



(12)发明专利申请

(10)申请公布号 CN 110633211 A

(43)申请公布日 2019.12.31

(21)申请号 201910813735.9

(22)申请日 2019.08.30

(71)申请人 深圳壹账通智能科技有限公司
地址 518000 广东省深圳市前海深港合作区前湾一路1号A栋201室

(72)发明人 陈雷

(74)专利代理机构 深圳中一联合知识产权代理有限公司 44414

代理人 李娟

(51) Int. Cl.
G06F 11/36(2006.01)

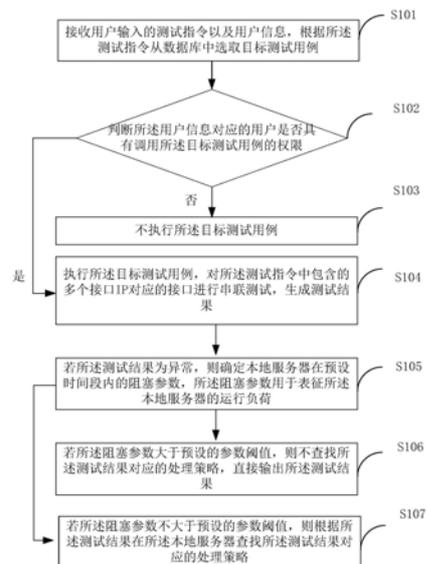
权利要求书3页 说明书9页 附图4页

(54)发明名称

对于多接口的测试方法、装置、服务器及介质

(57)摘要

本发明适用于数据处理技术领域,提供了一种对于多接口的测试方法、装置、服务器及介质,通过本地服务器接收用户输入的测试指令以及用户信息后,自动筛选出预先编译好的用于测试多个串联接口的目标测试用例,并在判定用户信息对应的用户具有调用目标测试用例的权限后,执行目标测试用例,以一次性对测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果,若测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,在本地服务器未超负荷运行的前提下,才会通过本地服务器根据上述测试结果自动地查找处理策略,从而在不额外增加本地服务器的运行负荷的情况下,尽量提高确定异常测试结果的处理策略的效率。



1. 一种对于多接口的测试方法,其特征在于,包括:

接收用户输入的测试指令以及用户信息,根据所述测试指令从数据库中选取目标测试用例;

判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限;

若所述用户信息对应的用户具有调用所述目标测试用例的权限,则执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果;

若所述测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,所述阻塞参数用于表征所述本地服务器的运行负荷;

若所述阻塞参数不大于预设的参数阈值,则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

2. 如权利要求1所述的对于多接口的测试方法,其特征在于,所述判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限,包括:

调取预设的分类树模型,并根据所述分类树模型确定所述用户信息的类别,根据预设的类别与等级参数的对应关系,确定所述用户信息的类别对应的等级参数,作为用户等级参数;

调取所述目标测试用例的初始等级参数以及生成时间;

根据公式: $\text{Textclass} = \frac{\text{InClass}}{2 - e^{-\gamma T}}$ 计算所述目标测试用例的当前等级参数,所述Textclass表示所述目标测试用例的当前等级参数,所述InClass表示所述目标测试用例的初始等级参数,所述T表示所述目标测试用例的生成时间到当前时刻的时间差值,所述Y为预设常数,所述e为自然常数;

若所述用户等级参数不小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户具有调用所述目标测试用例的权限;

若所述用户等级参数小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户不具有调用所述目标测试用例的权限。

3. 如权利要求1所述的对于多接口的测试方法,其特征在于,所述执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,包括:

按照所述目标测试用例中包含的各接口对应的测试代码的排列顺序,依次对各个接口进行测试,其中,对每条接口进行测试的步骤包括:

当开始执行一条接口对应的测试代码时,将当前正在执行的测试代码作为目标代码;验证预设哈希表中最新存入的变量名是否与所述目标代码对应的预设人参名一致,若一致,则将所述哈希表中最新存入的变量值带入所述目标代码,并继续执行所述目标代码;

在结束执行所述目标代码后,将所述目标代码测试的接口返回的变量名和变量值存入所述预设哈希表中;并在执行对下条接口的测试代码前,对所述预设哈希表中最新存入的变量值进行验证,生成该接口对应的验证结果。

4. 如权利要求1所述的对于多接口的测试方法,其特征在于,所述确定本地服务器在预设时间段内的阻塞参数,包括:

检测所述本地服务器的等候队列中的任务数量;

统计所述本地服务器在预设时间段段内各线程执行一个任务的平均执行时长;

通过公式：
$$\text{BloPar} = \left(\frac{2}{1+e^{-\text{num}}} \right)^{\text{time}}$$
 计算所述本地服务器在预设时间段内的阻塞参数，其中，BloPar表示所述本地服务器的阻塞参数，time表示所述平均执行时长，num表示所述等候队列中的任务数量，e表示自然常数。

5. 如权利要求1所述的对于多接口的测试方法，其特征在于，所述根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略，包括：

通过预设的关键词提取规则从所述测试结果中提取关键词，作为目标词；

从所述本地服务器中存储的策略数据库提取多个网页的文本数据，并对各个所述网页的文本数据进行分词处理，生成各个所述网页对应的词语集合；

通过公式：
$$\text{pro}_i = \frac{\text{NumPr}_i}{\text{TotalNum}_i} \times \ln \frac{\text{BaseTotal}}{\text{WebNum}}$$
 计算所述目标词在各个所述网页中的概率参数，其中，所述 pro_i 为所述目标词在网页i中的概率参数，所述 NumPr_i 为所述目标词在网页i对应的所述词语集合中的出现次数， TotalNum_i 为网页i对应的词语集合中含有的词语总和，所述BaseTotal为所述策略数据库中包含的网页的数量总和，所述WebNum为所述策略数据库中包含所述目标词的网页的数量；

