



(12) 发明专利申请

(10) 申请公布号 CN 115904940 A

(43) 申请公布日 2023. 04. 04

(21) 申请号 202211317710.8

(22) 申请日 2022.10.26

(71) 申请人 南方科技大学

地址 518055 广东省深圳市南山区桃源街
道学苑大道1088号

(72) 发明人 廖淳昊 康雨晖 杨文汉夫
彭佳然 王斯南 刘烨庞

(74) 专利代理机构 深圳市君胜知识产权代理事
务所(普通合伙) 44268
专利代理师 孙果

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

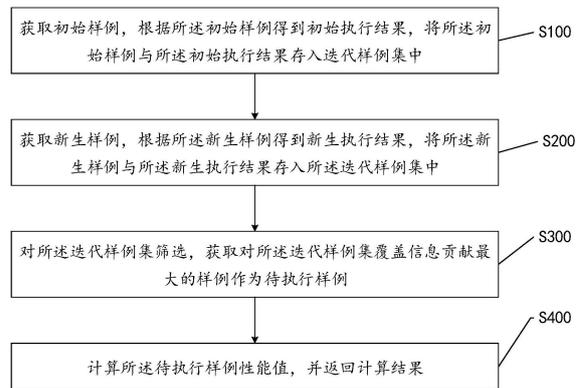
权利要求书2页 说明书7页 附图2页

(54) 发明名称

一种自适应回归随机测试方法、装置、终端及介质

(57) 摘要

本发明公开了一种自适应回归随机测试方法、装置、终端及介质,其中,方法包括:获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例;计算所述待执行样例性能值,并返回计算结果。本发明根据测试用例的代码覆盖情况,定义差异大小。本发明在保证覆盖信息的基础上减少测试用例数量可以大幅度地缩短测试周期及节约测试人力成本。



1. 一种自适应回归随机测试方法,其特征在于,所述自适应回归随机测试方法包括以下步骤:

获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;其中,所述初始执行结果为所述初始样例的覆盖信息;

获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;其中,所述新生执行结果为所述新生样例的覆盖信息;

对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例;

计算所述待执行样例性能值,并返回计算结果。

2. 根据权利要求1所述的自适应回归随机测试方法,其特征在于,所述获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中,包括:

获取所述初始样例;

记录所述初始样例的覆盖信息,存入迭代样例集中;其中,所述初始样例的覆盖信息包括所述初始样例的覆盖率和代码行的覆盖信息。

3. 根据权利要求1所述的自适应回归随机测试方法,其特征在于,所述获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中,包括:

获取所述新生样例;

记录所述新生样例的覆盖信息,存入迭代样例集中其中;其中,所述新生样例的覆盖信息包括所述新生样例的覆盖率和代码行的覆盖信息。

4. 根据权利要求1所述的自适应回归随机测试方法,其特征在于,所述对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例,包括:

根据所述迭代样例集的总体覆盖率信息,定义并且更新所述初始样例和所述新生样例的距离。

5. 根据权利要求1所述的自适应回归随机测试方法,其特征在于,所述对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例,还包括:

基于每个初始样例和每个新生样例对应的距离,使用贪心算法,迭代地选取出所述迭代样例集中的距离最大的样例,作为待执行样例,并放入迭代样例集。

6. 根据权利要求1所述的自适应回归随机测试方法,其特征在于,所述计算所述待执行样例性能值,并返回计算结果,包括:

将所述待执行样例传输至性能分析模型中进行计算,得到所述待执行样例的模型性能值,并返回;其中,所述性能分析模型采用模型性能值计算公式进行计算。

7. 根据权利要求6所述的自适应回归随机测试方法,其特征在于,所述模型性能值计算公式为: $E = \frac{a}{b} \div \frac{c}{d}$, 其中,E为模型性能值,a为筛选的样例的覆盖信息,b为所有样例的覆盖信息,c为筛选的样例数,d为样例总数。

8. 一种自适应回归随机测试装置,其特征在于,包括:

测试样例生成模块,用于获取初始样例,根据所述初始样例得到初始执行结果,将所述

初始样例与所述初始执行结果存入迭代样例集中；

还用于获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中；

测试样例筛选模块,用于对所述迭代样例集筛选,获取待执行样例；

性能分析模块,对所述迭代样例集进行性能分析。

9.一种终端,其特征在于,包括:处理器以及存储器,所述存储器存储有自适应回归随机测试程序,所述自适应回归随机测试程序被所述处理器执行时用于实现如权利要求1-7中任意一项所述的自适应回归随机测试方法。

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存储有自适应回归随机测试程序,所述自适应回归随机测试程序被处理器执行时用于实现如权利要求1-7中任意一项所述的自适应回归随机测试方法。

一种自适应回归随机测试方法、装置、终端及介质

技术领域

[0001] 本发明涉及终端应用领域,尤其涉及的是一种自适应回归随机测试方法、装置、终端及介质。

背景技术

[0002] 随着软件功能的日益强大,软件测试也越来越重要,要求也越来越高。搜索引擎后端测试需要在较短的时间内保证较高的运行稳定性,例如需要千万量级的数据下维持稳定的运行。

[0003] 在该需求的驱动下,对搜索引擎后端采用大数据量测试是保证其运行稳定性的常用方式。然而,大数据量测试就需要较长的测试时间,而在实际应用中由于产品的研发周期限制等原因,限制了测试时间。因此由于测试时间的制约,往往需要在测试周期和数据量之间做一个平衡。此外,搜索引擎后端功能和策略复杂度较高,为特定功能或策略手工甄选测试数据困难较大,同样也给测试人员带来了很大的不便。

[0004] 因此,如何在保证数据量要求的基础上,缩短测试时间,提高测试精度成为了亟待解决的问题。

发明内容

[0005] 本发明要解决的技术问题在于,针对现有技术缺陷,本发明提供一种自适应回归随机测试方法、装置、终端及介质,以解决目前测试时间长、数据量不能满足要求的缺陷的技术问题。

[0006] 本发明解决技术问题所采用的技术方案如下:

[0007] 第一方面,本发明提供了一种自适应回归随机测试方法,所述自适应回归随机测试方法包括以下步骤:

[0008] 一种自适应回归随机测试方法,其特征在于,所述自适应回归随机测试方法包括以下步骤:

[0009] 获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;其中,所述初始执行结果为所述初始样例的覆盖信息;

[0010] 获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;其中,所述新生执行结果为所述新生样例的覆盖信息;

[0011] 对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例;

[0012] 计算所述待执行样例性能值,并返回计算结果。

[0013] 在一种实现方式中,所述获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中,包括:

[0014] 获取所述初始样例;

[0015] 记录所述初始样例的覆盖信息,存入迭代样例集中;其中,所述初始样例的覆盖信

息包括所述初始样例的覆盖率和代码行的覆盖信息。

[0016] 在一种实现方式中,所述获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中,包括:

[0017] 获取所述新生样例;

[0018] 记录所述新生样例的覆盖信息,存入迭代样例集中其中;其中,所述新生样例的覆盖信息包括所述新生样例的覆盖率和代码行的覆盖信息。

[0019] 在一种实现方式中,所述对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例,包括:

[0020] 根据所述迭代样例集的总体覆盖率信息,定义并且更新所述初始样例和所述新生样例的距离。

[0021] 在一种实现方式中,所述对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例,还包括:

[0022] 基于每个初始样例和每个新生样例对应的距离,使用贪心算法,迭代地选取出所述迭代样例集中的距离最大的样例,作为待执行样例,并放入迭代样例集。

[0023] 在一种实现方式中,所述计算所述待执行样例性能值,并返回计算结果,包括:

[0024] 将所述待执行样例传输至性能分析模型中进行计算,得到所述待执行样例的模型性能值,并返回;其中,所述性能分析模型采用模型性能值计算公式进行计算。

[0025] 在一种实现方式中,所述模型性能值计算公式为: $E = \frac{a}{b} \div \frac{c}{d}$,其中,E为模型性能值,a为筛选的样例的覆盖信息,b为所有样例的覆盖信息,c为筛选的样例数,d为样例总数。

[0026] 第二方面,本发明提供了一种自适应回归随机测试装置,包括:

[0027] 测试样例生成模块,用于获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;还用于获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;

[0028] 测试样例筛选模块,用于对所述迭代样例集筛选,获取待执行样例;

[0029] 性能分析模块,对所述迭代样例集进行性能分析。

[0030] 第三方面,本发明提供了一种终端,包括:处理器以及存储器,所述存储器存储有自适应回归随机测试程序,所述自适应回归随机测试程序被所述处理器执行时用于实现第一方面所述的自适应回归随机测试方法。

[0031] 第四方面,本发明提供了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存储有自适应回归随机测试程序,所述自适应回归随机测试程序被处理器执行时用于实现第一方面所述的自适应回归随机测试方法。

[0032] 本发明采用上述技术方案具有以下效果:

[0033] 本发明通过代码覆盖信息的变化情况,判断对应的输入数据(即初始样例和新生样例)对代码覆盖信息是否构成了贡献,如果没有贡献则可将该测试样例的输入数据筛除,从而可以在保证代码覆盖信息不变的前提下可以有效地将原来千万量级的输入数据压缩至百量级,因此可以大幅度地提高测试速度,及缩短测试周期,并且无需人工干预,因此可以节约人力成本。

附图说明

[0034] 为了更清楚地说明本发明实施例或现有技术中的技术方案,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的前提下,还可以根据这些附图示出的结构获得其他的附图。

[0035] 图1是本发明的一种实现方式中自适应回归随机测试方法的流程图。

[0036] 图2是本发明的一种实现方式中终端的功能原理图。

[0037] 图3是本发明提供的装置结构框图。

[0038] 本发明目的的实现、功能特点及优点将结合实施例,参照附图做进一步说明。

具体实施方式

[0039] 为使本发明的目的、技术方案及优点更加清楚、明确,以下参照附图并举实施例对本发明进一步详细说明。应当理解,此处所描述的具体实施例仅仅用以解释本发明,并不用于限定本发明。

[0040] 发明人经过研究发现软件制品都会存在各种类型的缺陷,因此,开发者们需要应用各种软件质量保证的手段,这其中最主要的手段便是软件测试(Software Testing),软件测试是指在规定的条件下对待测程序进行操作,并评估其是否能满足原有需求的过程,一般包括单元测试、集成测试、系统测试等;随机测试(Random Testing)是指根据待测程序的输入域随机生成一系列的测试用例,通常被用于生成单元测试的测试套件(Test Suite)上,而自适应随机测试方法(Adaptive Random Testing,简称ART)是在随机测试的基础上,通过对生成的测试样例之间的差异进行评估,并以此对样例进行筛选,使得最终的测试套件具备一定分布特征(例如:均匀分布、条带分布、聚集分布)的一种测试方法,不同的ART技术之间的差异,一般体现在对测试样例间差异的定义、样例筛选的方法上。

[0041] 由于网络应用程序功能需求变动大、功能迭代迅速、并且需要修复缺陷,因此,后端的代码需要频繁修改,为了避免修改产生新的错误或导致其他原有代码产生错误,有必要引入回归测试(Regression Test),即在修改代码后,重新运行原有测试用例。开展回归测试需要对测试套件进行维护和管理。由于测试套件中的测试样例数一般非常多,全部运行需要非常大的时间开销和计算资源使用量(例如,一些商业软件的测试需要数周才能全部运行完毕)。这种巨大的开销对于快速迭代的软件项目来说是难以接受的。因此,回归测试一般会挑选并运行可能受修改影响的部分测试样例,这个过程被称为回归测试选择(Regression Test Selection,简称RTS)。现有的RTS技术普遍针对的是一般的软件类型,但针对RESTful API的回归测试场景,则比较少有相关的研究或发明。原有的RTS技术大多需要对代码逻辑进行分析、计算代码间依赖关系,无法直接被应用在抽象程度较高的RESTful API的软件形态上。

