实现本公开而不应被这里阐述的实施例所限制。相反，提供这些实施例是为了能够更透彻地理解本公开，并且能够将本公开的范围完整的传达给本领域的技术人员。

[0030] 图1示出了根据本发明一个实施例的请求处理系统100的示意图。如图1所示，请求处理系统100中包括一个或多个客户端110、代理服务器120、接口管理平台130和处理设备140。代理服务器120分别与客户端110、接口管理平台130和处理设备140通信连接，用于接收并转发客户端110的接口请求。应当指出，图1中的请求处理系统100仅是示例性的，在具体的实践情况中，系统100中可以有不同数量的代理服务器120、接口管理平台130和处理设备140，本发明对系统100中所包括的各设备数量不做限制。

[0031] 客户端110包括一个或多个服务的请求接口，处理设备140包括一个或多个服务的处理接口，可称这些处理接口为回源接口，也就是回到源服务进行处理的接口。例如，测试的接口A同时请求B服务的接口1和接口2，其中接口A为请求接口，接口1和2为处理接口。

[0032] 代理服务器120可以是一台单独的服务器，也可以是由若干台服务器组成的服务器集群，或者是一个云计算服务中心，用于组成服务器集群或云计算服务中心的多个服务器可以驻留在多个地理位置中，本发明对各服务器的部署方式不做限制。代理服务器120可以是目前常用的代理服务器，如可以为Nginx代理服务器，本发明对此不作限制。

[0033] 接口管理平台130也可称为接口管理系统，其为mock平台，在该平台可以配置并记录mock信息，如可以配置多个测试接口的域名、路径、请求方式、响应内容等，生成对应的配置文件。代理服务器120作为接口管理平台130的上游服务器，可以从接口管理平台130获取

对应的配置文件。

[0034] 根据一个实施例，系统100还可以包括数据库150，其与代理服务器120和接口管理平台130均通信连接，用于存储在接口管理平台130上生成的配置数据。这样，代理服务器120通过访问数据库150，即可得到该配置文件。或者，代理服务器120可以从该数据库150中缓存该配置文件。根据一个实施例，数据库150可以是redis数据库。

[0035] 客户端110中一般安装有目标应用，且被导入受信任的https证书以实现mock接口功能，在客户端110中还设置有代理服务器机制。这样客户端110发送的所有请求和响应就都会通过代理服务器120转发出去。通常，目标应用包括多个应用页面，一个页面可能同时会依赖多个接口，因此客户端110发送的测试请求中通常会包括一个或多个接口请求。这些接口中可能包括业务类接口和非业务接口。业务类接口即接口返回的字段和应用界面逻辑相关，非业务接口如埋点请求接口、广告曝光请求接口等，主要是为了运营统计和核销广告使用的，与应用正常界面显示没有直接关系。

[0036] 根据本发明的一个实施例，上述请求处理系统100中的客户端110、代理服务器120、接口管理平台130和处理设备140均可以通过如下所述的计算设备200来实现。图2示出了根据本发明一个实施例的计算设备200的结构框图。

[0037] 在基本的配置202中，计算设备200典型地包括系统存储器206和一个或者多个处理器204。存储器总线208可以用于在处理器204和系统存储器206之间的通信。

[0038] 取决于期望的配置，处理器204可以是任何类型的处理，包括但不限于：微处理器(μP)、微控制器(μC)、数字信息处理器(DSP)或者它们的任何组合。处理器204可以包括诸如一级高速缓存210和二级高速缓存212之类的一个或者多个级别的高速缓存、处理器核心214和寄存器216。示例的处理器核心214可以包括运算逻辑单元(ALU)、浮点数单元(FPU)、数字信号处理核心(DSP核心)或者它们的任何组合。示例的存储器控制器218可以与处理器204一起使用，或者在一些实现中，存储器控制器218可以是处理器204的一个内部部分。

[0039] 取决于期望的配置，系统存储器206可以是任意类型的存储器，包括但不限于：易失性存储器(诸如RAM)、非易失性存储器(诸如ROM、闪存等)或者它们的任何组合。系统存储器206可以包括操作系统220、一个或者多个应用222以及程序数据224。在一些实施方式中，应用222可以布置为在操作系统上利用程序数据224进行操作。程序数据224包括指令，在根据本发明的计算设备200中，程序数据224包含用于执行请求处理方法300的指令。

[0040] 计算设备200还可以包括有助于从各种接口设备(例如，输出设备242、外设接口244和通信设备246)到基本配置202经由总线/接口控制器230的通信的接口总线240。示例的输出设备242包括图形处理单元248和音频处理单元250。它们可以被配置为有助于经由一个或者多个A/V端口252与诸如显示器或者扬声器之类的各种外部设备进行通信。示例外设接口244可以包括串行接口控制器254和并行接口控制器256，它们可以被配置为有助于经由一个或者多个I/O端口258和诸如输入设备(例如，键盘、鼠标、笔、语音输入设备、触摸输入设备)或者其他外设(例如打印机、扫描仪等)之类的外部设备进行通信。示例的通信设备246可以包括网络控制器260，其可以被布置为便于经由一个或者多个通信端口264与一个或者多个其他计算设备262通过网络通信链路的通信。

[0041] 网络通信链路可以是通信介质的一个示例。通信介质通常可以体现为在诸如载波或者其他传输机制之类的调制数据信号中的计算机可读指令、数据结构、程序模块，并且可

