



(12)发明专利申请

(10)申请公布号 CN 108664372 A

(43)申请公布日 2018.10.16

(21)申请号 201810432475.6

(22)申请日 2018.05.08

(71)申请人 平安科技(深圳)有限公司

地址 518000 广东省深圳市福田区八卦岭
工业区平安大厦六楼

(72)发明人 邹文涛 毛皓 李克标

(74)专利代理机构 深圳市沃德知识产权代理事
务所(普通合伙) 44347

代理人 于志光 郭梦霞

(51)Int.Cl.

G06F 11/30(2006.01)

G06F 11/36(2006.01)

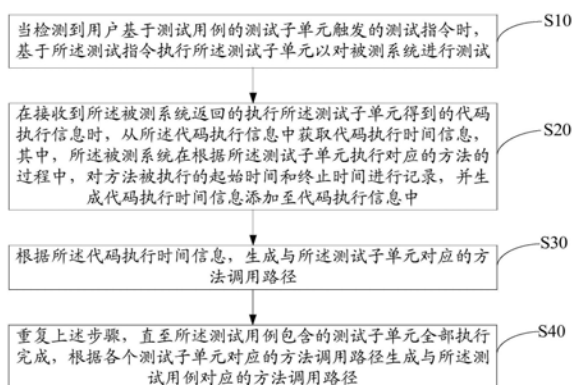
权利要求书3页 说明书10页 附图2页

(54)发明名称

测试过程的监控装置、方法及计算机可读存储介质

(57)摘要

本发明公开了一种测试过程的监控装置,包括存储器和处理器,存储器上存储有可在处理器上运行的测试监控程序,该程序被处理器执行时实现如下步骤:当检测到用户基于测试用例的测试子单元触发的测试指令时,基于测试指令执行测试子单元以对被测系统进行测试;在接收到被测系统返回的执行测试子单元得到的代码执行信息时,从代码执行信息中获取时间信息;根据时间信息,生成与测试子单元对应的方法调用路径;重复上述步骤,直至测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与测试用例对应的方法调用路径。本发明还提出一种测试过程的监控方法以及一种计算机可读存储介质。本发明实现了准确高效的进行测试。



1. 一种测试过程的监控装置,其特征在于,所述装置包括存储器和处理器,所述存储器上存储有可在所述处理器上运行的测试监控程序,所述测试监控程序被所述处理器执行时实现如下步骤:

当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

2. 如权利要求1所述的测试过程的监控装置,其特征在于,所述测试监控程序还可被所述处理器执行,以在所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,还实现如下步骤:

在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

3. 如权利要求1所述的测试过程的监控装置,其特征在于,所述测试监控程序还可被所述处理器执行,以在所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,还实现如下步骤:

获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

4. 如权利要求3所述的测试过程的监控装置,其特征在于,所述测试监控程序还可被所述处理器执行,以在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况进行实时监控,获取所述变更代码和所述关联代码的被执行情况的步骤之后,还实现如下步骤:

根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;

根据所述测试覆盖信息和各个测试子单元对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

5. 如权利要求1至4中任一项所述的测试过程的监控装置,其特征在于,所述根据所述

代码执行时间信息,生成与所述测试子单元对应的方法调用路径的步骤包括:

根据所述代码执行时间信息确定与该测试子单元对应的各个方法的方法名,以及各个方法执行的先后顺序;

根据所述方法名和各个方法被执行的先后顺序,生成所述测试子单元执行过程中被测系统的方法调用路径。

6. 一种测试过程的监控方法,其特征在于,所述方法包括:

当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

7. 如权利要求6所述的测试过程的监控方法,其特征在于,所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,所述方法还包括:

在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

8. 如权利要求6所述的测试过程的监控方法,其特征在于,所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,所述方法还包括:

获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

9. 如权利要求8所述的测试过程的监控方法,其特征在于,在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况进行实时监控,获取所述变更代码和所述关联代码的被执行情况的步骤之后,所述方法还包括:

根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;