[0042] 目前为了保证一定的测试代码覆盖信息往往需要大量的测试数据,从而导致测试时间过长,无法满足产品的设计周期。然而在这些测试数据之中,一些测试数据起到的作用是一样的,因此如果能将这些作用相同的测试数据进行缩减,即将重复的测试数据删除,则就能够大幅度地减小所需要的测试数据。在本发明的实施例中,可通过代码覆盖信息来检测哪些测试数据是整个样本的覆盖信息贡献最大。因此在本发明的实施例中,保留对样本

覆盖信息贡献最大的作为待执行样例。通过这样的对比,可以极大地减小所需要的测试数据。

[0043] 如图1所示,本发明实施例提供一种自适应回归随机测试方法,自适应回归随机测试方法包括以下步骤:

[0044] 步骤S100,获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;其中,所述初始执行结果为所述初始样例的覆盖信息。

[0045] 在本实施例中,在预准备阶段,该方案中对源代码进行编译,生成了相应的字节码。在需要统计覆盖信息的部分源代码中利用Soot工具进行插桩,使之能够在运行的过程中统计相应的覆盖信息。最后,将插桩过后的代码与未被插桩的代码进行合并,得到新的源代码,并将其编译,得到待测系统的可执行文件。

[0046] 在本实施例中,在信息收集阶段,该方案中首先针对待测网络应用的信息,利用openapi生成描述接口信息的yaml文件,随机生成测试样例。这些生成的测试样例即对应于上述的初始样例,也可以在系统发生变更时,对应于新生成的新生样例。该测试样例以报文为格式,且符合RESTful API的规范。再向待测系统发送在由控制器随机生成测试样例中随机生成的报文并执行时,则代表 workflow 进入了测试样例执行模型。在执行测试样例之后,通过信息反馈模型统计覆盖信息。传递的覆盖信息可以有多种算法,作为举例,其中一种算法为:对每个测试样例都生成一串01字符串,其中,某一位上的0代表相应行的代码未被覆盖,而1代表相应行的代码已被覆盖。最后该方案中将提取反馈模型中的覆盖信息,以供样例筛选阶段使用。

[0047] 在本实施例的一种实现方式中,步骤S100包括以下步骤:

[0048] 步骤S101,获取所述初始样例;

[0049] 步骤S102,记录所述初始样例的覆盖信息,存入迭代样例集中;其中,所述初始样例的覆盖信息包括所述初始样例的覆盖率和代码行的覆盖信息。

[0050] 在本实施例中,被测系统上线前:获取初始测试样例集,并执行样例集中的全部测试样例,监测被测系统,记录被测系统的代码覆盖情况。这些覆盖情况一般指测试样例所执行的被测系统中的行数或方法数等。将样例与各自的覆盖情况一一对应,存入迭代样例集中。

[0051] 在本实施例中,所述服务器略去所述至少两次覆盖信息统计中的源数据中的注释和格式变化,通过计算所述至少两次覆盖信息统计中的源数据获得与所述源数据对应的函数的唯一标识。

[0052] 在本实施例中,所述服务器存储所述唯一标识,和/或,服务器存储所述至少两次覆盖信息统计中的第一数据。

[0053] 在本实施例中,所述覆盖信息统计包括:针对相同的初始样例所属的文件的序号,对文件中的初始样例进行测试,将所述初始样例中已测试的块标记为被覆盖的块,统计所述初始样例的覆盖信息情况。

[0054] 在本实施例中,所述至少两次最新覆盖信息统计包括:针对相同的文件序号,按照时间顺序,分别统计所述文件中初始样例在不同时间的覆盖信息情况。

[0055] 在本实施例中,可通过BullseyeCoverage软件中的cov_eventCount库函数生成对

应的代码覆盖信息数据。当然在本实施例中,还可使用其他软件获得代码覆盖信息数据。

[0056] 如图1所示,在本发明实施例的一种实现方式中,自适应回归随机测试方法还包括以下步骤:

[0057] 步骤S200,获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;其中,所述新生执行结果为所述新生样例的覆盖信息;

[0058] 在本实施例的一种实现方式中,步骤S200包括以下步骤:

[0059] 步骤S201,获取所述新生样例;

[0060] 步骤S202,记录所述新生样例的覆盖信息,存入迭代样例集中。

[0061] 在本实施例中,被测系统更新阶段:当被测系统出现了添加新功能、修改旧功能或修复软件缺陷等事件时,通常会有一部分新的测试样例(即新生样例)。当出现新生样例后,测试系统将会执行并记录新生样例的执行结果,以及样例对被测系统的代码覆盖情况,这里的覆盖情况的信息格式与上述被测系统上线前所记录的信息格式保持一致。

[0062] 在本实施例中,在不断迭代的过程中,总会有新的新生样例产生。每次进行回归测试前,都使用评估和筛选模块对迭代样例集进行筛选,并且只执行迭代样例集中能对新生样例的覆盖情况产生补充的那部分原有测试用例。通过这种做法,能够在保持原有覆盖信息的基础上,尽可能地减少被执行的原有的测试用例数目,且逐渐排除覆盖信息较低的测试用例。最终选出的被执行的测试用例集,会在达到原有覆盖信息的同时,避免了测试用例数目的无限增长,从而最大程度地降低每次执行回归测试用例时产生的各种消耗。

[0063] 如图1所示,在本发明实施例的一种实现方式中,自适应回归随机测试方法还包括以下步骤:

[0064] 步骤S300,对所述迭代样例集筛选,获取对所述迭代样例集覆盖信息贡献最大的样例作为待执行样例。

[0065] 在本实施例中,在ART样例筛选与评估阶段,首先利用将覆盖信息传输至测试样例筛选模型中提取的覆盖信息作为指导信息,指导样例的筛选;利用将覆盖信息传输至测试样例筛选模型中输入的候选测试样例文件作为待筛选测试样例。

[0066] 在本实施例的一种实现方式中,步骤S300包括以下步骤:

[0067] 步骤S301,根据所述迭代样例集的总体覆盖率信息,定义并且更新所述初始样例和所述新生样例的距离。

[0068] 常用的差异度量算法有:基于编辑距离的定义、权重加和法等。现有技术中,对差异的度量大多从测试用例本身出发:例如,将测试用例视作字符串,取其编辑距离大小作为差异的度量,或是赋予测试用例对象的各个属性以不同权重,将存在差异的属性的权重之和作为差异的大小。这些方法的缺陷在于,计算测试之间差异的时候仅考虑了测试的代码,而忽视了测试用例对被测系统的覆盖情况。具体在于,有些测试用例,虽然内容上差异很大,但实际对被测系统的调用路径却几乎一致。这样的现象常见于登录模块、错误处理模块、表单填写模块等系统模块中。

[0069] 在本实施例中,针上述现象,本发明提出的以被测系统对测试用例的反应,即测试用例的代码覆盖情况,为差异大小的定义方法。相比于过往技术,本发明对被测系统具有更好的适应性,能够一定程度上克服过往方法中对被测系统关注相对缺失的不足。

[0070] 在本实施例的一种实现方式中,步骤S300还包括以下步骤:

[0071] 步骤S303,基于每个初始样例和每个新生样例对应的距离,使用贪心算法,迭代地选取出所述迭代样例集中的距离最大的样例,作为待执行样例,并放入迭代样例集。

[0072] 在本实施例中,在具体的筛选过程中,本发明利用了贪心算法,依据每个测试样例对应的覆盖信息,逐步选取出对当前系统覆盖信息贡献最大的测试样例,作为待执行样例,并放入迭代样例集。

