



(12) 发明专利申请

(10) 申请公布号 CN 113688025 A

(43) 申请公布日 2021. 11. 23

(21) 申请号 202111045953.6

(22) 申请日 2021.09.07

(71) 申请人 中国联合网络通信集团有限公司  
地址 100033 北京市西城区金融大街21号

(72) 发明人 胡鹏强

(74) 专利代理机构 北京同立钧成知识产权代理有限公司 11205

代理人 张晓霞 黄健

(51) Int. Cl.

G06F 11/34 (2006.01)

G06F 11/36 (2006.01)

G06F 16/25 (2019.01)

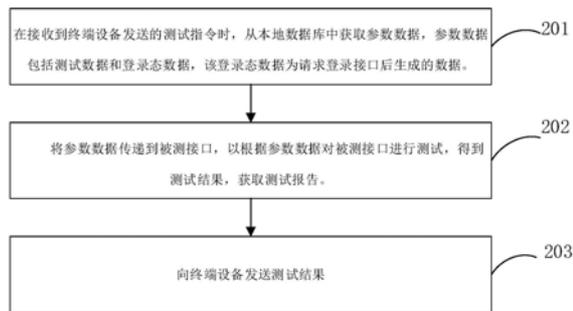
权利要求书2页 说明书11页 附图2页

(54) 发明名称

接口测试方法、装置、设备和存储介质

(57) 摘要

本申请提供一种接口测试方法、装置、设备和存储介质。该方法中,在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,其中参数数据包括测试数据和登录态数据,登录态数据为请求登录接口后生成的数据,然后将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果,最后向终端设备发送测试结果。本申请的方法能够减少测试人员工作量;提高了接口压测效率和准确率。



1. 一种接口测试方法,其特征在于,包括:

在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,所述参数数据包括测试数据和登录态数据,所述登录态数据为请求登录接口后生成的数据;

将所述参数数据传递到被测接口,以根据所述参数数据对所述被测接口进行测试,得到测试结果;

向所述终端设备发送所述测试结果。

2. 根据权利要求1所述的方法,其特征在于,所述根据所述参数数据对所述被测接口进行测试,得到测试结果,包括:

从所述本地数据库中调用与所述被测接口相关的业务数据;

根据所述参数数据和所述业务数据,对所述被测接口进行测试,得到测试结果。

3. 根据权利要求2所述的方法,其特征在于,所述获取与所述被测接口相关的业务数据之前,所述方法还包括:

确定所述被测接口的执行逻辑;

按照所述执行逻辑,将与所述被测接口相关的业务数据存储到所述本地数据库中。

4. 根据权利要求1-3任一项所述的方法,其特征在于,所述在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据之前,所述方法还包括:

获取所述测试数据;

根据所述测试数据调用所述登录接口,以生成所述登录态数据;

根据所述测试数据和所述登录态数据,生成所述参数数据,并将所述参数数据存储到所述本地数据库中。

5. 根据权利要求4所述的方法,其特征在于,所述获取所述测试数据,包括:

接收所述终端设备发送的用于生成测试数据的请求消息,所述请求消息中包括第一数量和类型信息;

根据所述类型信息,随机生成所述第一数量个所述测试数据。

6. 根据权利要求1-3任一项所述的方法,其特征在于,所述方法还包括:

在输出所述测试结果后的预设时间段后,删除所述本地数据库中的所述参数数据。

7. 根据权利要求1-3任一项所述的方法,其特征在于,所述根据所述参数数据对所述被测接口进行测试,得到测试结果,包括:

按照预先配置的并发线程数量,根据所述参数数据对所述被测接口进行测试,得到测试结果。

8. 一种接口测试装置,其特征在于,包括:

获取模块,用于在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,所述参数数据包括测试数据和登录态数据,所述登录态数据为请求登录接口后生成的数据;

处理模块,用于将所述参数数据传递到被测接口,以根据所述参数数据对所述被测接口进行测试,得到测试结果;

发送模块,用于向所述终端设备发送所述测试结果。

9. 一种终端设备,其特征在于,包括处理器和存储器;其中,

所述存储器,用于存储所述处理器可执行指令;

所述处理器,用于读取所述存储器存储的计算机程序,并根据所述存储器中的计算机程序执行上述权利要求1-7任一项所述的接口测试方法。

10.一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执行指令,当处理器执行所述计算机执行指令时,实现上述权利要求1-7任一项所述的接口测试方法。

## 接口测试方法、装置、设备和存储介质

### 技术领域

[0001] 本申请涉及通信技术,尤其涉及一种接口测试方法、装置、设备和存储介质。

### 背景技术

[0002] 为了保证系统能够准确、稳定的运行,通常需要对系统的业务场景进行性能测试。而接口作为各系统和模块间实现通信的桥梁,因此对系统的业务场景进行测试时,最重要的是对业务接口进行测试,如对业务接口进行性能压测,对业务接口进行性能压测能够帮助用户及时发现业务系统的潜在风险和性能瓶颈。

[0003] 在相关技术中,很多业务接口都需要在登录的前提下才能请求成功,因此,在对登录态下的业务接口进行性能压测时,必须先请求登录接口,才能将获取的登录态数据传递给被测的业务接口,从而实现对业务接口的性能压测。