根据所述测试覆盖信息和各个测试子单元对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

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上存储有测试监

控程序,所述测试监控程序可被一个或者多个处理器执行,以实现如权利要求6至9中任一项所述的测试过程的监控方法的步骤。

测试过程的监控装置、方法及计算机可读存储介质

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种测试过程的监控装置、方法及计算机可读存储介质。

背景技术

[0002] 现有的基于人工和自动化进行的系统测试中,需要根据测试情况对被测系统中的缺陷进行定位,而现有的测试方法中,根据测试用例的执行成功率来对测试情况评估,无法实现代码层级的具体缺陷定位,主要是由于现有的测试方法无法根据对测试过程的监控,获取到被测系统在测试过程中的方法调用路径,如果要获取被测系统的方法调用路径,只能通过静态分析的方法对被测系统的源代码的逻辑进行分析,自行绘制方法调用路径图,进而根据测试结果以及方法调用路径图对被测系统的缺陷进行代码层级的定位,这种方式不仅效率低下而且导致测试的准确率较低。

发明内容

[0003] 本发明提供一种测试过程的监控装置、方法及计算机可读存储介质,其主要目的在于解决无法根据对测试过程的监控,获取到被测系统在测试过程中的方法调用路径的技术问题,实现准确高效的进行测试。

[0004] 为实现上述目的,本发明提供一种测试过程的监控装置,该装置包括存储器和处理器,所述存储器中存储有可在所述处理器上运行的测试监控程序,所述测试监控程序被所述处理器执行时实现如下步骤:

[0005] 当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

[0006] 在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

[0007] 根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

[0008] 重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0009] 可选地,所述测试监控程序还可被所述处理器执行,以在所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,还实现如下步骤:

[0010] 在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

[0011] 可选地,所述测试监控程序还可被所述处理器执行,以在所述当检测到用户基于

测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,还实现如下步骤:

[0012] 获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

[0013] 根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

[0014] 在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

[0015] 根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

[0016] 可选地,所述测试监控程序还可被所述处理器执行,以在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况进行实时监控,获取所述变更代码和所述关联代码的被执行情况的步骤之后,还实现如下步骤:

[0017] 根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;

[0018] 根据所述测试覆盖信息和各个测试子单元对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

[0019] 可选地,所述根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径的步骤包括:

[0020] 根据所述代码执行时间信息确定与该测试子单元对应的各个方法的方法名,以及各个方法执行的先后顺序;

[0021] 根据所述方法名和各个方法被执行的先后顺序,生成所述测试子单元执行过程中被测系统的方法调用路径。

[0022] 此外,为实现上述目的,本发明还提供一种测试过程的监控方法,该方法包括:

[0023] 当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

[0024] 在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

[0025] 根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

[0026] 重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0027] 可选地,所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,所述方法还包括:

[0028] 在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

[0029] 可选地,所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,所述方法还包括:

[0030] 获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

[0031] 根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

[0032] 在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

[0033] 根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

[0034] 可选地,在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况进行实时监控,获取所述变更代码和所述关联代码的被执行情况的步骤之后,所述方法还包括:

[0035] 根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;

[0036] 根据所述测试覆盖信息和各个测试子单元对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

[0037] 此外,为实现上述目的,本发明还提供一种计算机可读存储介质,所述计算机可读存储介质上存储有测试监控程序,所述测试监控程序可被一个或者多个处理器执行,以实现如下步骤:

[0038] 当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

[0039] 在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

[0040] 根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

[0041] 重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0042] 本发明提出的测试过程的监控装置、方法及计算机可读存储介质,用户逐个基于测试用例中的测试子单元触发测试指令,对被测系统进行测试,被测系统执行测试子单元得到代码执行信息,并且在根据测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中,当接收到被测系统发送的代码执行信息时,从中获取时间信息,并根据获取到的时间信息获取与测试子单元对应的方法调用路径,重复上述步骤直至测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成测试用例对应的方法调用路径,本发明的方案对使用测试用例对被测系统进行测试的过程进行监控,获取测试过程中每一个方法被执行的时间信息,进而根据时间信息自动生成与测试案例对应的方法调用路径,解决了无法根据对测试过程的监控,获取到被测系统在测试过程中的方法调用路径的技术问题,实现准确高效的进行测试。

