



(12) 发明专利申请

(10) 申请公布号 CN 113760307 A

(43) 申请公布日 2021. 12. 07

(21) 申请号 202011302318.7

(22) 申请日 2020.11.19

(71) 申请人 北京沃东天骏信息技术有限公司
地址 100176 北京市北京经济技术开发区
科创十一街18号院2号楼4层A402室
申请人 北京京东世纪贸易有限公司

(72) 发明人 唐曼玲

(74) 专利代理机构 中原信达知识产权代理有限
责任公司 11219
代理人 韩黎捷 郭晗

(51) Int. Cl.
G06F 8/61 (2018.01)
G06F 8/71 (2018.01)
G06F 11/36 (2006.01)

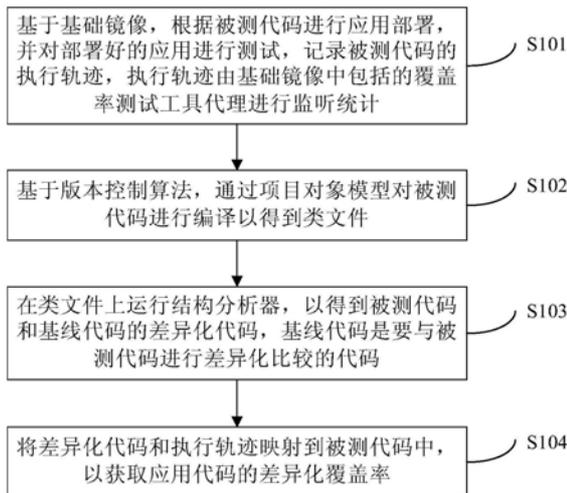
权利要求书2页 说明书11页 附图4页

(54) 发明名称

获取应用代码的差异化覆盖率的方法和装置

(57) 摘要

本发明公开了一种获取应用代码的差异化覆盖率的方法和装置,涉及计算机技术领域。该方法的一具体实施方式包括:基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录被测代码的执行轨迹,执行轨迹由基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对被测代码进行编译以得到类文件;在类文件上运行结构分析器,以得到被测代码和基线代码的差异化代码;将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率。该实施方式提升了软件研发和测试的效率。



1. 一种获取应用代码的差异化覆盖率的方法,其特征在于,包括:

基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计;

基于版本控制算法,通过项目对象模型对所述被测代码进行编译以得到类文件;

在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码,所述基线代码是要与所述被测代码进行差异化比较的代码;

将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

2. 根据权利要求1所述的方法,其特征在于,基于基础镜像,根据被测代码进行应用部署包括:

对被测代码进行编译;

基于基础镜像构建应用的测试环境;

修改所述测试环境的启动参数以部署所述应用,所述启动参数包括部署所述应用的服务器的IP地址和端口号。

3. 根据权利要求2所述的方法,其特征在于,在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码包括:

被测代码基线代码使用所述结构分析器得到所述被测代码和基线代码的差异化类文件;

对所述类文件和所述差异化类文件进行比较分析以得到差异化代码。

4. 根据权利要求1所述的方法,其特征在于,基于版本控制算法,通过项目对象模型对所述被测代码进行编译包括:

利用JGit提供的版本控制算法,将所述被测代码复制到差异化覆盖率系统中;

通过所述差异化覆盖率系统包括的项目对象模型对所述被测代码进行编译。

5. 根据权利要求1所述的方法,其特征在于,记录所述被测代码的执行轨迹之后,还包括:

通过远程控制读写的方式,将所述执行轨迹存储至差异化覆盖率系统中。

6. 根据权利要求1所述的方法,其特征在于,所述差异化代码支持基于不同分支差异的差异化代码和基于同一分支的不同版本差异的差异化代码。

7. 根据权利要求6所述的方法,其特征在于,在获取应用代码的差异化覆盖率之后,还包括:

生成差异化覆盖率报告,且所述差异化覆盖率报告的标签包括所述被测代码的分支和版本号、所述基线代码的分支和版本号,以及应用标识。

8. 一种获取应用代码的差异化覆盖率的装置,其特征在于,包括:

测试执行模块,用于基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计;

代码编译模块,用于基于版本控制算法,通过项目对象模型对所述被测代码进行编译以得到类文件;

差异化分析模块,用于在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码,所述基线代码是要与所述被测代码进行差异化比较的代码;

覆盖率计算模块,用于将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

9. 一种获取应用代码的差异化覆盖率的电子设备,其特征在于,包括:

一个或多个处理器;

存储装置,用于存储一个或多个程序,

当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现如权利要求1-7中任一所述的方法。

10. 一种计算机可读介质,其上存储有计算机程序,其特征在于,所述程序被处理器执行时实现如权利要求1-7中任一所述的方法。

获取应用代码的差异化覆盖率的方法和装置

技术领域

[0001] 本发明涉及计算机技术领域,尤其涉及一种获取应用代码的差异化覆盖率的方法和装置。

背景技术

[0002] 在进行软件应用开发的过程中,软件测试是检测应用代码的重要手段,通过计算测试用例对应用代码的覆盖率可以实现对应用代码的检测。当应用功能发生变化时,应用代码也会发生变化。通过计算变化前后版本的应用代码的差异化覆盖率可以更为精准地评价应用代码。

[0003] 目前的差异化覆盖率系统多是将测试用例模块和待测的应用代码模块相关联,并动态维护测试用例模块和待测的应用代码模块的关联,通过获取待测的应用代码对应的目标用例,执行目标用例对应用代码进行测试以得到测试用例对应用代码的差异化覆盖率。

[0004] 在实现本发明过程中,发明人发现现有技术中至少存在如下问题:

[0005] 现有的获取应用代码的差异化覆盖率的方法,需要测试人员进行数据维护,代码研发和维护成本高,且对测试场景的要求较高,无法满足测试需求,测试效率低。

发明内容

[0006] 有鉴于此,本发明实施例提供一种获取应用代码的差异化覆盖率的方法和装置,无需对被测试代码进行添加或者修改配置,减少了测试人员对测试用例和代码的维护工作量,提升了软件研发和测试的效率。

[0007] 为实现上述目的,根据本发明实施例的一个方面,提供了一种获取应用代码的差异化覆盖率的方法。

[0008] 一种获取应用代码的差异化覆盖率的方法,包括:基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对所述被测代码进行编译以得到类文件;在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码,所述基线代码是要与所述被测代码进行差异化比较的代码;将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

[0009] 可选地,基于基础镜像,根据被测代码进行应用部署包括:对被测代码进行编译;基于基础镜像构建应用的测试环境;修改所述测试环境的启动参数以部署所述应用,所述启动参数包括部署所述应用的服务器的IP地址和端口号。

[0010] 可选地,在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码包括:使用所述结构分析器得到所述被测代码和基线代码的差异化类文件;对所述类文件和所述差异化类文件进行比较分析以得到差异化代码。

[0011] 可选地,基于版本控制算法,通过项目对象模型对所述被测代码进行编译包括:利

用JGit提供的版本控制算法,将所述被测代码复制到差异化覆盖率系统中;通过所述差异化覆盖率系统包括的项目对象模型对所述被测代码进行编译。

[0012] 可选地,记录所述被测代码的执行轨迹之后,还包括:通过远程控制读写的方式,将所述执行轨迹存储至差异化覆盖率系统中。

[0013] 可选地,所述差异化代码支持基于不同分支差异的差异化代码和基于同一分支的不同版本差异的差异化代码。

[0014] 可选地,在获取应用代码的差异化覆盖率之后,还包括:生成差异化覆盖率报告,且所述差异化覆盖率报告的标签包括所述被测代码的分支和版本号、所述基线代码的分支和版本号,以及应用标识。