根据预设的概率参数区间与相关度的对应关系，确定所述目标词与各个网页的相关度；

抓取与所述目标词的相关度大于或等于预设相关度阈值的网页中的文本数据，作为所述测试结果对应的处理策略。

6. 一种对于多接口的测试装置，其特征在于，所述装置包括：

接收模块，用于接收用户输入的测试指令以及用户信息，根据所述测试指令从数据库中选取目标测试用例；

判断模块，用于判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限；

测试模块，用于若所述用户信息对应的用户具有调用所述目标测试用例的权限，则执行所述目标测试用例，对所述测试指令中包含的多个接口IP对应的接口进行串联测试，判断测试结果是否异常；

计算模块，用于若所述测试结果异常，则确定本地服务器在预设时间段内的阻塞参数，所述阻塞参数用于表征所述本地服务器的运行负荷；

搜索模块，用于若所述阻塞参数不大于预设的参数阈值，则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

7. 如权利要求6所述的对于多接口的测试装置，其特征在于，所述判断模块，具体用于：

调取预设的分类树模型，并根据所述分类树模型确定所述用户信息的类别，根据预设的类别与等级参数的对应关系，确定所述用户信息的类别对应的等级参数，作为用户等级参数；

调取所述目标测试用例的初始等级参数以及生成时间；

根据公式：
$$\text{Textclass} = \frac{\text{InClass}}{1+e^{-\gamma T}}$$
 计算所述目标测试用例的当前等级参数，所述Textclass表示所述目标测试用例的当前等级参数，所述InClass表示所述目标测试用例的

当前等级参数,所述T表示所述目标测试用例的生成时间到当前时刻的时间差值,所述Y为预设常数,所述e为自然常数;

若所述用户等级参数不小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户具有调用所述目标测试用例的权限;

若所述用户等级参数小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户不具有调用所述目标测试用例的权限。

8. 如权利要求6所述的对于多接口的测试装置,其特征在于,所述计算模块,具体用于:检测所述本地服务器的等候队列中的任务数量;

统计所述本地服务器在预设时间段内各线程执行一个任务的平均执行时长;

通过公式:
$$\text{BloPar} = \left(\frac{2}{1+e^{-\text{num}}} \right)^{\text{time}}$$
 计算所述本地服务器在预设时间段内的阻塞参数,其中,BloPar表示所述本地服务器的阻塞参数,time表示所述平均执行时长,num表示所述等候队列中的任务数量,e表示自然常数。

9. 一种服务器,包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现如权利要求1至5任一项所述方法的步骤。

10. 一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现如权利要求1至5任一项所述方法的步骤。

对于多接口的测试方法、装置、服务器及介质

技术领域

[0001] 本发明属于数据处理技术领域,尤其涉及一种对于多接口的测试方法、装置、服务器及介质。

背景技术

[0002] 当前,在一个接口开发完成之后,需要通过相应的测试用例对其进行测试,以验证该接口是否开发成功。当有多个在业务上具有参数传递关系的接口开发完成之后,现有技术往往需要工程人员选取多个测试用例对他们分别进行测试,并对测试结果进行逐一分析,在测试结果异常时人工确定对应的处理策略。

[0003] 然而如上文所述,在当前的多接口测试方法中,由于需要人工选取多个测试用例对多个接口进行测试,并指定相应的处理策略,且未考虑服务器的运行负荷对于搜索处理策略的影响,所以存在明显的测试以及异常处理的效率低下的问题。

发明内容

[0004] 有鉴于此,本发明实施例提供了一种对于多接口的测试方法、装置、服务器及介质,以解决现有技术存在的测试效率低下的问题。

[0005] 本发明实施例的第一方面提供了一种对于多接口的测试方法,包括:接收用户输入的测试指令以及用户信息,根据所述测试指令从数据库中选取目标测试用例;判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限;若所述用户信息对应的用户具有调用所述目标测试用例的权限,则执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果;若所述测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,所述阻塞参数用于表征所述本地服务器的运行负荷;若所述阻塞参数不大于预设的参数阈值,则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

[0006] 本发明实施例的第二方面提供了一种对于多接口的测试装置,包括:

[0007] 接收模块,用于接收用户输入的测试指令以及用户信息,根据所述测试指令从数据库中选取目标测试用例;判断模块,用于判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限;测试模块,用于若所述用户信息对应的用户具有调用所述目标测试用例的权限,则执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,判断测试结果是否异常;计算模块,用于若所述测试结果异常,则确定本地服务器在预设时间段内的阻塞参数,所述阻塞参数用于表征所述本地服务器的运行负荷;搜索模块,用于若所述阻塞参数不大于预设的参数阈值,则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

[0008] 本发明实施例的第三方面提供了一种服务器,包括:存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现本发明实施例的第一方面提供的方法的步骤。

[0009] 本发明实施例的第四方面提供了一种计算机可读存储介质,所述计算机可读存储介质存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现本发明实施例的第一方面提供的方法的步骤。

[0010] 在本发明实施例中,通过本地服务器接收用户输入的测试指令以及用户信息后,自动筛选出预先编译好的用于测试多个串联接口的目标测试用例,并在判定用户信息对应的用户具有调用所述目标测试用例的权限后,执行目标测试用例,以一次性对测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果,从而提高测试效率,若所述测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,以确定本地服务器是否超负荷运行,在本地服务器未超负荷运行的前提下,才会通过本地服务器根据上述测试结果自动地查找处理策略,从而在不额外增加本地服务器的运行负荷的情况下,尽量提高确定异常测试结果的处理策略的效率。

附图说明

[0011] 为了更清楚地说明本发明实施例中的技术方案,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性的前提下,还可以根据这些附图获得其他的附图。