附图说明

- [0043] 图1为本发明测试过程的监控装置较佳实施例的示意图；
- [0044] 图2为本发明测试过程的监控装置一实施例中测试监控程序的程序模块示意图；
- [0045] 图3为本发明测试过程的监控方法第一实施例的流程图。
- [0046] 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

[0047] 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。

[0048] 本发明提供一种测试过程的监控装置。参照图1所示，为本发明测试过程的监控装置较佳实施例的示意图。

[0049] 在本实施例中，测试过程的监控装置1可以是PC (Personal Computer, 个人电脑)，也可以是智能手机、平板电脑、便携计算机等终端设备。该测试过程的监控装置1至少包括存储器11、处理器12，通信总线13，以及网络接口14。

[0050] 其中，存储器11至少包括一种类型的可读存储介质，所述可读存储介质包括闪存、硬盘、多媒体卡、卡型存储器（例如，SD或DX存储器等）、磁性存储器、磁盘、光盘等。存储器11在一些实施例中可以是测试过程的监控装置1的内部存储单元，例如该测试过程的监控装置1的硬盘。存储器11在另一些实施例中也可以是测试过程的监控装置1的外部存储设备，例如测试过程的监控装置1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，存储器11还可以既包括测试过程的监控装置1的内部存储单元也包括外部存储设备。存储器11不仅可以用于存储安装于测试过程的监控装置1的应用软件及各类数据，例如测试监控程序01的代码等，还可以用于暂时地存储已经输出或者将要输出的数据。

[0051] 处理器12在一些实施例中可以是一中央处理器 (Central Processing Unit, CPU)、控制器、微控制器、微处理器或其他数据处理芯片，用于运行存储器11中存储的程序代码或处理数据，例如执行测试监控程序01等。

[0052] 通信总线13用于实现这些组件之间的连接通信。

[0053] 网络接口14可选的可以包括标准的有线接口、无线接口（如WI-FI接口），通常用于在该装置1与其他电子设备之间建立通信连接。

[0054] 图1仅示出了具有组件11-14以及测试监控程序01的测试过程的监控装置1，但是应理解的是，并不要求实施所有示出的组件，可以替代的实施更多或者更少的组件。

[0055] 可选地，该装置1还可以包括用户接口，用户接口可以包括显示器 (Display)、输入单元比如键盘 (Keyboard)，可选的用户接口还可以包括标准的有线接口、无线接口。可选地，在一些实施例中，显示器可以是LED显示器、液晶显示器、触控式液晶显示器以及OLED (Organic Light-Emitting Diode, 有机发光二极管) 触摸器等。其中，显示器也可以适当的称为显示屏或显示单元，用于显示在测试过程的监控装置1中处理的信息以及用于显示可视化的用户界面。

[0056] 在图1所示的装置1实施例中，存储器11中存储有测试监控程序01；处理器12执行存储器11中存储的测试监控程序01时实现如下步骤：

[0057] 当检测到用户基于测试用例的测试子单元触发的测试指令时，基于所述测试指令执行所述测试子单元以对被测系统进行测试。

[0058] 在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中。

[0059] 本实施例的装置上存储有预先编写好的测试用例,一个测试用例可以包括多个测试子单元,测试人员在进行测试时,建立该装置与被测终端之间的通讯连接,其中,被测系统运行在被测终端上。测试人员逐个基于测试子单元触发测试指令,监控装置在检测到测试指令时,根据对应的测试子单元对被测系统进行测试,被测系统根据测试子单元运行相应的代码。

[0060] 具体地,对于被测系统来说,在测试过程中,由代码执行容器来执行对应的系统源代码,并且在执行源代码的过程对执行各个方法的时间信息进行记录,主要包括各个方法被执行的开始时间和执行完成的终止之间,在测试子单元完成之后,将记录的时间信息添加至执行代码信息发送到监控装置。

[0061] 可选地,作为一种实施方式,在本实施例的监控装置上预先设置好一段时间记录代码,该时间记录代码用于实现对方法的被执行时间的记录。在测试开始之间,测试人员在监控装置上触发代码发送指令,监控装置在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