[0004] 然而,上述对登录态下业务接口进行压测的方法,在每次压测业务接口时,都需要先调用登录接口来获取登录态数据,从而使得业务接口的压测效率较低。

### 发明内容

[0005] 为解决现有技术中存在的问题,本申请提供一种接口测试方法、装置、设备和存储介质,不仅可以减小资源消耗,而且可以减少测试时间,提高测试效率,另外还可以聚焦目标被测接口,提高结果准确率。

[0006] 一方面,本申请提供一种接口测试方法,应用于终端设备,包括:

[0007] 在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,参数数据包括测试数据和登录态数据,登录态数据为请求登录接口后生成的数据;

[0008] 将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果;

[0009] 向终端设备发送测试结果。

[0010] 在一种可能的实现方式中,根据参数数据对被测接口进行测试,得到测试结果,包括:

[0011] 从本地数据库中调用与被测接口相关的业务数据;

[0012] 根据参数数据和业务数据,对被测接口进行测试,得到测试结果。

[0013] 在一种可能的实现方式中,获取与被测接口相关的业务数据之前,方法还包括:

[0014] 确定被测接口的执行逻辑;

[0015] 按照执行逻辑,将与被测接口相关的业务数据存储到本地数据库中。

[0016] 在一种可能的实现方式中,在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据之前,方法还包括:

[0017] 获取测试数据;

[0018] 根据测试数据调用登录接口,以生成登录态数据;

[0019] 根据测试数据和登录态数据,生成参数数据,并将参数数据存储到本地数据库中。

- [0020] 在一种可能的实现方式中,获取测试数据,包括:
- [0021] 接收终端设备发送的用于生成测试数据的请求消息,请求消息中包括第一数量和类型信息;
- [0022] 根据类型信息,随机生成第一数量个测试数据。
- [0023] 在一种可能的实现方式中,方法还包括:
- [0024] 在输出测试结果后的预设时间段后,删除本地数据库中的参数数据。
- [0025] 在一种可能的实现方式中,根据参数数据对被测接口进行测试,得到测试结果,包括:
- [0026] 按照预先配置的并发线程数量,根据参数数据对被测接口进行测试,得到测试结果。
- [0027] 第二方面,本申请实施例还提供了一种接口测试装置,包括:
- [0028] 获取模块,用于在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,参数数据包括测试数据和登录态数据,登录态数据为请求登录接口后生成的数据;
- [0029] 处理模块,用于将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果;
- [0030] 发送模块,用于向终端设备发送测试结果。
- [0031] 在一种可能实现的方式中,处理模块,具体用于:从本地数据库中调用与被测接口相关的业务数据;
- [0032] 根据参数数据和业务数据,对被测接口进行测试,得到测试结果。
- [0033] 在一种可能实现的方式中,处理模块,具体用于:确定被测接口的执行逻辑;
- [0034] 按照执行逻辑,将与被测接口相关的业务数据存储到本地数据库中。
- [0035] 在一种可能实现的方式中,获取模块,还用于获取测试数据;
- [0036] 处理模块,还用于根据测试数据调用登录接口,以生成登录态数据;
- [0037] 处理模块,还用于根据测试数据和登录态数据,生成参数数据,并将参数数据存储到本地数据库中。
- [0038] 在一种可能实现的方式中,获取模块,具体用于:接收终端设备发送的用于生成测试数据的请求消息,请求消息中包括第一数量和类型信息;
- [0039] 根据类型信息,随机生成第一数量个测试数据。
- [0040] 在一种可能实现的方式中,处理模块,还用于在输出测试结果后的预设时间段后,删除本地数据库中的参数数据。
- [0041] 在一种可能实现的方式中,处理模块,还用于按照预先配置的并发线程数量,根据参数数据对被测接口进行测试,得到测试结果。
- [0042] 第三方面,本申请实施例还提供了一种终端设备,该终端设备可以包括处理器和存储器;其中,
- [0043] 存储器,用于存储处理器可执行指令;
- [0044] 处理器,用于读取存储器存储的计算机程序,并根据存储器中的计算机程序执行上述第一方面任一种可能的实现方式中接口测试方法。
- [0045] 第四方面,本申请实施例还提供了一种计算机可读存储介质,计算机可读存储介质中存储有计算机执行指令,当处理器执行计算机执行指令时,实现上述第一方面任一种

可能的实现方式中的接口测试方法。

[0046] 本申请提供的接口测试方法、装置、设备和存储介质,通过接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,参数数据包括测试数据和登录态数据,登录态数据为请求登录接口后生成的数据,然后将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果,再向终端设备发送测试结果。由于直接将用户经过一次请求登录接口后生成的登录态数据与测试数据组合生成参数数据传入被测业务接口进行压测,避免了在每次对被测接口进行测试时都需要先调用登录接口才能获取到登录态数据的问题,从而在执行压测时将业务接口与登录接口分离,使得压测的全部焦点放在业务接口,减少了测试任务的工作量,提高了业务接口压测效率。

### 附图说明

[0047] 此处的附图被并入说明书中并构成本说明书的一部分,示出了符合本申请的实施例,并与说明书一起用于解释本申请的原理。

[0048] 图1为本申请实施例提供的接口测试方法的系统架构图;

[0049] 图2为本申请实施例提供的一种接口测试方法的流程示意图;

[0050] 图3为配置通用压测脚本示意图;

[0051] 图4为本申请实施例提供的一种接口测试装置的结构示意图;