[0012] 图1是本发明实施例提供的对于多接口的测试方法的实现流程图;

[0013] 图2是本发明实施例提供的对于多接口的测试方法S102的具体实现流程图;

[0014] 图3是本发明实施例提供的对于多接口的测试方法S107的具体实现流程图;

[0015] 图4是本发明实施例提供的对于多接口的测试装置的结构框图;

[0016] 图5是本发明实施例提供的服务器的示意图。

具体实施方式

[0017] 以下描述中,为了说明而不是为了限定,提出了诸如特定系统结构、技术之类的具体细节,以便透彻理解本发明实施例。然而,本领域的技术人员应当清楚,在没有这些具体细节的其它实施例中也可以实现本发明。在其它情况中,省略对众所周知的系统、装置、电路以及方法的详细说明,以免不必要的细节妨碍本发明的描述。

[0018] 为了说明本发明所述的技术方案,下面通过具体实施例来进行说明。

[0019] 图1示出了本发明实施例提供的对于多接口的测试方法的实现流程,该方法流程包括步骤S101至S107。各步骤的具体实现原理如下。

[0020] 在S101中,接收用户输入的测试指令以及用户信息,根据所述测试指令从数据库中选取目标测试用例。

[0021] 在本发明实施例中,预先设定了测试指令和测试用例的对应关系,因此在本地服务器接收到一个测试指令后,会自动筛选出与其对应的测试用例作为目标测试用例。值得注意地,本发明实施例中的测试用例包括针对一个接口进行测试的测试用例,还包括用于对多个有参数传递关系的串联接口进行测试的测试用例。对于用于对多个串联接口进行测试的测试用例而言,执行一个测试用例后生成的测试结果包含了一个以上的接口的测试数据,一旦其中一个接口的测试数据为异常,则整体的测试结果即为异常。

[0022] 可以理解地,用户信息用于描述向本地服务器输入测试指令的用户(即工程人员)的身份数据,其中,用户信息包括:用户的职级、开发团队、开发时长以及所处地区等。在本发明实施例中,用户信息可用于在其他步骤中确定用户可调取测试用例的权限,从而提高测试的安全性,保证接口数据的安全。

[0023] 在S102中,判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限。

[0024] 在实际测试过程中,用户的错误测试指令可能会影响正常接口的运行,导致本来开发正常的接口出现异常。此外,有些接口的测试情况并不能够公开,只有相应的开发者团队才可以了解这些接口的测试情况。因此,基于包括上述原因等情况,本发明实施例需要判断用户信息对应的用户是否具有调用目标测试用例的权限。

[0025] 作为本发明的一个实施例,如图2所示,上述S102包括:

[0026] S1021,调取预设的分类树模型,并根据所述分类树模型确定所述用户信息的类别,根据预设的类别与等级参数的对应关系,确定所述用户信息的类别对应的等级参数,作为用户等级参数。

[0027] 在本发明实施例中,本地服务器中预先存储一个分类树模型,分类树模型中除叶子节点外的每一个节点均对应一个分类规则,每一个分类规则针对用户信息中的一类数据进行分类。因此,通过该分类树模型可以将用户信息一层层的进行分类,最终将用户信息分配至一个叶子节点中。随后,再根据预设的叶子节点与等级参数的对应关系,就可以确定出用户信息对应的等级参数。

[0028] 示例性地,假设用户信息包括:“职级:10,开发团队:A,临时任务:01”,假设分类树模型的根节点通过“职级”进行分类,第二级节点通过“开发团队”进行分类,第三级节点通过“临时任务”进行分类,则通过三层的分类,可以将用户信息分配到一个叶子节点,再根据预设的叶子节点与等级参数的对应关系,就可以确定出用户信息对应的等级参数。

[0029] S1022,调取所述目标测试用例的初始等级参数以及生成时间。

[0030] 在本发明实施例中,目标测试用例的初始等级参数用于为在目标测试用例生成时,编译人员对其设定的等级参数;生成时间表示目标测试用例在编译后存入本地服务器的时间。

[0031] S1023,根据等级参数公式计算所述目标测试用例的当前等级参数。

[0032] 可选地,所述等级参数公式包括:
$$\text{Textclass} = \frac{\text{InClass}}{2 - e^{-YT}}$$
所述Textclass表示所述目标测试用例的当前等级参数,所述InClass表示所述目标测试用例的初始等级参数,所述T表示所述目标测试用例的生成时间到当前时刻的时间差值,所述Y为预设常数,所述e为自然常数。

[0033] 可以理解地,在现实的测试过程中,由于一个测试用例可能开始的时候敏感性和保密性要求很高,但随着时间的推移,该测试用例渐渐过时,其保密性要求自然降低。所以,在本发明实施例中,一个目标测试用例的当前等级参数与其初始等级成正比,与其时间差值成反比。

[0034] S1024,若所述用户等级参数不小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户具有调用所述目标测试用例的权限。

[0035] S1025,若所述用户等级参数小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户不具有调用所述目标测试用例的权限。

[0036] 可以理解地,本发明实施例通过对用户等级参数与目标测试用例的当前等级参数进行比较,判定用户信息对应的用户是否具有调用所述目标测试用例的权限,从而保证测试用例的数据安全。

[0037] 在S103中,若所述用户信息对应的用户不具有调用所述目标测试用例的权限,则不执行所述目标测试用例。

[0038] 在S104中,若所述用户信息对应的用户具有调用所述目标测试用例的权限,则执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果。

[0039] 在本发明实施例中,按照所述目标测试用例中包含的各接口对应的测试代码的排列顺序,依次对各个接口进行测试,其中,对每条接口进行测试的步骤包括:

[0040] 步骤1:当开始执行一条接口对应的测试代码时,将当前正在执行的测试代码作为目标代码;验证预设哈希表中最新存入的变量名是否与所述目标代码对应的预设入参名一致,若一致,则将所述哈希表中最新存入的变量值带入所述目标代码,并继续执行所述目标代码。

[0041] 具体地,通过for循环语句逐一执行对各个接口进行测试的测试代码,将本轮正在执行的测试代码作为目标代码。可以理解地,当前正在被测试的接口有其需要输入的入参,在本发明实施例中,由于各个被测试的接口之间存在参数传递关系,所以当前正在被测试的接口的入参应该与上一轮测试的接口的出参相对应,所以目标代码对应的预设入参名应该与上一轮测试的接口的出参名相一致。又由于在本发明实施例中,当一个接口的测试代码全部运行完毕后,会收到该接口返回的变量名和变量值,并将其存储于预设哈希表中,所以当刚开始执行一条接口对应的测试代码时,哈希表中最新存入的变量值和变量名为上一轮测试的接口的出参值和出参名。

[0042] 基于上述原因,通过验证预设哈希表中最新存入的变量名是否与所述目标代码对应的预设入参名一致,可以确定出目标代码对应的预设入参名是否与上一轮测试的接口的出参名相一致,若一致,则证明合规,可以将哈希表中最新存入的变量值(即上一轮测试的接口返回的出参值)带入所述目标代码,从而生成完整的可用于对当前正在被测试的接口进行完整测试的代码,以继续执行所述目标代码。

[0043] 步骤2:在结束执行所述目标代码后,将所述目标代码测试的接口返回的变量名和变量值存入所述预设哈希表中;并在执行对下条接口的测试代码前,对所述预设哈希表中最新存入的变量值进行验证,生成该接口对应的验证结果。

[0044] 可以理解地,如上文所述,在完全执行完本轮的一条测试代码后,该测试代码所测试的接口会返回变量名和变量值,在本地服务器将这两个参数进行存储后,哈希表中最新存入的变量名和变量值自然就发生了更新。在本发明实施例中,更新后的最新存入的变量名和变量值一方面可以用于对下个接口进行测试,一方面可以用于生成本轮所测试的接口的验证结果。

[0045] 可选地,对所述预设哈希表中最新存入的变量值进行验证,生成该接口对应的验证结果,包括:从所述目标测试用例中取出本轮所测试的接口的变量值区间与验证结果的

对应关系,判断所述最新存入的变量值所属的所述变量值区间,并基于所述变量值区间与验证结果的对应关系,确定出本轮所测试的接口的验证结果。

[0046] 显然,本发明实施例通过一个目标测试用例实现对多个接口的串联测试,简化了测试人员的工作量,提高了测试效率。

[0047] 在S105中,若所述测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,所述阻塞参数用于表征所述本地服务器的运行负荷。

[0048] 本发明实施例考虑到在验证结果为异常时,测试工程师或其他用户可能需要查找相应的处理策略,以修复接口功能,所以设计了一个自动化查找处理策略的方法。但是由于测试工作的主要目的就是为生成验证结果,所以如果在本地服务器的运行负荷过大时,证明当前可能有多个测试用例被同时执行,这时如果查找相应的处理策略,就可能进一步增加本地服务器的运行负荷,从而减慢处理对接口进行测试,生成测试结果的效率。基于此原因,本发明实施例在自动查找处理策略之前,会首先确定本地服务器的运行负荷,只有在运行负荷没有达到预设的门限值时,才会进一步查找对异常验证结果的处理策略。

[0049] 在本发明实施例,通过确定本地服务器在预设时间段内的阻塞参数,以确定本地服务器的运行负荷,具体步骤如下:

[0050] 检测所述本地服务器的等候队列中的任务数量;统计所述本地服务器在预设时间段内各线程执行一个任务的平均执行时长。

[0051] 通过公式:
$$\text{BloPar} = \left(\frac{2}{1+e^{-\text{num}}} \right)^{\text{time}}$$
计算所述本地服务器在预设时间段内的

阻塞参数,其中,BloPar表示所述本地服务器的阻塞参数,time表示所述平均执行时长,num表示所述等候队列中的任务数量,e表示自然常数。

[0052] 显然,在本发明实施例中,阻塞参数与任务数量成正比,也与平均执行时长成正比,但是由于上述公式的各参数的形式,导致阻塞参数不会随着任务数量和平均执行时长出现较大幅度的波动,可以在一定程度上维持在合理区间。

[0053] 在S106中,若所述阻塞参数大于预设的参数阈值,则不查找所述测试结果对应的处理策略,直接输出所述测试结果。

[0054] 在S107中,若所述阻塞参数不大于预设的参数阈值,则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

[0055] 在本发明实施例中,需要根据测试结果的关键词在本地服务器中存储的策略数据库中提取相应的处理策略,具体步骤如图3所示包括S1071-S1075,详述如下:

[0056] S1071,通过预设的关键词提取规则从所述测试结果中提取关键词,作为目标词。

[0057] 可选地,预设的关键词提取规则中包含了多个接口和关键词集合的对应关系,所以一旦某个接口对应的测试结果中出现了该接口对应的关键词集合中的词语,则把该词语提取出来作为目标词。

[0058] S1072,从所述本地服务器中存储的策略数据库提取多个网页的文本数据,并对各个所述网页的文本数据进行分词处理,生成各个所述网页对应的词语集合。

[0059] 在本发明实施例中,事先收集了多个包含处理策略的网页,并将这些网页存储于本地服务器的策略数据库中,以供本地服务器在需要时调取相关的网页从而获取处理策略。可以理解地,一个网页可能针对着多种异常的测试结果的处理策略,多个网页也可能针