[0062] 被测终端在接收到时间记录代码时,将该时间记录代码存储至缓存区域。当根据测试子单元执行对应的方法的代码时,代码执行容器将待执行的方法加载至容器中,并从缓存区域获取上述时间记录代码,分别将时间记录代码在即将被执行的方法的代码之前和代码之后添加时间记录代码,然后执行加载了时间记录代码的方法,在该方法的执行过程中,时间记录代码对方法被执行的起始时间和终止时间进行记录。需要说明的是,在根据一个测试子单元进行测试时,在被测系统中可能有多个方法需要执行,因此,需要在每一个方法的代码的前后添加时间记录代码,以对每一个方法的被执行的起始时间和终止时间进行记录。

[0063] 根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径。

[0064] 重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0065] 监控装置在接收到被测终端发送的执行上述测试子单元得到的代码执行信息时,从代码执行信息中获取时间信息,进而生成方法调用路径,具体地,根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径的步骤包括:根据所述代码执行时间信息确定与该测试子单元对应的各个方法的方法名,以及各个方法执行的先后顺序;根据所述方法名和各个方法被执行的先后顺序,生成所述测试子单元执行过程中被测系统的方法调用路径。

[0066] 监控装置从代码执行信息中获取到测试子单元被执行中的各个方法被执行的开始时间和终止时间,通过这些时间信息能够确定各个方法被执行的先后顺序,通过先后顺序能够确定多个方法之间的调用关系,然后根据方法名和各个方法被执行的先后顺序生成测试子单元执行过程中被测系统的方法调用路径。

[0067] 可以理解的是,一个测试案例中一般包含有多个测试子单元,测试人员会逐个进行测试。重复执行上述步骤,当测试案例中的测试子单元全部执行完成后,根据各个测试子单元的方法调用路径生成测试用例对应的方法调用路径。进一步地,将得到的方法调用路径展示在显示界面上,测试人员在根据测试结果对测试情况进行分析以对被测系统的缺陷进行定位时,可以根据展示的该测试案例对应的方法调用路径实现对缺陷的快速定位,实现准确高效的进行测试。

[0068] 进一步地,在一实施例中,测试监控程序还可被处理器执行,以实现如下步骤:在测试过程中,对被测系统的源代码的代码测试覆盖率进行实时监控,并将监控得到的代码测试覆盖率实时地展示在显示界面上。

[0069] 以上实施例提出的测试过程的监控装置,用户逐个基于测试用例中的测试子单元触发测试指令,对被测系统进行测试,被测系统执行测试子单元得到代码执行信息,并且在根据测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中,当接收到被测系统发送的代码执行信息时,从中获取时间信息,并根据获取到的时间信息获取与测试子单元对应的方法调用路径,重复上述步骤直至测试用例包含的测试子单元全部执行完成,根据每个测试子单元对应的方法调用路径生成测试用例对应的方法调用路径,本发明的方案对使用测试用例对被测系统进行测试的过程进行监控,获取测试过程中每一个方法被执行的时间信息,进而根据时间信息自动生成与测试案例对应的方法调用路径,解决了无法根据对测试过程的监控,获取到被测系统在测试过程中的方法调用路径的技术问题,实现准确高效的进行测试。

[0070] 进一步地,在另一实施例中,测试监控程序还可被所述处理器执行,以在所述当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试的步骤之前,还实现如下步骤:

[0071] 获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

[0072] 根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

[0073] 在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

[0074] 根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

[0075] 本实施例的监控装置还可对被测系统的代码变更情况进行监控,首先根据当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定哪些代码行是发生变更的代码行,包括修改和新增的代码,然后确定与这些代码行具有关联关系的代码。在测试过程中,对这些代码的执行情况进行监控,获取针对变更代码和所述关联代码的测试覆盖率,因为对于被测系统来说,系统升级以后,新版本系统的新功能是主要的测试对象,因此,通过对变更代码的测试覆盖率进行实时地监测,有益于测试人员及时了解到针对这些新功能的测试执行情况。

[0076] 进一步地,在另一实施例中,测试监控程序还可被所述处理器执行,以实现如下步骤:根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;根据所述测试覆盖信息和各个测试子单元