[0052] 图5为本申请实施例提供的一种终端设备的结构示意图。

[0053] 通过上述附图,已示出本申请明确的实施例,后文中将有更详细的描述。这些附图和文字描述并不是为了通过任何方式限制本申请构思的范围,而是通过参考特定实施例为本领域技术人员说明本申请的概念。

### 具体实施方式

[0054] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本申请相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本申请的一些方面相一致的装置和方法的例子。

[0055] 本申请实施例提供的接口测试方法,可以应用于对系统中的业务接口进行压测的场景中,尤其可以应用于对处于登录态的业务接口进行性能压测的场景中。其中,对处于登录态的业务接口进行性能压测,是指通过大量业务参数数据、被测系统里相关联的业务数据,对业务接口进行大批量并发请求,最后分析所有的请求的返回结果以及被测系统状态是否符合预期。通常在某个系统上线之前,为了保证该系统的正常运行,便于提前发现系统中存在的缺陷和问题,都会先对该系统中的各个业务接口进行压测,在压测通过之后再提交上线。另外,在系统的正常运行或者维护过程中,可能也经常需要对业务接口的性能进行压测。

[0056] 以购物系统中对购物应用程序(Application,APP)中选购所要购买的商品后请求提交订单的业务接口进行压测为例,用户在选购所需商品后提交订单时需要在用户登录的状态下才可以进行。当在大量用户同时登录并提交订单的情况下,服务器需要保证每一位用户提交订单的业务接口稳定、准确的返回请求结果。因此在该APP上线前或者日常定期维

护中压测该提交订单业务接口,以进行性能瓶颈和存在的潜在风险分析对于该APP的稳定运行是非常必要的。

[0057] 图1为本申请实施例提供的一种接口测试方法的系统架构图,如图1所示,该系统中包括终端设备11可以与服务器12进行数据通信,用户可以通过终端设备11进行压测脚本和参数的配置。在一些实施例中,终端设备11还可以根据用户配置的信息,随机生成测试数据,以用于对被测接口进行压测或测试。另外,服务器12中部署有数据库13,其中,数据库13可以为Redis数据库、SQL数据库等,该数据库13中存储有参数数据,该参数数据包括测试数据和登录态数据。

[0058] 应该理解,在上述系统架构中的用户、终端设备11、服务器12以及数据库13的数目仅仅是示例性的,更多或更少的数量都属于本申请的保护范畴。并且,在上述系统中,终端设备11例如可以是个人计算机、服务器、平板、手机、PDA、笔记本或其它任何具有联网功能的计算设备。而服务器12则可以利用具有更强大处理能力和更高安全性的一个服务器或服务器组来实现。而它们之间所使用的网络可以包括各种类型的有线和无线网络,例如但不限于:互联网、局域网、WIFI、WLAN、蜂窝通信网络(GPRS、CDMA、2G/3G/4G/5G蜂窝网络)、卫星通信网络等等。

[0059] 现有的接口性能测试方法主要采用两种方案,一种是使用开源压测工具,开发、配置好性能测试脚本,并根据业务场景的实际需求准备测试数据,再通过压测命令执行脚本进行压力测试,最后打开压测报告进行查看分析;另一种是自主研发性能测试平台,开发符合定制化需求的功能,从而开展相关业务的性能测试。自主研发的性能测试平台,主要有两种,一种是基于主流开源工具进行的二次封装、二次开发,另一种是基于开发语言提供的类和方法进行研发。在实际的业务场景中,大部分业务接口均需要保证用户在登录的前提下才能完成成功的请求,一次成功的请求,是指被测系统认为用户已经登录,并且传入的业务参数、系统里相关联的业务数据都正确,然后执行相关业务逻辑后,返回正确的响应给到请求方。因此执行压测脚本对被测业务接口进行压测时需先调用登录接口获取该用户的登录态数据,并将登录态数据作为被测接口的业务参数数据传入。

[0060] 然而,不论是上述的哪一种测试方式,在对业务接口进行压测时,需要先请求登录接口,获取到登录态数据后,再请求业务接口,从而存在接口依赖的情况。业内常采用的方法,一种是压测业务接口时连同登录接口一起压测;另一种是测试人员也可以开发功能模块对接口进行改动,使之不鉴权即可实现压测接口。然而,对开发功能模块对接口进行改动耗费精力和时间,且效率低、工作量大;而压测业务接口时连同登录接口一起压测的方式,使登录接口易成为性能瓶颈,从而无法真正压测到业务接口,直接影响了业务接口的压测效率,导致系统资源的浪费。

[0061] 发明人在实现本申请的过程中发现,引起业务接口压测效率下降的原因,主要是因为登录接口与被测业务接口的接口依赖情况,基于上述问题,本申请中将登录接口和被测业务接口的处理分离,使得在压测业务接口时无需再次调取登录接口,只对业务接口进行压测性能分析,从而实现真正意义上的业务接口压测性能分析,直接提高了业务接口的压测效率。

[0062] 基于上述构思,本申请实施例提供一种接口测试方法,在该测试方法中,在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,该参数数据包括测试数据和

登录态数据,其中,该登录态数据为请求登录接口后生成的数据,然后将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果,并将测试结果发送给终端设备。由于在对被测接口进行测试时,可以从本地数据库中直接获取到登录态数据,而无需调用登录接口获得该登录态数据,由此可以解除登录接口与业务接口的依赖关系,从而达到了仅对被测业务接口进行压测的目的,由此可以提高业务接口测试的效率。