对一种异常的测试结果的处理策略。

[0060] 可选地,通过jieba分词器对各个网页的文本数据进行分词处理。

[0061] S1073,通过预设的概率公式计算所述目标词在各个所述网页中的概率参数。

[0062] 可选地,所述概率公式包括: $pro_i = \frac{NumPr_i}{TotalNum_i} \times \ln \frac{BaseTotal}{WebNum}$,其中,所述 pro_i 为

所述目标词在网页i中的概率参数,所述 $NumPr_i$ 为所述目标词在网页i对应的所述词语集合中的出现次数, $TotalNum_i$ 为网页i对应的词语集合中含有的词语总和,所述BaseTotal为所述策略数据库中包含的网页的数量总和,所述WebNum为所述策略数据库中包含所述目标词的网页的数量。

[0063] 本发明实施例主要通过目标词在网页中的出现次数来确定概率参数,但是为了避免由于不同网页本身的篇幅不同对最终概率参数的影响,采用了上述的概率公式,可以更加客观地计算出用于体现网页和目标词关联程度的概率参数。

[0064] S1074,根据预设的概率参数区间与相关度的对应关系,确定所述目标词与各个网页的相关度。

[0065] S1075,抓取与所述目标词的相关度大于或等于预设相关度阈值的网页中的文本数据,作为所述测试结果对应的处理策略。

[0066] 在本发明实施例中,通过本地服务器接收用户输入的测试指令以及用户信息后,自动筛选出预先编译好的用于测试多个串联接口的目标测试用例,并在判定用户信息对应的用户具有调用所述目标测试用例的权限后,执行目标测试用例,以一次性对测试指令中包含的多个接口IP对应的接口进行串联测试,生成测试结果,从而提高测试效率,若所述测试结果为异常,则确定本地服务器在预设时间段内的阻塞参数,以确定本地服务器是否超负荷运行,在本地服务器未超负荷运行的前提下,才会通过本地服务器根据上述测试结果自动地查找处理策略,从而在不额外增加本地服务器的运行负荷的情况下,尽量提高确定异常测试结果的处理策略的效率。

[0067] 对应于上文实施例所述的对于多接口的测试方法,图4示出了本发明实施例提供的对于多接口的测试装置的结构框图,为了便于说明,仅示出了与本发明实施例相关的部分。

[0068] 参照图4,该装置包括:

[0069] 接收模块401,用于接收用户输入的测试指令以及用户信息,根据所述测试指令从数据库中选取目标测试用例;

[0070] 判断模块402,用于判断所述用户信息对应的用户是否具有调用所述目标测试用例的权限;

[0071] 测试模块403,用于若所述用户信息对应的用户具有调用所述目标测试用例的权限,则执行所述目标测试用例,对所述测试指令中包含的多个接口IP对应的接口进行串联测试,判断测试结果是否异常;

[0072] 计算模块404,用于若所述测试结果异常,则确定本地服务器在预设时间段内的阻塞参数,所述阻塞参数用于表征所述本地服务器的运行负荷;

[0073] 搜索模块405,用于若所述阻塞参数不大于预设的参数阈值,则根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略。

[0074] 可选地,所述判断模块,具体用于:

[0075] 调取预设的分类树模型,并根据所述分类树模型确定所述用户信息的类别,根据预设的类别与等级参数的对应关系,确定所述用户信息的类别对应的等级参数,作为用户等级参数;

[0076] 调取所述目标测试用例的初始等级参数以及生成时间;

[0077] 根据公式: $\text{Textclass} = \frac{\text{InClass}}{1+e^{-YT}}$ 计算所述目标测试用例的当前等级参数,所述

Textclass表示所述目标测试用例的当前等级参数,所述InClass表示所述目标测试用例的当前等级参数,所述T表示所述目标测试用例的生成时间到当前时刻的时间差值,所述Y为预设常数,所述e为自然常数;

[0078] 若所述用户等级参数不小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户具有调用所述目标测试用例的权限;

[0079] 若所述用户等级参数小于所述目标测试用例的当前等级参数,则判定所述用户信息对应的用户不具有调用所述目标测试用例的权限。

[0080] 可选地,所述测试模块,具体用于:按照所述目标测试用例中包含的各接口对应的测试代码的排列顺序,依次对各个接口进行测试,其中,对每条接口进行测试的步骤包括:

[0081] 当开始执行一条接口对应的测试代码时,将当前正在执行的测试代码作为目标代码;验证预设哈希表中最新存入的变量名是否与所述目标代码对应的预设人参名一致,若一致,则将所述哈希表中最新存入的变量值带入所述目标代码,并继续执行所述目标代码;

[0082] 在结束执行所述目标代码后,将所述目标代码测试的接口返回的变量名和变量值存入所述预设哈希表中;并在执行对下条接口的测试代码前,对所述预设哈希表中最新存入的变量值进行验证,生成该接口对应的验证结果。

[0083] 可选地,所述确定本地服务器在预设时间段内的阻塞参数,包括:

[0084] 检测所述本地服务器的等候队列中的任务数量;

[0085] 统计所述本地服务器在预设时间段段内各线程执行一个任务的平均执行时长;

[0086] 通过公式: $\text{BloPar} = \left(\frac{2}{1+e^{-\text{num}}} \right)^{\text{time}}$ 计算所述本地服务器在预设时间段内的

阻塞参数,其中,BloPar表示所述本地服务器的阻塞参数,time表示所述平均执行时长,num表示所述等候队列中的任务数量,e表示自然常数。

[0087] 可选地,所述根据所述测试结果在所述本地服务器查找所述测试结果对应的处理策略,包括:

[0088] 通过预设的关键词提取规则从所述测试结果中提取关键词,作为目标词;

[0089] 从所述本地服务器中存储的策略数据库提取多个网页的文本数据,并对各个所述网页的文本数据进行分词处理,生成各个所述网页对应的词语集合;