对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

[0077] 通过对变更代码和其关联代码的被执行情况监控,获取变更代码和关联代码的测试覆盖信息,也就是说获取到使用该测试用例进行测试的过程中,变更代码和关联代码中有哪些代码行被执行了。而方法与代码行之间有对应关系,进而结合方法调用路径,可以确定每个方法中被执行的代码行,进而可以得到测试子单元对应的被执行的代码行,即获取到测试子单元与被测系统的代码行之间的映射关系。这里的被测系统的代码行主要是上述变更代码和关联代码中的代码行。在其他实施例中,还可以结合完整的被测系统的代码覆盖情况与方法调用路径获取完整的代码行与测试子单元之间的映射关系。

[0078] 进一步地,将上述映射关系展示在显示界面上,以供测试人员根据对测试情况进行分析,实现对缺陷的快速定位,进一步提高测试的效率和准确度。

[0079] 可选地,在其他的实施例中,测试监控程序还可以被分割为一个或者多个模块,一个或者多个模块被存储于存储器11中,并由一个或多个处理器(本实施例为处理器12)所执行以完成本发明,本发明所称的模块是指能够完成特定功能的一系列计算机程序指令段,用于描述测试监控程序在测试过程的监控装置中的执行过程。

[0080] 例如,参照图2所示,为本发明测试过程的监控装置一实施例中的测试监控程序的程序模块示意图,该实施例中,测试监控程序可以被分割为测试执行模块10、信息获取模块20和路径生成模块30,示例性地:

[0081] 测试执行模块10用于:当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

[0082] 信息获取模块20用于:在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

[0083] 路径生成模块30用于:根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

[0084] 测试执行模块10、信息获取模块20和路径生成模块30重复执行上述操作,直至测试用例包含的测试子单元全部执行完成,路径生成模块30根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0085] 上述测试执行模块10、信息获取模块20和路径生成模块30等程序模块被执行时所实现的功能或操作步骤与上述实施例大体相同,在此不再赘述。

[0086] 此外,本发明还提供一种测试过程的监控方法。参照图3所示,为本发明测试过程的监控方法第一实施例的流程图。该方法可以由一个装置执行,该装置可以由软件和/或硬件实现。以下实施例中以测试过程的监控装置作为执行主体对本发明的方法进行说明。

[0087] 在本实施例中,测试过程的监控方法包括:

[0088] 步骤S10,当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试。

[0089] 步骤S20,在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成

代码执行时间信息添加至代码执行信息中。

[0090] 作为本实施例的执行主体的监控装置上存储有预先编写好的测试用例,一个测试用例可以包括多个测试子单元,测试人员在进行测试时,建立监控装置与被测终端之间的通讯连接,其中,被测系统运行在被测终端上。测试人员逐个基于测试子单元触发测试指令,监控装置在检测到测试指令时,根据对应的测试子单元对被测系统进行测试,被测系统根据测试子单元运行相应的代码。

[0091] 具体地,对于被测系统来说,在测试过程中,由代码执行容器来执行对应的系统源代码,并且在执行源代码的过程对执行各个方法的时间信息进行记录,主要包括各个方法被执行的开始时间和执行完成的终止之间,在测试子单元完成之后,将记录的时间信息添加至执行代码信息发送到监控装置。

[0092] 可选地,作为一种实施方式,在监控装置上预先设置好一段时间记录代码,该时间记录代码用于实现对方法的被执行时间的记录。在测试开始之间,测试人员在监控装置上触发代码发送指令,监控装置在检测到代码发送指令时,获取时间记录代码,并将所述时间记录代码发送至所述被测系统所在的终端,以供所述终端存储所述时间记录代码并根据所述时间记录代码对方法被执行的起始时间和终止时间进行记录。

[0093] 被测终端在接收到时间记录代码时,将该时间记录代码存储至缓存区域。当根据测试子单元执行对应的方法的代码时,代码执行容器将待执行的方法加载至容器中,并从缓存区域获取上述时间记录代码,分别将时间记录代码在即将被执行的方法的代码之前和代码之后添加时间记录代码,然后执行加载了时间记录代码的方法,在该方法的执行过程中,时间记录代码对方法被执行的起始时间和终止时间进行记录。需要说明的是,在根据一个测试子单元进行测试时,在被测系统中可能有多个方法需要执行,因此,需要在每一个方法的代码的前后添加时间记录代码,以对每一个方法的被执行的起始时间和终止时间进行记录。