以包括任何信息递送介质。“调制数据信号”可以这样的信号，它的数据集中的一个或者多个或者它的改变可以在信号中编码信息的方式进行。作为非限制性的示例，通信介质可以包括诸如有线网络或者专线网络之类的有线介质，以及诸如声音、射频(RF)、微波、红外(IR)或者其它无线介质在内的各种无线介质。这里使用的术语计算机可读介质可以包括存储介质和通信介质二者。

[0042] 计算设备200可以实现为服务器，例如文件服务器、数据库服务器、应用程序服务器和WEB服务器等，也可以实现为小尺寸便携(或者移动)电子设备的一部分，这些电子设备可以是诸如蜂窝电话、个人数字助理(PDA)、无线网络浏览设备、应用专用设备、或者可以包括上面任何功能的混合设备。计算设备200还可以实现为包括桌面计算机和笔记本计算机配置的个人计算机。在一些实施例中，计算设备200被配置为执行请求处理方法300。

[0043] 图3示出了根据本发明一个实施例的请求处理方法300的流程示意图。方法300在代理服务器中执行，如在代理服务器120中执行，以便客户端发起的请求进行处理。

[0044] 如图3所示，该方法始于步骤S310。在步骤S310中，接收用户发送的接口请求，并提取该接口请求的请求参数，该请求参数包括域名和路径。当该接口请求为广告接口请求时，所提取的请求参数还包括广告位标识。

[0045] 随后，在步骤S320中，从预先配置好的配置文件中查找接口请求的域名和路径，该配置文件存储有多个回源接口的域名，以及接口管理平台中多个测试接口的域名、路径和对应的响应内容。

[0046] 其中，代理服务器可以通过访问数据库中的配置文件来查询该接口请求的域名和路径。如图4所示，用户在接口管理平台(mock平台)配置mock数据，生成的配置文件主要包括请求参数、请求匹配规则、回源域名(host)配置，响应(response)配置。其中，请求参数包括域名、路径和参数中的至少一种。请求匹配规则是为动态请求参数而配置的规则机。回源域名配置是为了解决用户部分请求回源的需求而进行的配置。响应配置包括固定结果响应、动态请求匹配规则的响应和基于动态编码的响应中的至少一种。

[0047] 其中，固定结果响应的返回类型为固定的字符串，比如{"result":"hello mock"}，直接固定返回该配置的字符串，响应内容不会随着输入参数变化而变化。动态请求匹配规则的响应通过对接口请求中的请求参数进行正则匹配和占位符替换来配置。这种方式返回的信息随着输入的参数变化而变化，并且有一定的规律，对应的响应内容可设置为：{"result":"hello mock param:${param},a:${a}"}，其中param和a为B服务的测试接口的输入参数，随着输入参数的不同，根据正则匹配和占位符返回不同的信息。动态编码的响应通过python代码方式配置。当需要模拟B服务的测试接口返回的信息超出了第二种能支持的范围时，如模拟日期、天气、温度等数据时，需要有复杂的计算逻辑。此时可以采用编程方式来解决该问题，以一段python代码的方式动态生成响应结果：

```
res = {"data":[],"status":0,"statusInfo": {"global": "OK"}}
id_arr = ids.split(',')
if not id_arr or len(id_arr) <= 0:
    res = ''
else:
    for id in id_arr:
        arr = {}
        arr['id'] = id
        arr['exposeNum'] = 1000
        arr['clickNum'] = 900
        res['data'].append(arr)
print json.dumps(res)
```

[0048]

[0049]

[0050] 应当理解的是，图4中的测试接口的域名路径配置和回源接口的域名配置可以作为两个独立的个体，分别在不同页面或不同平台配置，最后配置的信息会存入redis数据库中。用户请求通过配置本地域名进行使用，也就是，在获取到该接口请求的域名之后，将接口请求的域名对应的IP地址设置为代理服务器的IP地址。把请求域名对应的IP地址切换为mock系统的Nginx的固定IP地址之后，请求就可以通过mock平台进行信息的转发处理。

[0051] 以具体场景为例，若A服务通过域名访问B服务的一个测试接口，该测试接口是http://testmock.com/test?param＝123&a＝111。B服务尚未完成开发，为了联调时进行A的测试，在mock平台配置B服务的测试接口如图4所示，host为testmock.com，请求路径为/test，参数的key值固定为param和a，value值不固定。在进行联调时，获取接口管理平台的代理服务器的唯一固定IP地址，假设为127.0.0.1，则可根据之前设置的接口信息来更新域名为127.0.0.1testmock.com，即可进行请求的处理和转发。

[0052] 随后，在步骤S330中，若接口请求的域名或路径匹配某测试接口，则返回代理服务器的IP地址和端口，以进行mock的请求处理。

[0053] 或者，在步骤S340中，若接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口，则将该请求转发给所匹配上的回源接口进行请求处理。

[0054] 或者，在步骤S350中，若接口请求的域名和路径均不匹配任何测试接口、且该域名也不匹配任何回源接口，则返回代理服务器的IP地址和端口，以进行mock的请求处理。

[0055] 举例而言，若测试的接口A请求B服务的两个接口，其中请求的接口1通过mock服务处理返回，接口2则不希望通过mock服务返回，而是直接请求B服务。此时就需要路由转发进行控制，代理服务器收到的请求首先进行Lua转发。Lua由标准C编写而成，代码简洁优美，几乎在所有操作系统和平台上都可以编译运行。在转发时，首先判定请求的域名或路径在

redis内是否匹配，如果匹配的话，Lua返回mock平台的Nginx的IP地址和端口，以进行mock的请求处理。此时也就是将该请求转发给mock平台进行处理。如果域名和路径均不匹配，并且判断存在回源域名配置，则转发到回源接口进行处理。如果域名和路径均不匹配，且不存在回源域名配置，则同样返回mock平台的Nginx的IP地址和端口，以进行mock的请求处理。