[0063] 下面以具体地实施例对本申请的技术方案以及本申请的技术方案如何解决上述技术问题进行详细说明。下面这几个具体的实施例可以相互结合,对于相同或相似的概念或过程可能在某些实施例中不再赘述。下面将结合附图,对本申请的实施例进行描述。

[0064] 图2为本申请实施例提供的一种接口测试方法的流程示意图,本申请实施例提供了一种接口测试方法,该方法可以由任意执行接口测试方法的装置来执行,该装置可以通过软件和/或硬件实现。本实施例中,该装置可以集成在如图1所示的服务器12中。如图2所示,本申请实施例提供的接口测试方法包括如下步骤:

[0065] 步骤201:在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,参数数据包括测试数据和登录态数据,该登录态数据为请求登录接口后生成的数据。

[0066] 在本步骤中,测试指令用于指示服务器对被测接口的性能进行测试或者压测。用户在对被测业务接口进行测试时,可以向终端设备发送测试指令,终端设备再将接收到的测试指令发送给服务器。其中,被测业务接口是等待被压测分析的业务接口。

[0067] 其中,本地数据库是服务器中的数据库,在使用压测工具进行接口压测时会与本地数据库有数据的交互,例如:Redis缓存数据库、SQL数据库等或者其他的数据库。其中,压测工具是一种帮助测试人员在模拟真实场景下预测业务系统性能和稳定性等性能的工具,例如:LoadRunner、Apache Jmeter以及其他用于测试系统性能的工具。

[0068] 具体的,服务器在接收到终端设备发送的测试指令时,将从本地数据库中获取参数数据,该参数数据中包括有测试数据和登录态数据,其中,该登录态数据为服务器在通过测试数据调用登录接口后生成的数据。示例性的,该登录接口是用户提供身份信息请求访问系统内部资源的接口。

[0069] 应理解,在从本地数据库中获取参数数据之前,需要先生成测试数据和登录态数据,从而将根据该测试数据和登录态数据生成的参数数据保存到本地数据库中。示例性的,服务器通过获取测试数据,并根据测试数据调用登录接口,以生成登录态数据,并根据测试数据和登录态数据,生成参数数据,并将参数数据存储到本地数据库中。

[0070] 进一步的,测试数据是模拟业务场景下生成的多行信息数据,每一行代表一个用户信息。测试数据主要包含两方面数据,一是被测业务接口的传参数据,二是被测系统中相关联的业务数据。测试数据会通过调用服务器中的测试数据生成模块自动生成,调用测试数据生成模块后会默认生成若干行测试数据,例如:默认生成1000行,其中的每行数据均包含手机号、随机字母+数字组成的15位随机值。

[0071] 可选的,用户也可以根据实际的测试需求,生成不同的测试数据。示例性的,服务器通过接收终端设备发送的用于生成测试数据的请求消息,该请求消息中包括第一数量和类型信息,然后根据类型信息,随机生成第一数量个测试数据。

[0072] 同时,如果需要指定生成测试数据的数量,只需要在调用测试数据生成模块时传入该数量即可。如果需要生成其他位数的随机值,只需要传入位数即可。如果需要生成更多

个随机值,只需要传入表示随机值数的值即可。

[0073] 另外,测试人员也可根据公司业务或者测试需求,在测试数据生成方法的列表中增加要生成测试数据类型,其中测试数据类型是原测试数据中没有的类型信息,例如:联系电话、家庭住址、是否已婚等类型信息。对于业务系统而言,测试数据一般都是由唯一确定的随机值组成,所以此测试数据生成方法具有通用性。现有技术中,测试人员在接口压测前需要花费较长的时间准备测试数据,且针对不同的业务场景和测试需求手动增加或减少相应的数据数量或类型,不具有通用性。对此,按照本申请实施例中自动生成测试数据的方式,可根据用户的实际需求增加或减少测试数据的数量和类型信息,便可解决测试数据准备困难的问题,间接提高了业务接口的压测效率,降低人力成本。

[0074] 在获取到测试数据后,将以测试数据作为登录接口的参数并请求登录接口后,即可获取到登录态数据,数据中记录了用户的登录状态。如果登录态数据在有效期内,代表用户已经成功登录;否则,就需要重新访问登录接口获取登录态数据。

[0075] 具体的,登录态数据的获取方式包括如下三种方式:一种是通过校验密码获取到登录态数据;一种是通过检验短信验证码获得登录态数据;还有一种是通过传入已有的第三方登录态信息获取业务系统的登录态数据。

[0076] 在被测业务接口测试过程中,为了简化登录流程,对于第一种登录方式,只需结合测试数据和登录态数据生成相对应的正确密码即可;对于第二种登录方式,需要改造系统的登录接口为无需校验验证码即可成功登录;对于第三种登录方式,需要改造系统的登录接口,使之允许通过传入唯一的id来模拟已有登录态数据。同时对于系统登录接口的改造是在在测试环境中改造的,这个改造不会影响真实生产环境中的业务逻辑。

[0077] 进一步的,测试数据中含有预设的多个用户信息,与多用户按顺序请求登录接口获取到登录态数据组合,生成新的参数数据。其中,参数数据中保存了完整的测试数据和登录态数据,例如:手机号、唯一标识id、token、cookie等信息。

[0078] 在生成参数数据之后,可以将参数数据存储到本地数据库中,并在执行压测时从本地数据库中获取参数数据以传入到被测接口。

[0079] 在本实施例中,可以根据测试数据和登录态数据生成的参数数据存储到本地数据库中,这样,在进行被测接口的测试时,不用每次都调用登录接口获取登录态数据,而是直接从本地数据库中获取即可,由此可以提高测试效率。