[0094] 步骤S30,根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径。

[0095] 步骤S40,重复步骤S10至步骤S30,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0096] 监控装置在接收到被测终端发送的执行上述测试子单元得到的代码执行信息时,从代码执行信息中获取时间信息,进而生成方法调用路径,具体地,步骤S30包括以下细化步骤:根据所述代码执行时间信息确定与该测试子单元对应的各个方法的方法名,以及各个方法执行的先后顺序;根据所述方法名和各个方法被执行的先后顺序,生成所述测试子单元执行过程中被测系统的方法调用路径。

[0097] 监控装置从代码执行信息中获取到测试子单元被执行中的各个方法被执行的开始时间和终止时间,通过这些时间信息能够确定各个方法被执行的先后顺序,通过先后顺序能够确定多个方法之间的调用关系,然后根据方法名和各个方法被执行的先后顺序生成测试子单元执行过程中被测系统的方法调用路径。

[0098] 可以理解的是,一个测试案例中一般包含有多个测试子单元,测试人员会逐个进行测试。重复执行上述步骤,当测试案例中的测试子单元全部执行完成后,根据各个测试子

单元的方法调用路径生成测试用例对应的方法调用路径。进一步地,将得到的方法调用路径展示在显示界面上,测试人员在根据测试结果对测试情况进行分析以对被测系统的缺陷进行定位时,可以根据展示的该测试案例对应的方法调用路径实现对缺陷的快速定位,实现准确高效的进行测试。

[0099] 进一步地,在一实施例中,该方法还包括步骤:在测试过程中,对被测系统的源代码的代码测试覆盖率进行实时监控,并将监控得到的代码测试覆盖率实时地展示在显示界面上。

[0100] 以上实施例提出的测试过程的监控方法,用户逐个基于测试用例中的测试子单元触发测试指令,对被测系统进行测试,被测系统执行测试子单元得到代码执行信息,并且在根据测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中,当接收到被测系统发送的代码执行信息时,从中获取时间信息,并根据获取到的时间信息获取与测试子单元对应的方法调用路径,重复上述步骤直至测试用例包含的测试子单元全部执行完成,根据每个测试子单元对应的方法调用路径生成测试用例对应的方法调用路径,本发明的方案对使用测试用例对被测系统进行测试的过程进行监控,获取测试过程中每一个方法被执行的时间信息,进而根据时间信息自动生成与测试案例对应的方法调用路径,解决了无法根据对测试过程的监控,获取到被测系统在测试过程中的方法调用路径的技术问题,实现准确高效的进行测试。

[0101] 进一步地,在另一实施例中,该方法还包括如下步骤::

[0102] 获取所述被测系统的当前版本的系统源代码,以及历史版本的系统源代码;

[0103] 根据对当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定当前版本的系统源代码的变更代码和所述变更代码的关联代码;

[0104] 在所述测试用例的执行过程中,对所述变更代码和所述关联代码的被执行情况实时监控,获取所述变更代码和所述关联代码的被执行情况;

[0105] 根据所述被执行情况计算所述变更代码和所述关联代码的测试覆盖率,并实时输出计算得到的测试覆盖率。

[0106] 在该实施例中,还可对被测系统的代码变更情况进行监控,首先根据当前版本的系统源代码和历史版本的系统源代码进行对比分析,确定哪些代码行是发生变更的代码行,包括修改和新增的代码,然后确定与这些代码行具有关联关系的代码。在测试过程中,对这些代码的执行情况进行监控,获取针对变更代码和所述关联代码的测试覆盖率,因为对于被测系统来说,系统升级以后,新版本系统的新功能是主要的测试对象,因此,通过对变更代码的测试覆盖率进行实时地监测,有益于测试人员及时了解到针对这些新功能的测试执行情况。

[0107] 进一步地,在另一实施例中,该方法还包括如下步骤:根据所述变更代码和所述关联代码的被执行情况,获取使用所述测试用例进行测试的过程中,变更代码和关联代码的测试覆盖信息;根据所述测试覆盖信息和各个测试子单元对应的方法调用路径,获取测试子单元与被测系统的代码行之间的映射关系。