[0056] 请求mock对应的IP地址和端口后，Apache转发给用Django框架实现的控制器进行mock的核心逻辑处理。该逻辑按照两层优先级进行匹配，其中第一优先级是通过对比请求和存储的配置文件判断请求的域名和路径是否同时匹配。第一优先级匹配不上的话，按照第二优先级进行匹配，第二优先级直接按照路径进行匹配，例如匹配/test，如果存储数据同时存在多个路径/test，按照时间的倒序自动匹配修改时间最晚的路径。如果两种优先级都匹配不到，则返回mock结果为空。

[0057] 具体而言，在进行mock的请求处理时，判断接口请求的域名和路径是否与某测试接口匹配，也就是两者同时匹配。若是，则向用户返回该测试接口的响应内容，也就是在配置文件中配置的该测试接口的响应内容。若否，则判断接口请求的路径是否有匹配的测试接口，若有，则向用户返回该测试接口的响应内容，若没有，则返回结果为空。

[0058] 如前文所述，mock的结果数据可随着输入参数进行动态变化，也可按照用户使用习惯灵活动态变化。前一种可以按照规则解析和占位符进行匹配。后一种可选用python编码的方式来动态生成响应结果，其信息录入如图4所示，对应的编码解释和信息返回如图5所示。而对广告模版的mock就采用后一种动态编程的方式进行操作。

[0059] 一般地，基于动态编码的响应包括广告请求的响应，广告请求的请求参数中域名和路径是不变的信息，唯一变化的是请求的广告位标识这个参数psids，请求返回的结果中也只有广告位标识这个参数发生变化。多个广告位标识可共用同一个广告模板，例如广告位5471对应的模版是PC首页自动下推通栏模板，5472-5479对应的模版也是该模版。

[0060] 基于此，用户可通过以下方式配置广告模板的响应数据：从线上抓取该广告模板的广告数据，并将该广告数据作为响应内容，以代码方式存入该广告模板的各广告位的响应配置中。模版的数据从线上请求抓取之后存储到mock的存储系统中，所生成的每个广告模板的响应数据就可一并存入配置文件中，方便后续用户再次请求该广告模板数据时直接返回结果。

[0061]
```
{
    "meta": {
        "engineInfo": {
            "creativeId": "1512175",
            "engine": "engine",
            "endTime": "2019-03-19",
            "pubTime": "2019-03-12",
```

[0062] 对于一个广告接口请求，当该请求通过mock平台进行处理时，所对应的mock响应数据及为该广告位标识所对应的线上广告数据，也就是该广告位标识所属的广告模板的线上广告数据。当需要做广告测试时，如图6所示，只需在接口管理平台进行配置，然后配置域

名即可:127.0.0.1adproxy.autohome.com.cn。

[0063] 另外一种测试场景是在对页面进行边界测试时,客户端通过代理服务器向一个或多个回源接口发送对目标应用中某页面的页面启动请求,该页面启动请求包括本次启动的唯一任务标识和一个或多个接口请求。如前文所述,用户的接口请求包括业务类接口请求和非业务类接口。代理服务器在接收到这些接口请求时,会识别其中的业务类接口请求,并对业务类接口请求的响应进行mock修改后再返回给客户端进行运行。而对非业务类接口的响应内容,代理服务器会直接放行。

[0064] 通常,代理服务器120中可以存储一个业务类接口的域名白名单,当代理服务器接收到接口请求时,可获取接口请求的请求头中的域名值,并根据该域名白名单识别其中的业务类接口请求,并对业务类接口请求的响应内容进行修改。如果HOST字段是非业务接口域名,如埋点接口域名或广告接口域名等,则不执行mock逻辑,将从接口管理平台获取到的埋点接口和广告接口的原始响应返回给客户端。这样排除掉非业务类接口请求的响应后,大大减少了需要做mock规则处理的请求数量,提高边界测试的效率。

[0065] 一般地,业务类接口请求的响应内容可采用JSON格式表示,其有多层级的数据接口,层级2嵌套在层级1中,层级3嵌套在层级2中,依此类推。每层级之间一般会用不同的空格字符表示,有相同空格字符的一般是同一层级。这样,代理服务器接收到业务类接口的请求响应时,可按照预定规则对每个响应内容中的同一层字段统一修改为某一类型的边界值,并将修改后的响应值返回给客户端进行运行。其中,边界值可以包括以下至少一种类型:随机字符串、超长字符串、特殊字符串、整型、单精度浮点型、双精度浮点型、时间、数组、布尔型、url、邮箱、IP、图片、表情、空数组、空对象、null、undefined。预定规则包括:逐层修改每个响应内容的字段,每次修改所有响应内容的同一层字段,且每次将该层字段修改为同一种类型的边界值;以及当将一层字段的所有类型的边界值遍历完毕后,再修改所有响应内容的下一层字段。

[0066] 而且,代理服务器将修改后的响应值返回给客户端后,还会将本次页面启动或本次修改的唯一任务标识、页面标识、所修改的层级和边界值类型、设备标识和运行开始时间发送到消息队列中,此时可认为完成了一次页面测试任务。每完成一次页面测试任务后,就会清理目标应用的测试环境,并进行其他页面标识或其他层级或其他边界值类型的页面测试任务,直至完成了所有页面标识下所有层级的所有边界值类型的页面测试任务。