[0080] 步骤202:将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果。

[0081] 在本步骤中,测试结果是指通过压测脚本对服务器的服务性能和被测接口的压测性能进行性能测试所获得的结果。服务性能包括在对被测接口进行压测时服务器的性能,例如:服务器的内存、中央处理器(Central Processing Unit,CPU)等。被测接口的压测性能是指对被测接口进行压测时,被测接口的性能,例如:被测接口的响应时间、错误率、每秒传输的事物处理个数(Transactions Per Second,TPS)等。

[0082] 在一种可能的实现方式中,在根据参数数据对被测接口进行测试时,可以从本地数据库中调用与被测接口相关联的业务数据,并根据参数数据和业务数据,对被测接口进行测试,得到测试结果。

[0083] 具体的,与被测接口相关联的业务数据,是指在请求被测接口前必需具备的一些

业务数据。例如：购物APP在提交支付之前，需在购物车中添加待支付的商品且待支付的商品库存不为0。在该场景下购物车中的商品便对应被测接口的相关联业务数据。

[0084] 服务器在从本地数据库中调用与被测接口相关的业务数据后，可以将获取的参数数据和该业务数据，一起传递到被测接口，以对该被测接口进行测试，并接收返回的测试结果，从而根据测试结果分析被测接口是否符合预期。由于直接从本地数据库中调用与被测接口相关的业务数据，避免了在每次进行接口测试时，都需要生成业务数据的现象，从而可以提高测试的效率。

[0085] 可以理解的是，为了在对被测接口进行测试时，能及时准确的从本地数据库中调取与被测接口相关的业务数据，则需要预先将这些业务数据存储到本地数据库中。示例性的，可以预先确定被测接口的执行逻辑，并按照该执行逻辑，将与被测接口相关的业务数据存储到本地数据库中。

[0086] 具体的，被测接口的相关联业务数据是通过调用数据插入模块将被测系统在执行被测接口时所需的相关联业务数据提前插入到系统缓存或者数据库中的。其中，相关联接业务数据是通过固定格式的前缀叠加标识id，变为被测业务接口需要插入的缓存数据格式插入到系统缓存或者数据库中，其中标识id从测试数据中读取；需要插入的缓存数据格式则是传入被测业务接口需要的数据格式，一般来说，代表同一业务含义的缓存数据前缀都是一致的。例如：代表商品已经加入购物车的缓存数据格式为product\_in\_car\_xxx，其中，xxx是商品的标识id，product\_in\_car为前缀。同时在相关联接业务数据被插入缓存时，系统会自动设置缓存的过期时间，测试人员可以按照自己的需要，调整缓存保留的时间，过期后的缓存数据会自动清理，从而释放服务器内存空间，避免占用系统资源。

[0087] 更进一步的，如果将相关联接业务数据插入数据库，则先配置要插入的库和表、要插入的数据的SQL字符串；如果将相关联接业务数据插入系统缓存中，需要先配置要插入的缓存的key和对应value的字符串值，如果有多个缓存需要插入，则配置多对key-value即可。在配置缓存字符串或者SQL字符串时，一定需组合或者拼接通过调用测试数据生成模块自动生成的测试数据，使插入系统的数据成为有效的测试数据。将相关联接业务数据提前插入到系统缓存或者数据库以便在压测被测接口时，系统能够按照预想的逻辑正确处理请求。

[0088] 进一步的，在对被测接口进行测试时，需要使用压测工具打开通用压测脚本配置并运行。其中，通用压测脚本是开发人员预先配置好的用于进行压力测试的脚本。

[0089] 具体的，图3为对被测接口进行测试的示意图，如图3所示，在进行压测之前，需要预先对服务器配置需要压测的被测接口，并在压测工具中配置需要压测的业务接口的请求地址url、请求参数、响应参数、用token、cookie两个变量传递给被测业务接口的信息头，并将请求头中的cookie和token的值分别用\${cookie}、\${token}代替等信息，最后配置希望达到的并发线程数。其中，用token、cookie两个变量传递登录态信息给被测业务接口。通用压测脚本，默认是所有并发线程在1秒内启动，如果需要模拟一段时间内逐渐加大测试的压力，则可以配置为非1秒启动，例如：配置为10秒内启动10000个线程。配置完成后，点击保存，即可完成压测脚本的配置。

[0090] 进一步的，配置完成测试脚本后便可调用压测脚本执行模块方法运行压测脚本，并按照预先配置的并发线程数量，根据参数数据对被测接口进行测试，得到测试结果。在此方法中，会启动一个cmd或shell进程，通过该进程执行启动压测的命令。其中，cmd或shell

进程是系统中底层具有的基本功能的程序。

[0091] 在本实施例中,按照预先配置的多个并发线程,对被测接口进行测试,和单进程测试的方式相比,该方式可以提高测试的效率。

[0092] 更进一步的,在运行压测过程中会将实时生成的日志输出到日志文件,在压测完成后将含有本次压测的结果信息输出为报告并将报告保存为网页形式内容的文件。压测结束后生成的文件将调用报告提取模块方法自动启动浏览器并打开生成的测试报告网页文件,以便测试人员通过自动打开的结果报告进行被测业务接口的压测性能分析。