[0090] 通过公式: $\text{pro}_i = \frac{\text{NumPr}_i}{\text{TotalNum}_i} \times \ln \frac{\text{BaseTotal}}{\text{WebNum}}$ 计算所述目标词在各个所述网页中的

的概率参数,其中,所述 pro_i 为所述目标词在网页i中的概率参数,所述 NumPr_i 为所述目标词在网页i对应的所述词语集合中的出现次数, TotalNum_i 为网页i对应的词语集合中含有的词语总和,所述BaseTotal为所述策略数据库中包含的网页的数量总和,所述WebNum为所述

策略数据库中包含所述目标词的网页的数量；

[0091] 根据预设的概率参数区间与相关度的对应关系，确定所述目标词与各个网页的相关度；

[0092] 抓取与所述目标词的相关度大于或等于预设相关度阈值的网页中的文本数据，作为所述测试结果对应的处理策略。

[0093] 在本发明实施例中，通过本地服务器接收用户输入的测试指令以及用户信息后，自动筛选出预先编译好的用于测试多个串联接口的目标测试用例，并在判定用户信息对应的用户具有调用所述目标测试用例的权限后，执行目标测试用例，以一次性对测试指令中包含的多个接口IP对应的接口进行串联测试，生成测试结果，从而提高测试效率，若所述测试结果为异常，则确定本地服务器在预设时间段内的阻塞参数，以确定本地服务器是否超负荷运行，在本地服务器未超负荷运行的前提下，才会通过本地服务器根据上述测试结果自动地查找处理策略，从而在不额外增加本地服务器的运行负荷的情况下，尽量提高确定异常测试结果的处理策略的效率。

[0094] 图5是本发明一实施例提供的服务器的示意图。如图5所示，该实施例的服务器5包括：处理器50、存储器51以及存储在所述存储器51中并可在所述处理器50上运行的计算机程序52，例如对于多接口的测试程序。所述处理器50执行所述计算机程序52时实现上述各个对于多接口的测试方法实施例中的步骤，例如图1所示的步骤101至107。或者，所述处理器50执行所述计算机程序52时实现上述各装置实施例中各模块/单元的功能，例如图4所示单元401至405的功能。

[0095] 示例性的，所述计算机程序52可以被分割成一个或多个模块/单元，所述一个或者多个模块/单元被存储在所述存储器51中，并由所述处理器50执行，以完成本发明。所述一个或多个模块/单元可以是能够完成特定功能的一系列计算机程序指令段，该指令段用于描述所述计算机程序52在所述服务器5中的执行过程。

[0096] 所述服务器5可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。所述服务器可包括，但不仅限于，处理器50、存储器51。本领域技术人员可以理解，图5仅仅是服务器5的示例，并不构成对服务器5的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件，例如所述服务器还可以包括输入输出设备、网络接入设备、总线等。

[0097] 所称处理器50可以是中央处理单元 (Central Processing Unit, CPU)，还可以是其他通用处理器、数字信号处理器 (Digital Signal Processor, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现成可编程门阵列 (Field-Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。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。

[0098] 所述存储器51可以是所述服务器5的内部存储单元，例如服务器5的硬盘或内存。所述存储器51也可以是所述服务器5的外部存储设备，例如所述服务器5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，所述存储器51还可以既包括所述服务器5的内部存储单元也包括外部存储设备。所述存储器51用于存储所述计算机程序以及所述服务器所需的其他程序

和数据。所述存储器51还可以用于暂时地存储已经输出或者将要输出的数据。

[0099] 在上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述或记载的部分,可以参见其它实施例的相关描述。

[0100] 所述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,作为单元显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的单元来实现本实施例方案的目的。

[0101] 所述集成的模块/单元如果以软件功能单元的形式实现并作为独立的产品销售或使用,可以存储在一个计算机可读取存储介质中。基于这样的理解,本发明实现上述实施例方法中的全部的或部分流程,也可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一计算机可读存储介质中。

[0102] 以上所述实施例仅用以说明本发明的技术方案,而非对其限制;尽管参照前述实施例对本发明进行了详细的说明,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发明各实施例技术方案的精神和范围,均应包含在本发明的保护范围之内。