[0067] 也就是,本发明逐页的进行应用边界测试,即让应用中指定页面在预定的各种边界值条件下运行,并在运行过程中自动捕捉应用端的接口日志和故障信息。对于某一目标页面,其通过schema方式进行第一次页面启动,之后代理服务器会拦截该页面启动请求中的业务类接口的响应内容进行mock修改。在修改时,先将所有业务类接口的层级1的字段统一修改为一个类型的边界值后返回给客户端运行,如统一修改为随机字符串类型的边界值。

[0068] 之后,清理测试环境后,对该目标页面进行第二次页面启动,此时代理服务器会再次拦截该第二次页面启动请求中的业务类接口的响应进行修改。在这第二次修改时,就可以将层级1的字段修改为第二种类型的边界值。依此类推,直到层级1的所有边界值类型都已遍历完毕。对于一个层级,如果有18种类型的边界值,则会进行18次页面启动和18次响应内容拦截和修改。之后,再开始进行层级2的边界值测试,当层级2的所有类型边界值都遍历

完成后，再进行层级3的边界值测试，依此类推。

[0069] 当一个页面的所有层级的所有边界值类型都mock完毕后，再对下一个目标页面进行边界测试，同理也是逐层级的测试。这种逐层逐类型的mcok方式可以充分挖掘应用端潜在的故障信息，保证应用端对接口每层处理逻辑中潜在的问题暴露出来。单层级和单类型的修改也方便开发人员快速实现故障定位，提高测试效率。

[0070] 对于边界测试，当应用约定的字段类型为整型时，整形数值范围是-65536～65536，那么在mock该字段类型时会分别赋值65537、null、5.5等异常值，然后观察应用端是否会发生故障。而当应用约定的字段类型是字符串型时，则在mock时会将字符串赋值为特殊字符、超长文本、表情符、null、空字符等值，然后观察应用端是否会发生故障。

[0071] 考虑到需要对多个页面和多个类型的边界值进行测试，因此在一种实现方式中，可以将多个页面标识分配给多个客户端，并存储客户端的设备标识与所分配的页面标识的关联关系表，以便各客户端执行各页面标识所对应的页面测试任务。这样，每个客户端分别负责对几个页面的边界测试，其在完成对一个页面的测试后，会自动启动对下一个页面标识的测试。这里在进行分配时，可以按照设备标识数目和页面标识数目进行平均分配，也可以根据各客户端的软硬件进行均衡分配。

[0072] 在另一种实现方式中，可以将边界值的多个类型分配给多个客户端，并存储各客户端的设备标识与所分配的边界值类型的关联关系表，以便各客户端对各页面执行对应类型的边界值测试。本发明实现了对单一设备的接口mock，代理服务器根据客户端的到不同设备标识来对接口请求响应做不同的修改。例如，A设备可以mock整型字段，B设备可以mock字符串型字段，两台设备可以并行测试而不相互影响。这里，一个设备可以执行一个或多个边界值类型，如有18个客户端，则每个客户端可只执行一种边界值类型，代理服务器在进行响应mock修改时只会把响应内容修改为对应类型的边界值。

[0073] 如有9个客户端，则每个客户端可分配两种类型边界值，在进行测试时只用完成该两种类型的边界值测试即可。当然，还可以根据客户端的内存、网速等硬件和软件差异进行均衡分配，以从整体上实现测试效率最大化。

[0074] 这样，就可从客户端上报的故障信息中判断该唯一任务标识是否有对应的故障信息，若有，则将该故障信息和对应的响应值进行自动关联。客户端在运行目标应用时，会将运行过程中的接口请求和响应上报到业务服务器，并附带设备标识、应用版本信息、软件包名称、运行时间和唯一任务标识。一旦运行过程中发生故障，就会自动捕捉故障的异常堆栈信息一并上传给业务服务器。

[0075] 图7示出了根据本发明一个实施例的请求处理装置700的结构框图，该装置700可以驻留在代理服务器120中。如图7所示，装置700包括请求接收模块710、数据查找模块720、第一处理模块730、第二处理模块740和第三处理模块。

[0076] 请求接收模块710接收用户发送的接口请求，并提取该接口请求的请求参数，该请求参数包括域名和路径。请求接收模块710还可以将接口请求的域名对应的IP地址设置为所述代理服务器的IP地址。请求接收模块710可以进行与上面在步骤S310中描述的处理相对应的处理，这里不再展开赘述。

[0077] 数据查找模块720从预先配置好的配置文件中查找接口请求的域名和路径，该配置文件存储有多个回源接口的域名，以及接口管理平台中多个测试接口的域名、路径和对

应的响应内容。数据查找模块720可以进行与上面在步骤S320中描述的处理相对应的处理，这里不再展开赘述。

[0078] 第一处理模块730当接口请求的域名或路径匹配某测试接口时，返回所述代理服务器的IP地址和端口，以进行mock的请求处理。第一处理模块730可以进行与上面在步骤S330中描述的处理相对应的处理，这里不再展开赘述。

[0079] 第二处理模块740当接口请求的域名和路径均不匹配任何测试接口、但该域名匹配某回源接口时，将该请求转发给所匹配上的回源接口进行请求处理。第二处理模块740可以进行与上面在步骤S340中描述的处理相对应的处理，这里不再展开赘述。