[0015] 根据本发明实施例的另一方面,提供了一种获取应用代码的差异化覆盖率的装置。

[0016] 一种获取应用代码的差异化覆盖率的装置,包括:测试执行模块,用于基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计;代码编译模块,用于基于版本控制算法,通过项目对象模型对所述被测代码进行编译以得到类文件;差异化分析模块,用于在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码,所述基线代码是要与所述被测代码进行差异化比较的代码;覆盖率计算模块,用于将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

[0017] 根据本发明实施例的又一方面,提供了一种获取应用代码的差异化覆盖率的电子设备。

[0018] 一种获取应用代码的差异化覆盖率的电子设备,包括:一个或多个处理器;存储装置,用于存储一个或多个程序,当所述一个或多个程序被所述一个或多个处理器执行,使得所述一个或多个处理器实现本发明实施例所提供的获取应用代码的差异化覆盖率的方法。

[0019] 根据本发明实施例的再一方面,提供了一种计算机可读介质。

[0020] 一种计算机可读介质,其上存储有计算机程序,所述程序被处理器执行时实现本发明实施例所提供的获取应用代码的差异化覆盖率的方法。

[0021] 上述发明中的一个实施例具有如下优点或有益效果:通过基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录被测代码的执行轨迹,执行轨迹由基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对被测代码进行编译以得到类文件;在类文件上运行结构分析器,以得到被测代码和基线代码的差异化代码;将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率,可以实现对于不同版本源码的差异化代码进行差异化覆盖率分析,支持解析单源码、多源码的不同分支或同一分支不同版本号的差异化代码和获取对应的差异化覆盖率报告,无需对被测试代码进行添加或者修改配置,减少了测试人员对测试用例和代码的维护工作量,较大地提升了软件研发和测试的效率。通过根据自定义的路径规则,将每次生成的报告上传至统一存储平台中,通过请求统一存储平台的报告路径,来查看报告,实现了对报告的统一管理,便于查看和比较,进一步提高了软件研发和测试的效率。另外,根据本发明的技术方案,还可以实现一键部署测试环境,简化了测试操作,提高了测试效率。

[0022] 上述的非惯用的可选方式所具有的进一步效果将在下文中结合具体实施方式加以说明。

附图说明

[0023] 附图用于更好地理解本发明,不构成对本发明的不当限定。其中:

[0024] 图1是根据本发明实施例的获取应用代码的差异化覆盖率的方法的主要步骤示意图;

[0025] 图2是本发明实施例的获取差异化覆盖率的方法的实现流程示意图;

[0026] 图3是根据本发明实施例的获取应用代码的差异化覆盖率的装置的主要模块示意图;

[0027] 图4是本发明实施例可以应用于其中的示例性系统架构图;

[0028] 图5是适于用来实现本发明实施例的终端设备或服务器的计算机系统的结构示意图。

具体实施方式

[0029] 以下结合附图对本发明的示范性实施例做出说明,其中包括本发明实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本发明的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0030] 目前的差异化覆盖率系统,多是基于测试用例和应用代码的关联关系进行覆盖率测试的,既浪费了大量的人力和时间成本,又需要对代码逻辑进行研究,对开发人员的要求也高,故而开发和后期维护成本很高。

[0031] 而在进行差异化覆盖率测试时,可能会涉及对应用的不同分支的代码进行差异化覆盖率计算,也可能涉及对同一分支的不同版本的代码进行差异化覆盖率计算,因此测试场景和测试需求比较多样化,需要一种可以适应多样化测试场景的差异化覆盖率测试方法。

[0032] 为了解决上述的技术问题,本发明提供了一种获取应用代码的差异化覆盖率的方法,针对测试人员在web应用的手工测试和接口自动化测试,支持解析单源码、多源码的同一分支不同版本号的差异化代码和获取对应的差异化覆盖率报告,支持查看历史报告,不需要对待测试代码添加或者修改配置,不增加测试人员对用例和代码的维护工作量,将获取覆盖率报告和测试环境部署集合在一个平台上,实现一键部署测试环境,极大地提升软件研发和测试的效率。