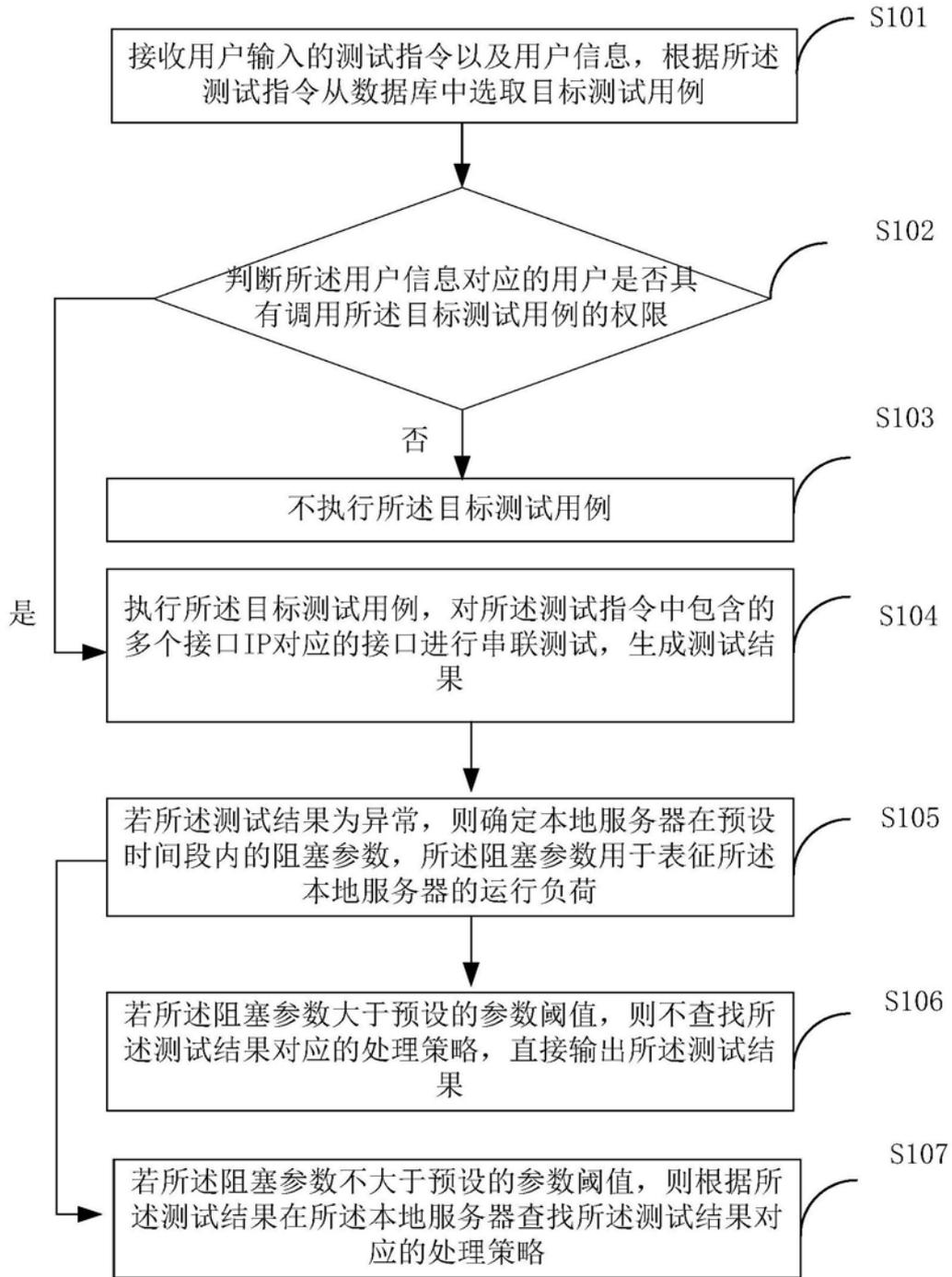


图1

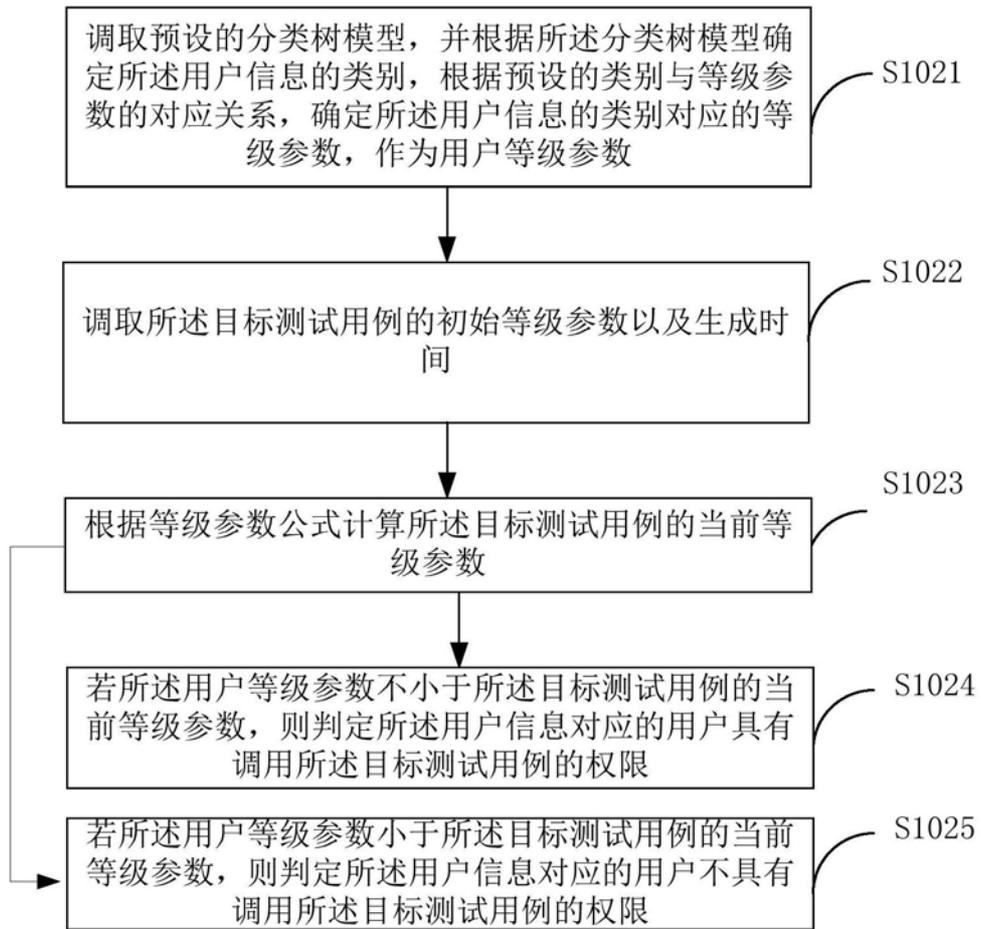


图2

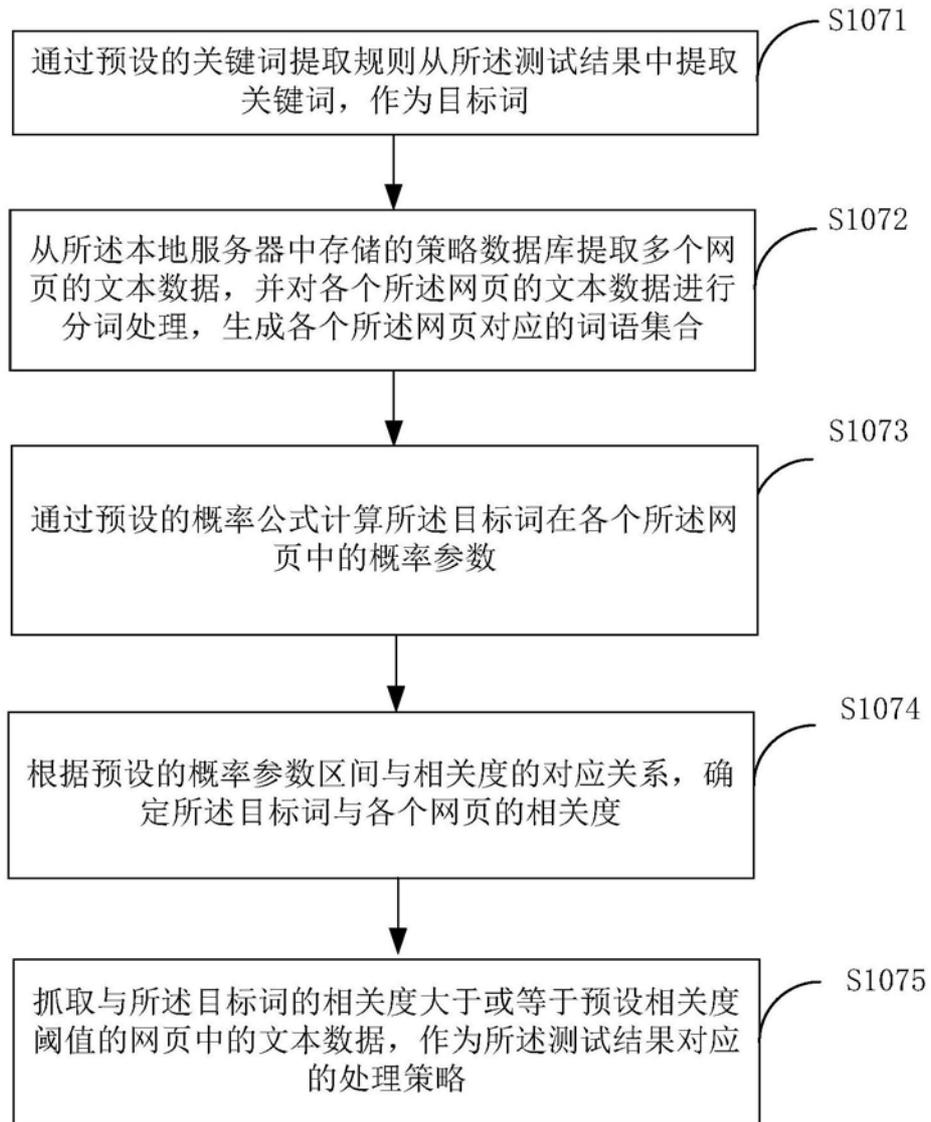