[0080] 第三处理模块750当接口请求的域名和路径均不匹配任何测试接口、且该域名也不匹配任何回源接口时，返回代理服务器的IP地址和端口，以进行mock的请求处理。第三处理模块750可以进行与上面在步骤S340中描述的处理相对应的处理，这里不再展开赘述。

[0081] 根据本发明的一个实施例，装置700还可以包括第四处理模块(图中未示出)，适于在执行页面的边界测试时，识别业务类接口请求，并修改业务了接口请求的响应内容。其具体识别和修改过程，已在基于方法300的描述中详细公开，这里不再展开赘述。

[0082] 根据本发明的技术方案，直接配置本地host即可使用，不用修改源代码和源配置，无代码侵入性，使用成本低且使用方便，路由转发灵活控制；支持灵活的mock方式，mock结果数据页面python动态编程，允许使用方在页面写python脚本进行动态mock；支持对品牌广告模版这样的通用场景，进行线上录制，线下一键关联即可完成复杂数据的一键mock关联。本发明能够以较低的开发和维护成本应用在大多数mock使用的场景。

[0083] A8、如A6所述的方法，其中，所述接口请求为广告接口请求，所提取的请求参数还包括广告位标识，该广告接口请求所对应的mock响应数据为该广告位标识所对应的广告数据。A9、如A1-A8中任一项所述的方法，其中，所述配置文件存储在redis数据库中，所述代理服务器通过访问所述redis数据库来从预先配置好的配置文件中查找所述接口请求的域名和路径。

[0084] 这里描述的各种技术可结合硬件或软件，或者它们的组合一起实现。从而，本发明的方法和设备，或者本发明的方法和设备的某些方面或部分可采取嵌入有形媒介，例如可移动硬盘、U盘、软盘、CD-ROM或者其它任意机器可读的存储介质中的程序代码(即指令)的形式，其中当程序被载入诸如计算机之类的机器，并被所述机器执行时，所述机器变成实践本发明的设备。

[0085] 在程序代码在可编程计算机上执行的情况下，计算设备一般包括处理器、处理器可读的存储介质(包括易失性和非易失性存储器和/或存储元件)，至少一个输入装置，和至少一个输出装置。其中，存储器被配置用于存储程序代码；处理器被配置用于根据该存储器中存储的所述程序代码中的指令，执行本发明的请求处理方法。

[0086] 以示例而非限制的方式，可读介质包括可读存储介质和通信介质。可读存储介质存储诸如计算机可读指令、数据结构、程序模块或其它数据等信息。通信介质一般以诸如载波或其它传输机制等已调制数据信号来体现计算机可读指令、数据结构、程序模块或其它数据，并且包括任何信息传递介质。以上的任一种的组合也包括在可读介质的范围之内。

[0087] 在此处所提供的说明书中，算法和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。各种通用系统也可以与本发明的示例一起使用。根据上面的描述，构造这类

系统所要求的结构是显而易见的。此外，本发明也不针对任何特定编程语言。应当明白，可以利用各种编程语言实现在此描述的本发明的内容，并且上面对特定语言所做的描述是为了披露本发明的最佳实施方式。

[0088] 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被实践。在一些实例中，并未详细示出公知的方法、结构和技术，以便不模糊对本说明书的理解。

[0089] 类似地，应当理解，为了精简本公开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。然而，并不应将该公开的方法解释成反映如下意图：即所要求保护的本发明要求比在每个权利要求中所明确记载的特征更多特征。更确切地说，如下面的权利要求书所反映的那样，发明方面在于少于前面公开的单个实施例的所有特征。因此，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。

[0090] 本领域那些技术人员应当理解在本文所公开的示例中的设备的模块或单元或组件可以布置在如该实施例中所描述的设备中，或者可替换地可以定位在与该示例中的设备不同的一个或多个设备中。前述示例中的模块可以组合为一个模块或者此外可以分成多个子模块。

[0091] 本领域那些技术人员可以理解，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，以及此外可以把它们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，可以采用任何组合对本说明书(包括伴随的权利要求、摘要和附图)中公开的所有特征以及如此公开的任何方法或者设备的所有过程或单元进行组合。除非另外明确陈述，本说明书(包括伴随的权利要求、摘要和附图)中公开的每个特征可以由提供相同、等同或相似目的的替代特征来代替。

[0092] 此外，本领域的技术人员能够理解，尽管在此所述的一些实施例包括其它实施例中所包括的某些特征而不是其它特征，但是不同实施例的特征的组合意味着处于本发明的范围之内并且形成不同的实施例。例如，在下面的权利要求书中，所要求保护的实施例的任意之一都可以以任意的组合方式来使用。

[0093] 此外，所述实施例中的一些在此被描述成可以由计算机系统的处理器或者由执行所述功能的其它装置实施的方法或方法元素的组合。因此，具有用于实施所述方法或方法元素的必要指令的处理器形成用于实施该方法或方法元素的装置。此外，装置实施例的在此所述的元素是如下装置的例子：该装置用于实施由为了实施该发明的目的的元素所执行的功能。

[0094] 如在此所使用的那样，除非另行规定，使用序数词“第一”、“第二”、“第三”等等来描述普通对象仅仅表示涉及类似对象的不同实例，并且并不意图暗示这样被描述的对象必须具有时间上、空间上、排序方面或者以任意其它方式的给定顺序。