[0108] 通过对变更代码和其关联代码的被执行情况的监控,获取变更代码和关联代码的测试覆盖信息,也就是说获取到使用该测试用例进行测试的过程中,变更代码和关联代码中有哪些代码行被执行了。而方法与代码行之间有对应关系,进而结合方法调用路径,可以

确定每个方法中被执行的代码行,进而可以得到测试子单元对应的被执行的代码行,即获取到测试子单元与被测系统的代码行之间的映射关系。这里的被测系统的代码行主要是上述变更代码和关联代码中的代码行。在其他实施例中,还可以结合完整的被测系统的代码覆盖情况与方法调用路径获取完整的代码行与测试子单元之间的映射关系。

[0109] 进一步地,将上述映射关系展示在显示界面上,以供测试人员根据对测试情况进行分析,实现对缺陷的快速定位,进一步提高测试的效率和准确度。

[0110] 此外,本发明实施例还提出一种计算机可读存储介质,所述计算机可读存储介质上存储有测试监控程序,所述测试监控程序可被一个或多个处理器执行,以实现如下操作:

[0111] 本发明计算机可读存储介质具体实施方式与上述测试过程的监控装置和方法各实施例基本相同,在此不作累述。

[0112] 当检测到用户基于测试用例的测试子单元触发的测试指令时,基于所述测试指令执行所述测试子单元以对被测系统进行测试;

[0113] 在接收到所述被测系统返回的执行所述测试子单元得到的代码执行信息时,从所述代码执行信息中获取代码执行时间信息,其中,所述被测系统在根据所述测试子单元执行对应的方法的过程中,对方法被执行的起始时间和终止时间进行记录,并生成代码执行时间信息添加至代码执行信息中;

[0114] 根据所述代码执行时间信息,生成与所述测试子单元对应的方法调用路径;

[0115] 重复上述步骤,直至所述测试用例包含的测试子单元全部执行完成,根据各个测试子单元对应的方法调用路径生成与所述测试用例对应的方法调用路径。

[0116] 需要说明的是,上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。并且本文中的术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含,从而使得包括一系列要素的过程、装置、物品或者方法不仅包括那些要素,而且还包括没有明确列出的其他要素,或者是还包括为这种过程、装置、物品或者方法所固有的要素。在没有更多限制的情况下,由语句“包括一个……”限定的要素,并不排除在包括该要素的过程、装置、物品或者方法中还存在另外的相同要素。

[0117] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到上述实施例方法可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在如上所述的一个存储介质(如ROM/RAM、磁碟、光盘)中,包括若干指令用以使得一台终端设备(可以是手机,计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0118] 以上仅为本发明的优选实施例,并非因此限制本发明的专利范围,凡是利用本发明说明书及附图内容所作的等效结构或等效流程变换,或直接或间接运用在其他相关的技术领域,均同理包括在本发明的专利保护范围内。