[0093] 步骤203:向终端设备发送测试结果。

[0094] 其中,测试报告是根据被测业务接口经过压测脚本执行模块生成包含测试结果信息的报告。

[0095] 在本步骤中,经过执行压测脚本后产生的测试报告会返回到终端设备供测试人员对被测接口的性能进行分析。同时按照本地数据库情况和测试需求实现数据清除。

[0096] 在本申请中,测试人员在打开通用压测脚本并配置好被测业务接口信息后点击执行按钮,服务器便可自动化的、连续性的调用测试数据生成模块、数据插入模块、参数数据生成模块、打开通用压测脚本并调用压测脚本执行模块以及报告提取模块。实现一键启动就可以完成从数据准备到测试完成并查看测试报告的整个过程,使压测变得轻松、稳定、高效。

[0097] 进一步的,因性能测试需要在系统中准备大量相关联的业务数据,每次测试完成后,都会给系统或数据库产生大量脏数据,并造成存储资源的浪费,容易成为影响系统运行不稳定的因素,因此每次测试完成后,都需要对本次压测产生的数据进行清除。另外,为了保证测试人员有足够的时间对本次测试进行分析,在具体的实现过程中,可以是在输出测试结果后的预设时间段后,删除本地数据库中的参数数据。

[0098] 其中,预设时间段可以根据实际情况或者经验进行设置,例如可以为一天或者12小时等,对于预设时间段的具体数值,本申请实施例对此不做限制。

[0099] 具体的,数据清除工作是通过调用数据清除模块对数据库中因本次压测产生的测试数据进行清除。测试人员可以根据测试需要、数据存的状态等因素自主选择是否要启用数据清除工作对数据进行清除。如果数据库无存储压力,或者需要保留数据以供后续测试分析或问题定位,则可以不启用数据清除模块,在不需要测试数据时,再单独执行数据清除模块清除数据即可。

[0100] 进一步的,执行数据清除模块需要先配置连接数据库的连接串和表名、要执行的delete语句。其中,delete语句中的条件应拼接从参数数据生成模块生成的参数数据文件的信息,从而只删除由此次压测造成的数据。

[0101] 在本实施例中,通过自主选择数据清理为测试人员提供了更加人性化的体验,进一步提高了压测效率。

[0102] 本申请实施例提供的接口测试方法,在接收到终端设备发送的测试指令时,可以直接从本地数据库中获取参数数据,该参数数据中包括测试数据和登录态数据,并将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果,最后向终端设备发送测试结果。由于直接从数据库中获取登录态数据,以将该登录态数据与测试数据组合生成参数数据传入被测业务接口进行压测,避免了在每次对被测接口进行测试时都需要

先调用登录接口才能获取到登录态数据的问题,从而在执行压测时将业务接口与登录接口分离,使得压测全部焦点放在业务接口,减少了测试任务的工作量,提高了业务接口压测效率。

[0103] 图4为本申请实施例提供的一种接口测试装置40的结构示意图,示例的,请参见图4所示,该接口测试装置40可以包括:

[0104] 获取模块401,用于在接收到终端设备发送的测试指令时,从本地数据库中获取参数数据,参数数据包括测试数据和登录态数据,登录态数据为请求登录接口后生成的数据;

[0105] 处理模块402,用于将参数数据传递到被测接口,以根据参数数据对被测接口进行测试,得到测试结果;

[0106] 发送模块403,用于向终端设备发送测试结果。

[0107] 在一种可能实现的方式中,处理模块402,具体用于:从本地数据库中调用与被测接口相关的业务数据;

[0108] 根据参数数据和业务数据,对被测接口进行测试,得到测试结果。

[0109] 在一种可能实现的方式中,处理模块402,具体用于:确定被测接口的执行逻辑;

[0110] 按照执行逻辑,将与被测接口相关的业务数据存储到本地数据库中。

[0111] 在一种可能实现的方式中,获取模块401,还用于获取测试数据;

[0112] 处理模块402,还用于根据测试数据调用登录接口,以生成登录态数据;

[0113] 处理模块402,还用于根据测试数据和登录态数据,生成参数数据,并将参数数据存储到本地数据库中。

[0114] 在一种可能实现的方式中,获取模块401,具体用于:接收终端设备发送的用于生成测试数据的请求消息,请求消息中包括第一数量和类型信息;

[0115] 根据类型信息,随机生成第一数量个测试数据。

[0116] 在一种可能实现的方式中,处理模块402,还用于在输出测试结果后的预设时间段后,删除本地数据库中的参数数据。

[0117] 在一种可能实现的方式中,处理模块402,还用于按照预先配置的并发线程数量,根据参数数据对被测接口进行测试,得到测试结果。本申请实施例提供的接口测试装置40,可以执行上述任一实施例中的接口测试方法的技术方案,其实现原理以及有益效果与接口测试方法的实现原理及有益效果类似,可参见接口测试方法的实现原理及有益效果,此处不再进行赘述。

[0118] 图5为本申请实施例提供的一种终端设备50的结构示意图,示例的,请参见图5所示,该终端设备可以包括处理器501和存储器502;其中,

[0119] 存储器502,用于存储计算机程序。

[0120] 处理器501,用于读取存储器502存储的计算机程序,并根据存储器502中的计算机程序执行上述任一实施例中的接口测试方法的技术方案。

[0121] 可选地,存储器502既可以是独立的,也可以跟处理器501集成在一起。当存储器502是独立于处理器501之外的器件时,终端设备还可以包括:总线,用于连接存储器502和处理器501。