[0095] 尽管根据有限数量的实施例描述了本发明，但是受益于上面的描述，本技术领域内的技术人员明白，在由此描述的本发明的范围内，可以设想其它实施例。此外，应当注意，

本说明书中使用的语言主要是为了可读性和教导的目的而选择的，而不是为了解释或者限定本发明的主题而选择的。因此，在不偏离所附权利要求书的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。对于本发明的范围，对本发明所做的公开是说明性的而非限制性的，本发明的范围由所附权利要求书限定。

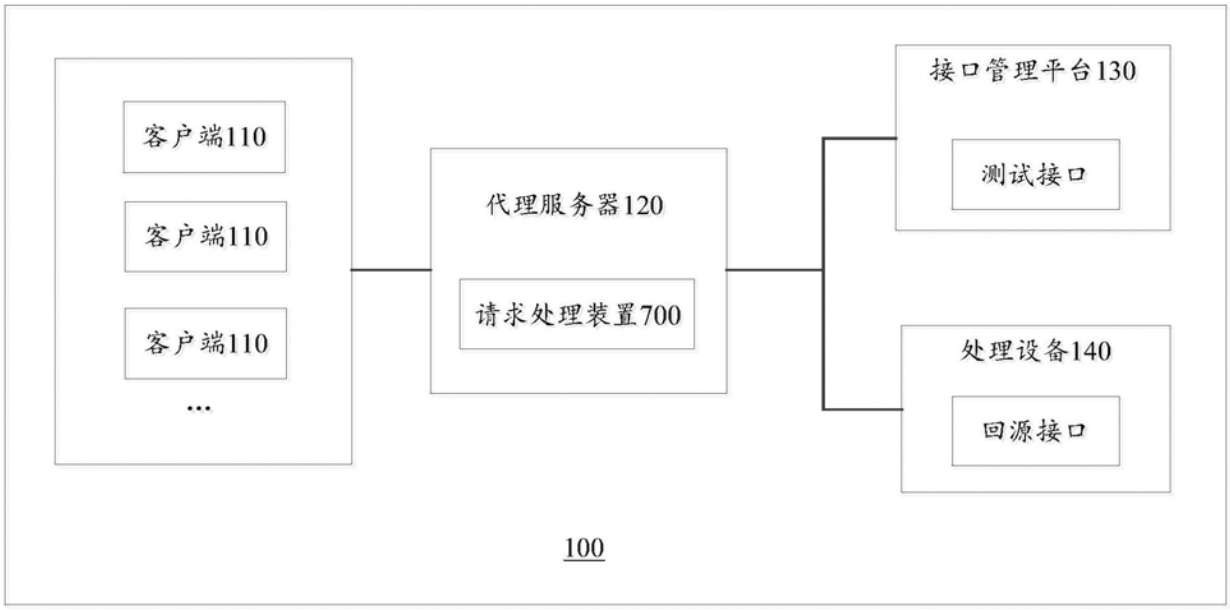

客户端110
客户端110
客户端110
...
代理服务器120
请求处理装置700
接口管理平台130
测试接口
处理设备140
回源接口
100

图1

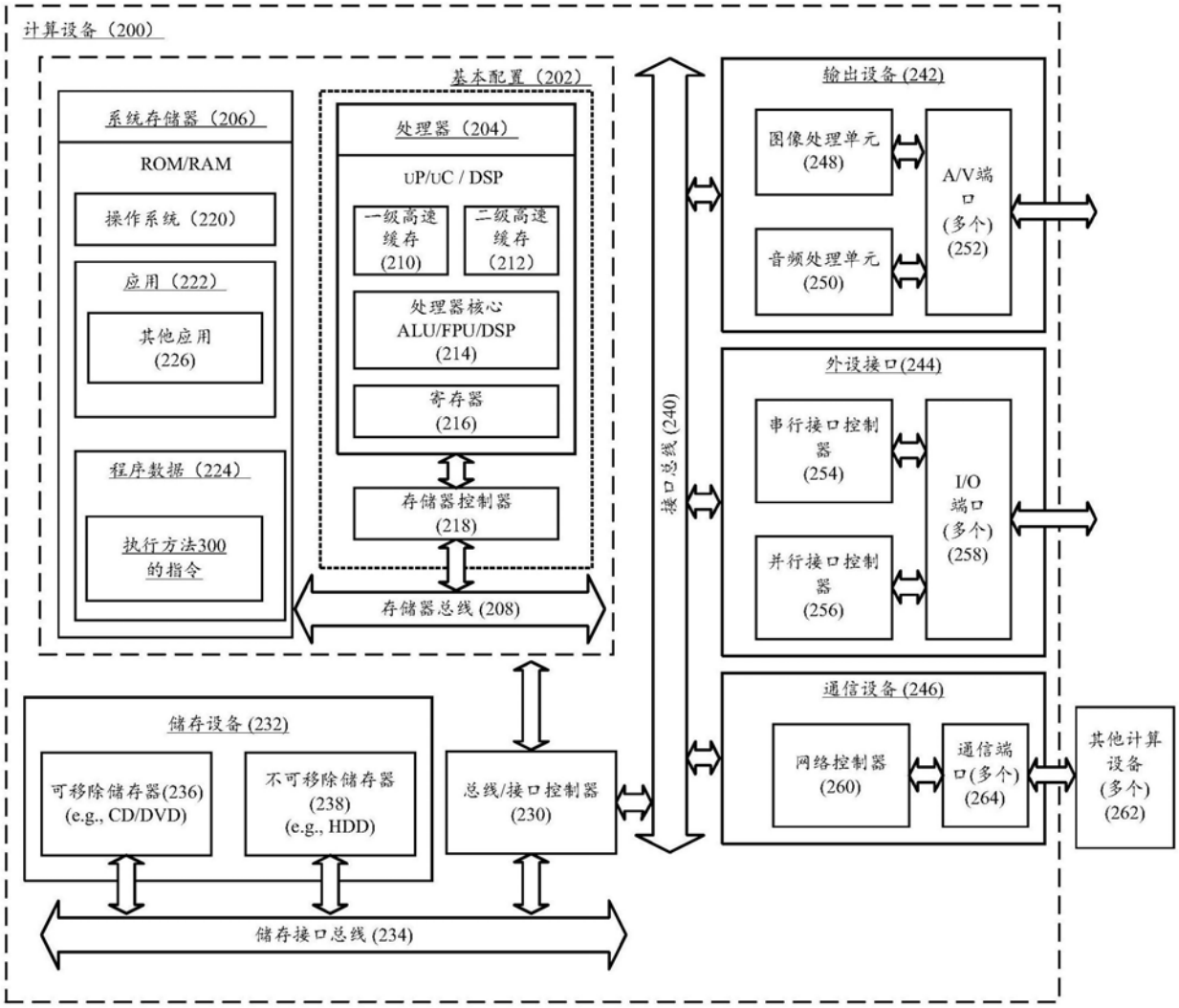

计算设备（200）
基本配置（202）
系统存储器（206）
ROM/RAM
操作系统（220）
应用（222）
其他应用
(226)
程序数据（224）
执行方法300
的指令
处理器（204）
UP/UC / DSP
一级高速
缓存
(210)
二级高速
缓存
（212）
处理器核心
ALU/FPU/DSP
(214)
寄存器
(216)
存储器控制器
(218)
存储器总线 (208)
接口总线 (240)
输出设备 (242)
图像处理单元
(248)
音频处理单元
(250)
A/V端
口
(多个)
(252)
外设接口(244)
串行接口控制
器
(254)
并行接口控制
器
(256)
I/O
端口
(多个)
(258)
通信设备 (246)
网络控制器
(260)
通信端
口(多个)
(264)
其他计算
设备
(多个)
(262)
储存设备 (232)
可移除储存器(236)
(e.g., CD/DVD)
不可移除储存器
(238)
(e.g., HDD)
总线/接口控制器
(230)
储存接口总线 (234)