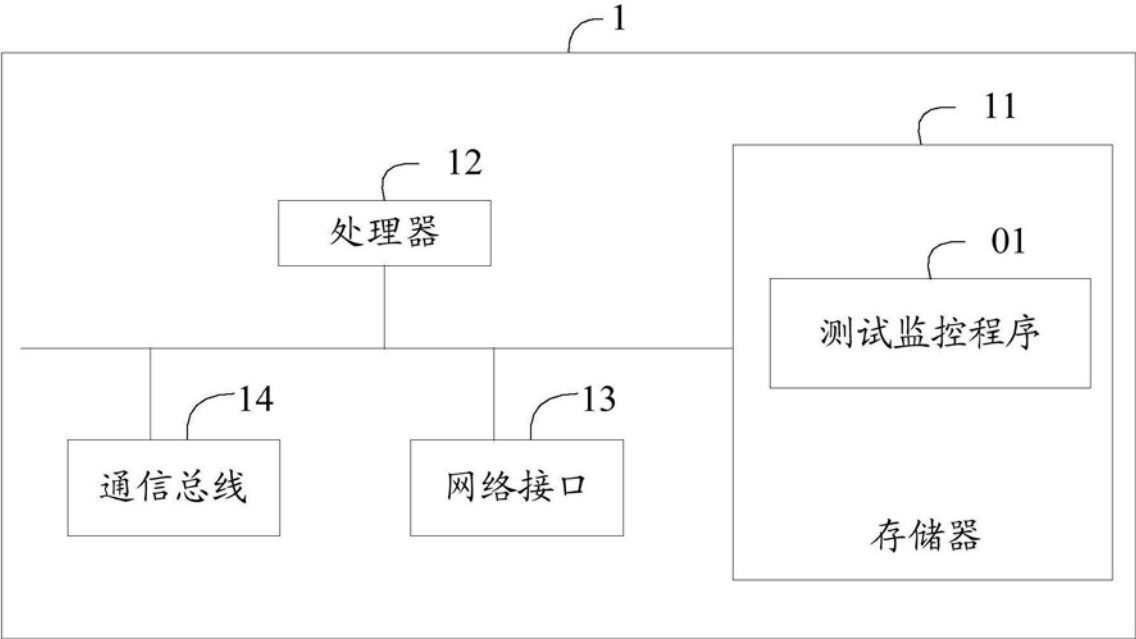


图1



图2

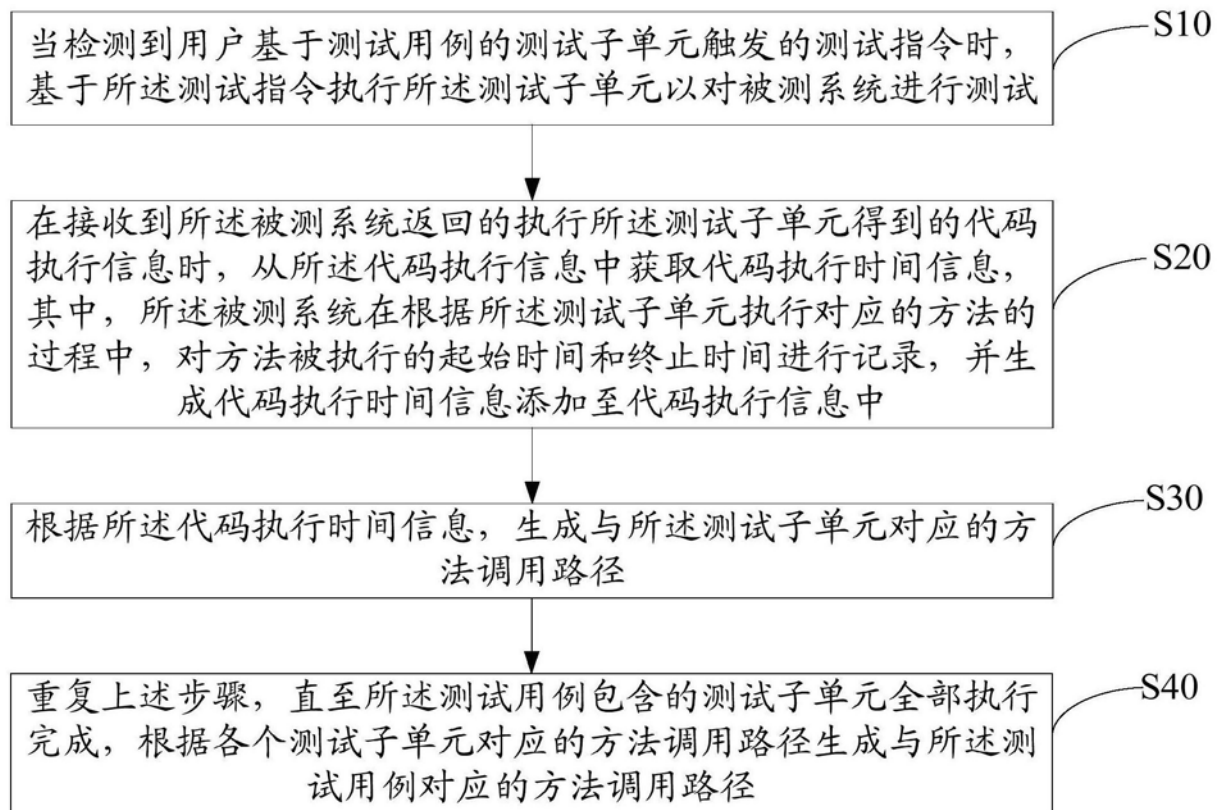


图3