[0122] 可选地,本实施例还包括:通信接口,该通信接口可以通过总线与处理器501连接。处理器501可以控制通信接口来实现上述终端设备的获取和发送的功能。

[0123] 本申请实施例所示的终端设备,可以执行上述任一实施例中的接口测试方法的技术方案,其实现原理以及有益效果与接口测试方法的实现原理及有益效果类似,可参见接口测试方法的实现原理及有益效果,此处不再进行赘述。

[0124] 本申请实施例还提供一种计算机可读存储介质,该计算机可读存储介质中存储有计算机执行指令,当处理器执行计算机执行指令时,实现上述任一实施例中的接口测试方法的技术方案,其实现原理以及有益效果与接口测试方法的实现原理及有益效果类似,可参见接口实现原理及有益效果,此处不再进行赘述。

[0125] 本申请实施例还提供了一种计算机程序产品,包括计算机程序,该计算机程序被处理器执行时,实现上述任一实施例中的接口测试方法的技术方案,其实现原理以及有益效果与接口测试方法的实现原理及有益效果类似,可参见接口测试方法的实现原理及有益效果,此处不再进行赘述。

[0126] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,可以通过其它的方式实现。例如,以上所描述的装置实施例仅仅是示意性的,例如,单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所展示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0127] 作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元展示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。另外,在本申请各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用硬件加软件功能单元的形式实现。

[0128] 上述以软件功能模块的形式实现的集成的模块,可以存储在一个计算机可读存储介质中。上述软件功能模块存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)或处理器(英文:processor)执行本申请各个实施例方法的部分步骤。

[0129] 应理解的是,上述处理器可以是中央处理单元(英文:Central Processing Unit,简称:CPU),还可以是其他通用处理器、数字信号处理器(英文:Digital Signal Processor,简称:DSP)、专用集成电路(英文:Application Specific Integrated Circuit,简称:ASIC)等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合发明所公开的方法的步骤可以直接体现为硬件处理器执行完成,或者用处理器中的硬件及软件模块组合执行完成。

[0130] 存储器可能包含高速RAM存储器,也可能还包括非易失性存储NVM,例如至少一个磁盘存储器,还可以为U盘、移动硬盘、只读存储器、磁盘或光盘等。

[0131] 总线可以是工业标准体系结构(Industry Standard Architecture,ISA)总线、外部设备互连(Peripheral Component,PCI)总线或扩展工业标准体系结构(Extended Industry Standard Architecture,EISA)总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示,本申请附图中的总线并不限定仅有一根总线或一种类型的总线。

[0132] 上述计算机可读存储介质可以是由任何类型的易失性或非易失性存储设备或者它们的组合实现,如静态随机存取存储器 (SRAM),电可擦除可编程只读存储器 (EEPROM),可擦除可编程只读存储器 (EPROM),可编程只读存储器 (PROM),只读存储器 (ROM),磁存储器,快闪存储器,磁盘或光盘。存储介质可以是通用或专用计算机能够存取的任何可用介质。

[0133] 最后应说明的是:以上各实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述各实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分或者全部技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