图2

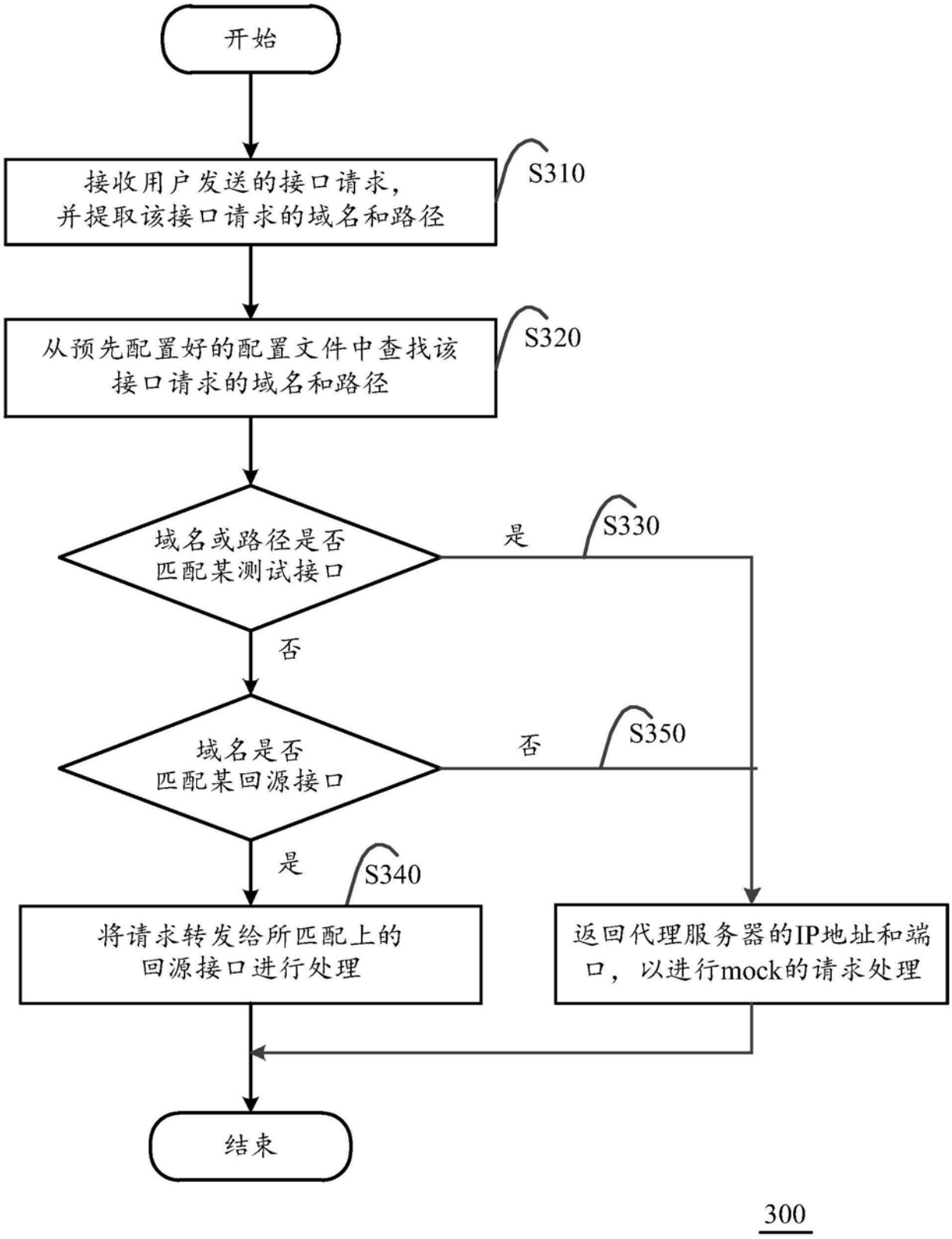

开始
接收用户发送的接口请求，
并提取该接口请求的域名和路径
S310
从预先配置好的配置文件中查找该
接口请求的域名和路径
S320
域名或路径是否
匹配某测试接口
是
S330
否
域名是否
匹配某回源接口
否
S350
是
S340
将请求转发给所匹配上的
回源接口进行处理
返回代理服务器的IP地址和端
口，以进行mock的请求处理
结束
300

图3


* api（域名+路径+url参数）
http://testmock.com/test?param=123&a=111
* api类型：（get post）
get
* content type
application/json
匹配规则
${param}=param|${ab}=a
测试host设置
无
无：走mock逻辑输出mock结果，选择host配置：走host配置，输出host的结果
* 动态response编码
否
* response
{
"result": "hello mock param:${param},a:${ab}"
}

图4


* response