[0073] 在本实施例中,在计算对系统覆盖信息的贡献时,该方案选取了当前测试样例的覆盖信息和已执行测试样例的覆盖信息的距离作为参考。若距离越大说明贡献越多。其中距离的定义方式为多样的。例如,对于上述的01字符串表示,两个测试样例的覆盖情况的差异可以是其对应的01字符串的海明距离或欧几里得距离。另外,还可以依据对资源的操作来定义距离。例如,在RESTful API中,对于user资源可以GET/POST/DELETE/PUT等种操作,在该方案中需要优先保证所有资源都有操作被覆盖。为了达到这个目标,可以用两个资源的“未被覆盖过的操作数量的差值”作为两者距离的度量,例如,两个资源分别覆盖信息4种和1种操作,则它们的距离为3,这样,比起已经覆盖了所有操作的资源来说,仅有少量操作被覆盖的资源就有更大的机会被选择到。

[0074] 如图1所示,在本发明实施例的一种实现方式中,自适应回归随机测试方法还包括以下步骤:

[0075] 步骤S400,计算所述待执行样例性能值,并返回计算结果。

[0076] 在本实施例的一种实现方式中,步骤S400包括以下步骤:

[0077] 步骤S401,将所述待执行样例传输至性能分析模型中进行计算,得到所述待执行样例的模型性能值,并返回;其中,所述性能分析模型采用模型性能值计算公式进行计算。

[0078] 在本实施例的一种实现方式中,步骤S401包括以下步骤:

[0079] 步骤S4011,所述模型性能值计算公式为: $E = \frac{a}{b} \div \frac{c}{d}$,其中,E为模型性能值,a为筛选的样例的覆盖信息,b为所有样例的覆盖信息,c为筛选的样例数,d为样例总数。

[0080] 在本实施例中,当经筛选后的测试样例能够覆盖到未筛选前测试所覆盖的所有方法或行数时,停止继续筛选。在性能分析阶段,系统会量化地计算并评估某一次样例筛选的效果好坏,具体来说,采用如下公式: $E = \frac{a}{b} \div \frac{c}{d}$,对于除号的前后两项,前一项越大(其最大值为1),则说明筛选的样例能更为贴近全体样例的覆盖情况,则筛选效果越好;后一项越小,说明筛选的样例数越少。这样,当公式的值越大时,说明样例筛选的结果更好,即同时满足高覆盖信息和少样例数的需求。

[0081] 本实施例通过代码覆盖信息的变化情况,判断对应的输入数据(即初始样例和新生样例)对代码覆盖信息是否构成了贡献,如果没有贡献则可将该测试样例的输入数据剔除,从而可以在保证代码覆盖信息不变的前提下可以有效地将原来千万量级的输入数据压缩至百量级,因此可以大幅度地提高测试速度,及缩短测试周期。另外,在本申请的实施例中,上述方法完全自动运行,无需人工干预,因此可以节约人力成本。

[0082] 基于上述自适应回归随机测试方法,本实施例提供了一种自适应回归随机测试装置,包括:

[0083] 测试样例生成模块1,用于获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;还用于获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;

[0084] 测试样例筛选模块2,用于对所述迭代样例集筛选,获取待执行样例;

[0085] 性能分析模块3,对所述迭代样例集进行性能分析。

[0086] 示例性设备

[0087] 基于上述实施例,本发明还提供一种终端,其原理框图可以如图2所示。

[0088] 该终端包括:通过系统总线连接的处理器、存储器、接口、显示屏以及通讯模块;其中,该终端的处理器用于提供计算和控制能力;该终端的存储器包括计算机可读存储介质以及内存储器;该计算机可读存储介质存储有操作系统和计算机程序;该内存储器为计算机可读存储介质中的操作系统和计算机程序的运行提供环境;该接口用于连接外部终端设备,例如,移动终端以及计算机等设备;该显示屏用于显示相应的自适应回归随机测试信息;该通讯模块用于与云端服务器或移动终端进行通讯。