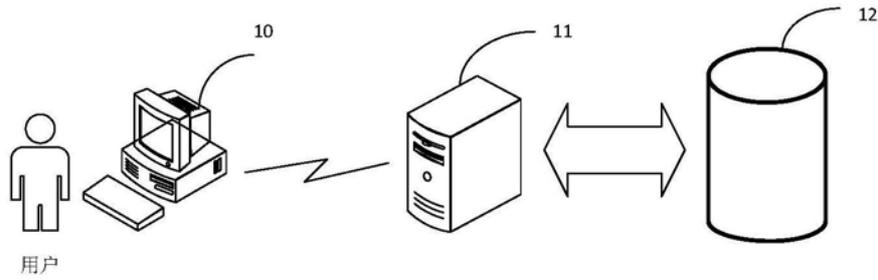


图1

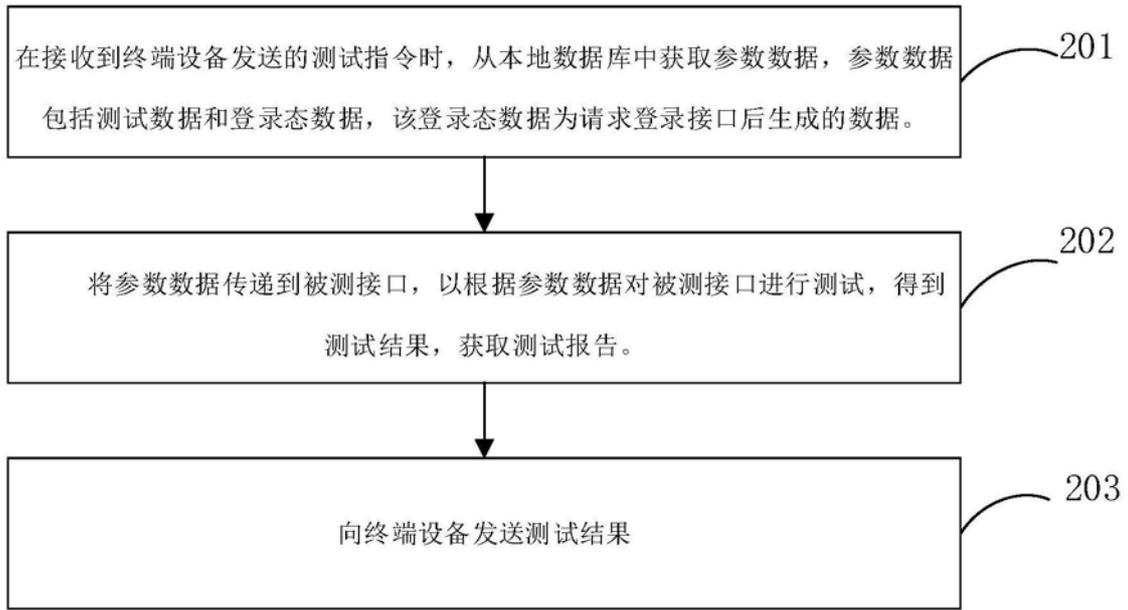


图2

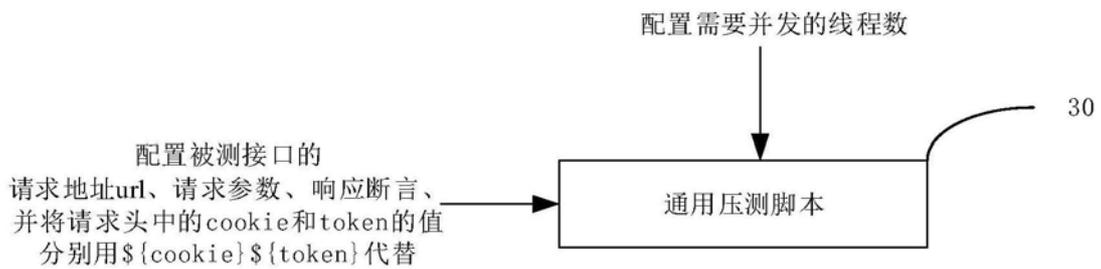


图3

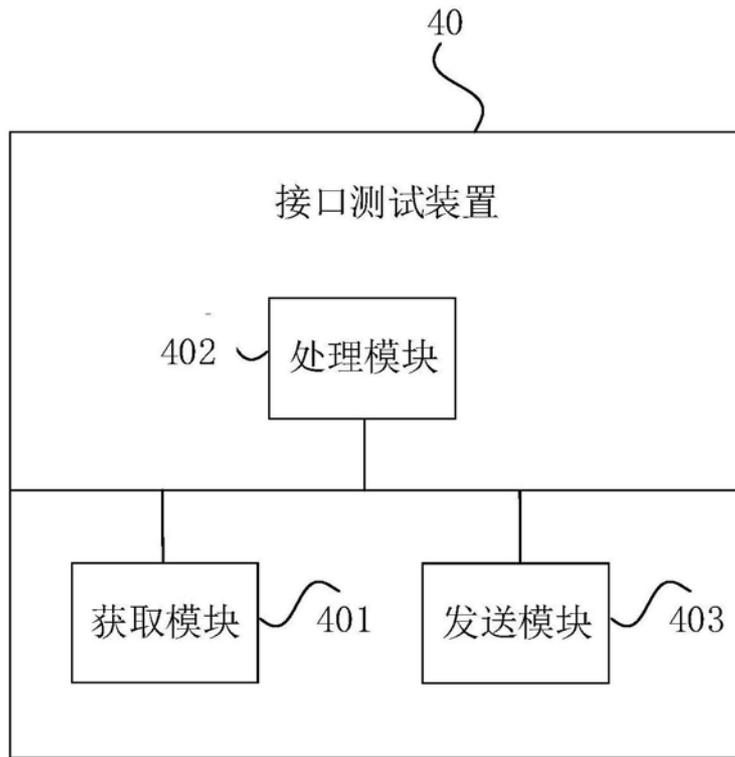


图4

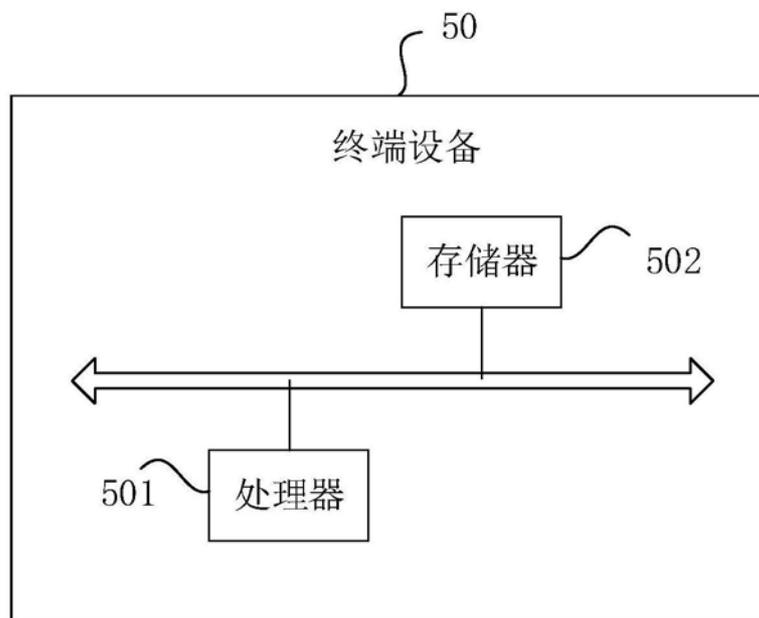


图5