res = {"data":[],"status":0,"statusInfo": {"global": "OK"}}
id_arr = ids.split(',')
if not id_arr or len(id_arr) <= 0:
res = ''
else:
for id in id_arr:
arr = {}
arr['id'] = id
arr['exposeNum'] = 1000
arr['clickNum'] = 900
res['data'].append(arr)
print json.dumps(res)
ids对应是mock规则配置的等号后面，例如${ids}=ids,可以直接被编程代码拿过来做变量使用
print的结果就是所需要的response

图5


修改广告位
* 模版选择
模版ID(533)，PC首页自动下推通栏模板
* 广告位id
5471
* 广告位名称
通栏1.1(A/B/C)(2018年6月启用)
描述
首页
立即修改
返回

图6

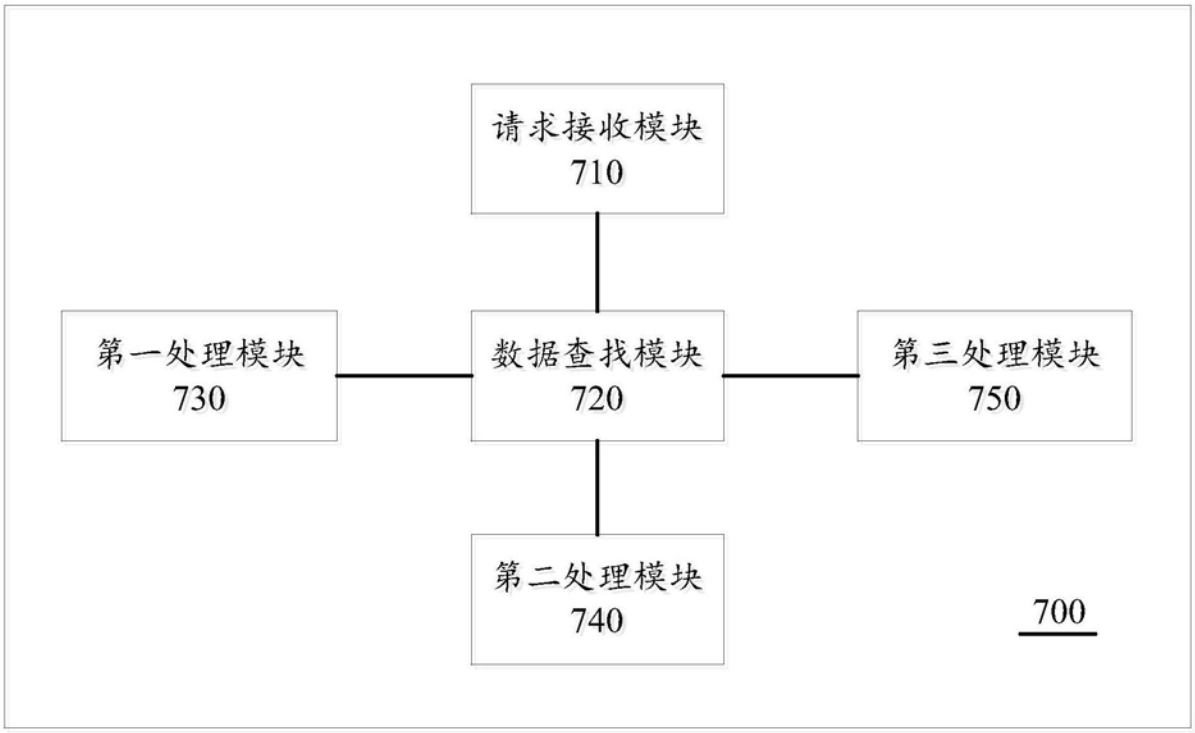

请求接收模块
710
第一处理模块
730
数据查找模块
720
第三处理模块
750
第二处理模块
740
700

图7