图3

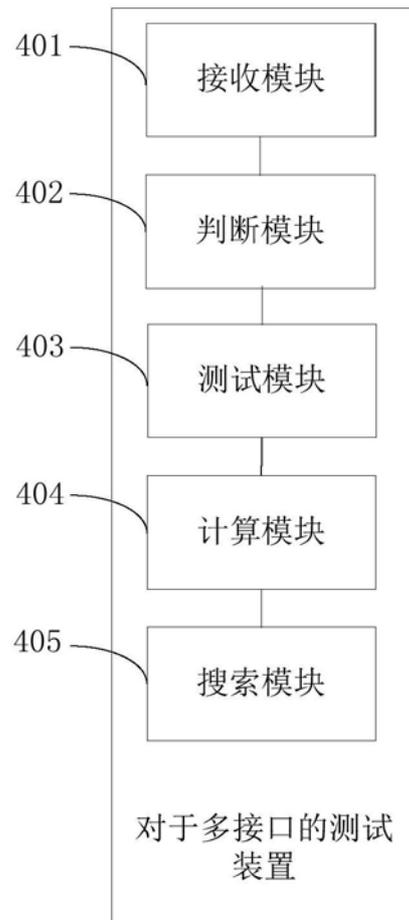


图4

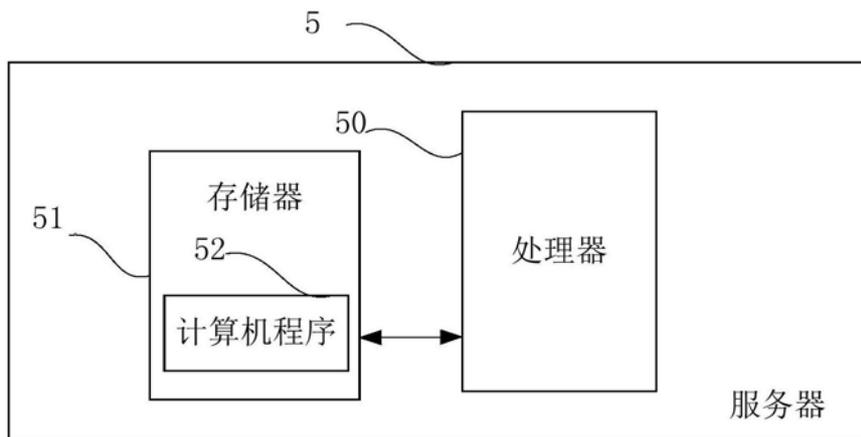


图5