[0089] 该计算机程序被处理器执行时用以实现一种自适应回归随机测试方法。

[0090] 本领域技术人员可以理解的是,图2中示出的原理框图,仅仅是与本发明方案相关的部分结构的框图,并不构成对本发明方案所应用于其上的终端的限定,具体的终端可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0091] 在一个实施例中,提供了一种终端,其中,包括:处理器和存储器,存储器存储有自适应回归随机测试程序,该自适应回归随机测试程序被处理器执行时用于实现如上所述的自适应回归随机测试方法。

[0092] 在一个实施例中,提供了一种计算机可读存储介质,其中,该计算机可读存储介质存储有自适应回归随机测试程序,自适应回归随机测试程序被处理器执行时用于实现如上所述的自适应回归随机测试方法。

[0093] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本发明所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。

[0094] 综上,本发明公开了一种自适应回归随机测试方法,其中,方法包括:获取初始样例,根据所述初始样例得到初始执行结果,将所述初始样例与所述初始执行结果存入迭代样例集中;所述初始执行结果为所述初始样例的覆盖信息;获取新生样例,根据所述新生样例得到新生执行结果,将所述新生样例与所述新生执行结果存入所述迭代样例集中;其中,所述新生执行结果为所述新生样例的覆盖信息;对所述迭代样例集筛选,获取待执行样例;计算所述待执行样例性能值,并返回计算结果。本发明根据测试用例的代码覆盖情况,定义差异大小。本发明可以大幅度地提高测试速度、缩短测试周期及节约测试人力成本。

[0095] 应当理解的是,本发明的应用不限于上述的举例,对本领域普通技术人员来说,可以根据上述说明加以改进或变换,所有这些改进和变换都应属于本发明所附权利要求的保护范围。

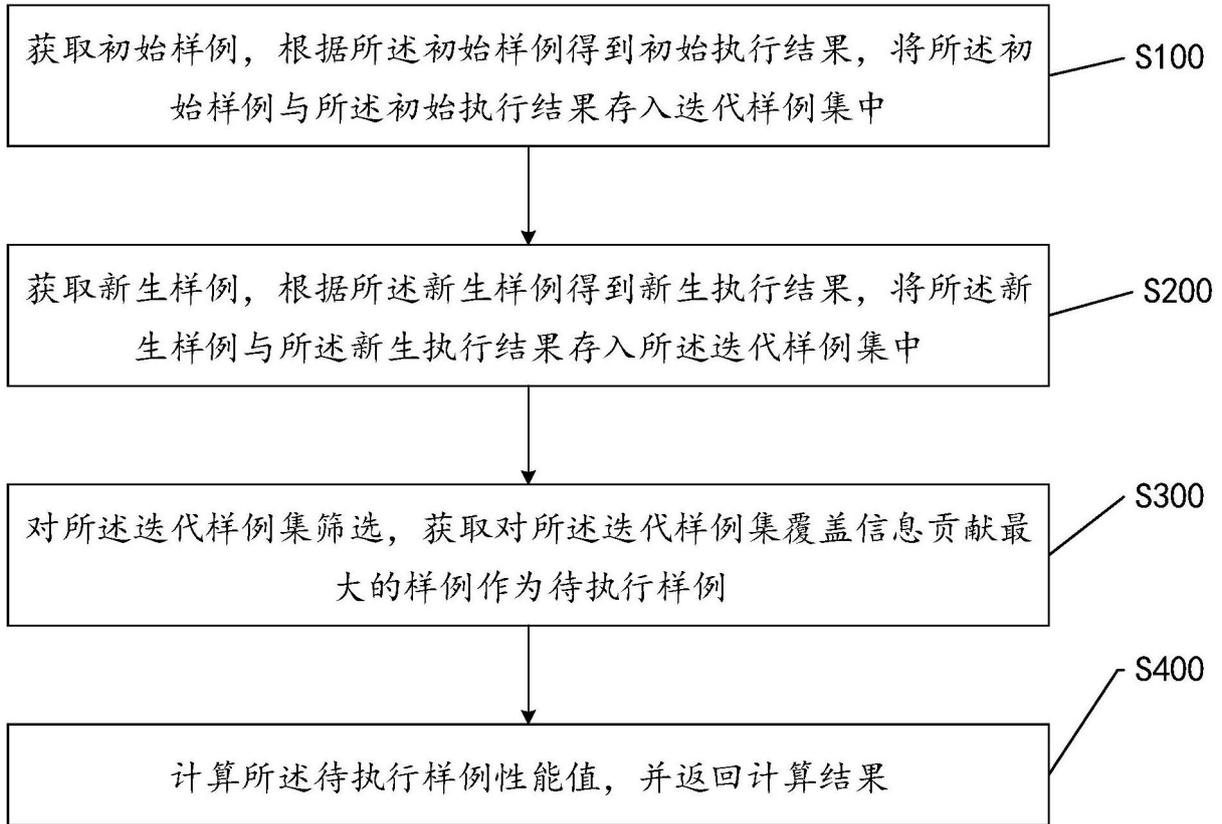


图1

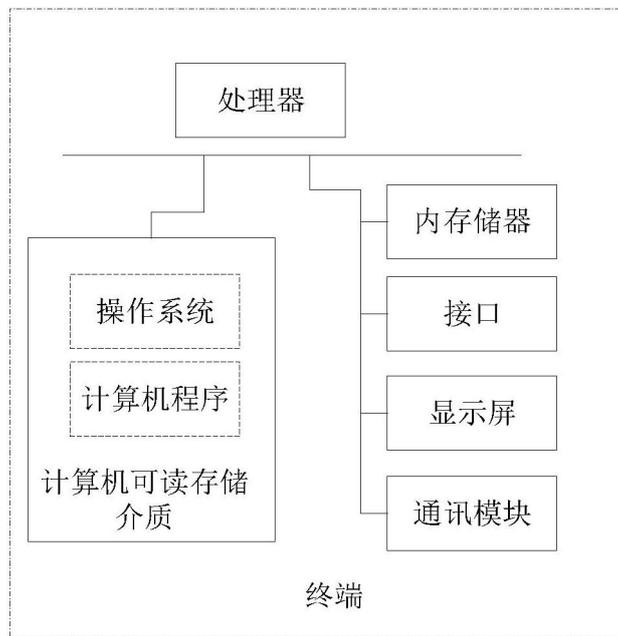


图2

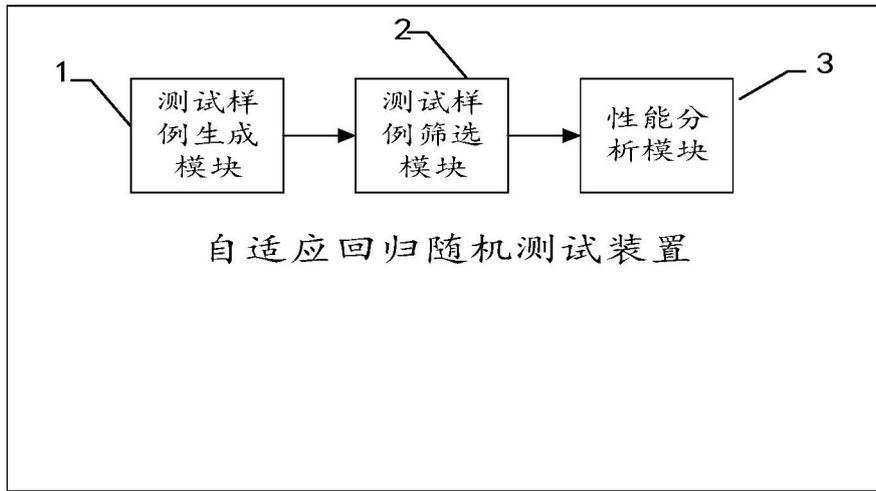


图3