[0033] 本发明的技术方案主要由3部分组成,依次是:测试执行、差异化覆盖率系统以及应用部署平台。差异化覆盖率系统主要用于一键部署测试环境,并进行差异化覆盖率分析,获取差异化代码覆盖率报告。

[0034] 一键部署测试环境是通过调用应用部署平台的编译、构建和部署API来实现某应用的某分支某版本代码(即:应用的被测代码)的测试环境部署,其前提是将覆盖率测试工具代理jacocoagent的代理监听作为基础镜像,覆盖率测试工具代理在应用启动后实时监听被测代码的执行情况。当应用应用部署平台上部署完成后,即可进行应用测试,并通过

基础镜像中的覆盖率测试工具代理统计分析被测代码的执行轨迹数据。

[0035] 在获取代码的差异化覆盖率时,差异化覆盖率系统利用JGit(JGit是一个轻量级纯Java的类库,用来实现Git的版本控制系统的访问,以及提供核心的版本控制算法)将远程代码平台的被测代码复制到差异化覆盖率系统中,通过差异化覆盖率系统自身带有的项目对象模型maven工具对被测代码进行编译,接着通过网络与应用部署平台上部署的应用建立连接,远程获取或下载被测代码的执行轨迹数据(exec格式的文件)到差异化覆盖率系统中,再在差异化覆盖率系统对被测代码编译后生成的类文件class文件上运行结构分析器,建立覆盖率模型,最后生成差异化代码对应的覆盖率报告。特别地,本发明的差异化覆盖率算法支持基于不同分支代码的差异化覆盖率、基于同一分支的不同版本代码的差异化覆盖率,代码差异化过滤了非java文件、空格、测试文件,仅识别java文件的差异化。

[0036] 为了方便测试人员查看历史报告,将每次生成后的差异化覆盖率报告上传至数据统一存储平台,报告标签tag以“分支名+新版本号+旧版本号+时间戳”来命名,有助于对比不同版本的测试覆盖率,帮助测试人员更全面地覆盖业务场景。根据本发明的实施例,分支名即为被测代码的分支名,新版本号即为被测代码的版本号,旧版本号即为基线代码的版本号,时间戳即为测试时间。

[0037] 图1是根据本发明实施例的获取应用代码的差异化覆盖率的方法的主要步骤示意图。如图1所示,本发明实施例的获取应用代码的差异化覆盖率的方法主要包括如下的步骤S101至步骤S104。

[0038] 步骤S101:基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录被测代码的执行轨迹,执行轨迹由基础镜像中包括的覆盖率测试工具代理进行监听统计;

[0039] 步骤S102:基于版本控制算法,通过项目对象模型对被测代码进行编译以得到类文件;

[0040] 步骤S103:在类文件上运行结构分析器,以得到被测代码和基线代码的差异化代码,基线代码是要与被测代码进行差异化比较的代码;

[0041] 步骤S104:将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率。

[0042] 根据本发明的一个实施例,在步骤S101中,基于基础镜像,根据被测代码进行应用部署时,具体可以按照以下的步骤来执行:

[0043] 对被测代码进行编译;

[0044] 基于基础镜像构建应用的测试环境;

[0045] 修改测试环境的启动参数以部署应用,启动参数包括部署应用的服务器的IP地址和端口号。

[0046] 差异化覆盖率系统获取分析覆盖率的前提,首先需要制作带有覆盖率测试工具代理jacocoagent的基础镜像;再将应用的被测代码进行编译、构建测试环境,其中在构建测试环境时需要结合之前已制作好的基础镜像来构建。基础镜像是启动一个应用所需要的环境,包含服务器操作系统及版本、tomcat(Web应用服务器)及版本、java及版本、Nginx(HTTP和反向代理web服务器)及版本等,本发明的实施例中,为了监听应用的被测代码执行过程,需要再加上覆盖率测试工具代理jacocoagent。

[0047] 之后,通过修改应用的测试环境的tomcat JVM(Java虚拟机)的启动参数以完成应用部署。具体地,各参数的设置情况如下:

[0048] set JAVA_OPTS="-javaagent:your localPath\jacocoagent.jar,includes=*,output=tcpserver,port=8080,address=IP";

[0049] 其中,your localPath是应用上存放jacocoagent.jar的地址,includes=*是指收集所有类,output=tcpserver是指在不停止应用服务的情况下获取被测代码的执行轨迹exec文件,address和port是部署应用的服务器的IP地址和端口号。

[0050] 通过差异化覆盖率系统的一键执行功能来将应用部署到应用部署平台上,以及在差异化覆盖率系统服务器上复制、编译被测代码。基线代码与被测代码可能是基于不同分支的代码,也可能是基于同一分支的不同版本的代码,基线代码是要与被测代码进行差异化比较的代码。应用启动后,通过终端命令查看服务器上是否存在jacocoagent.jar以确认基础镜像中是否包含覆盖率测试工具代理jacocoagent。同时,差异化覆盖率系统还可以获取被测代码编译后生成的类文件(class文件)。

[0051] 在应用部署完成之后,即可对应用部署平台上部署的应用连接进行代码测试,记录被测应用的代码执行轨迹,代码执行轨迹由覆盖率测试工具代理进行监听统计。

[0052] 在本发明的实施例中,差异化覆盖率系统支持的测试执行方式有两种,一种是手工执行(在应用的测试环境中,按照提前设计好的测试用例,进行功能测试),一种是接口自动化执行。

[0053] 其中,接口自动化的执行是将差异化覆盖率系统和测试接口自动化管理平台打通。通过调用用户在intest接口自动化管理平台维护的已有接口用例,对测试环境进行接口自动化测试。在执行测试的过程中,jacocoagent会监听前台测试行为“走”过了哪些后台代码,并记录成一种exec格式类型的文件,即代码执行轨迹。在记录被测应用的代码执行轨迹之后,还可以通过远程控制读写(RemoteControlReader)的方式,将代码执行轨迹存储至差异化覆盖率系统中。

[0054] 根据本发明的实施例,基于版本控制算法,通过项目对象模型对被测代码进行编译时,利用JGit提供的版本控制算法,将被测代码复制到差异化覆盖率系统中;然后,通过差异化覆盖率系统包括的项目对象模型对被测代码进行编译。具体地,差异化覆盖率系统利用JGit将远程部署的被测代码或者将分布式版本控制系统Git中的被测代码复制到覆盖率测试系统中,通过覆盖率测试系统自身的带有的maven工具对被测代码进行编译即可得到被测代码的类文件。

[0055] 根据本发明的技术方案,应用部署并进行测试以获取被测代码的执行轨迹的步骤,和对被测代码进行编译以得到类文件的步骤,两者并没有严格的执行顺序,可以同时执行,也可以分别先后执行,只要可以获取到被测代码的执行轨迹和类文件即可。

[0056] 然后,差异化覆盖率系统将进行差异化覆盖率计算。具体包括步骤S103在被测代码的类文件上运行结构分析器对被测代码和基线代码进行分析以得到差异化代码,和步骤S104将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率。

[0057] 在本发明的一个实施例中,使用结构分析器对被测代码和基线代码进行分析以得到差异化代码时,具体可以包括:使用结构分析器得到被测代码和基线代码的差异化类文件;对被测代码的类文件和差异化类文件进行比较分析以得到差异化代码。

[0058] 结构分析可以使开发者清楚软件系统的整体实现构架,减少在开发中恐慌与困惑。代码结构分析器用于对代码进行结构分析,具体地,是对被测代码和基线代码进行基于分支名和/或版本号的差异化分析,以得到被测代码和基线代码中的差异化类信息,如:java类文件名、类文件中的类、类中的方法信息(方法的md5值、方法名、方法入参等)、对比不同版号的代码增加的行数、删除的行数等等。代码编译时所生成的类文件中记录了代码中包括的类和每个类包括的方法,以及类之间的调用关系,例如:java类文件名、类文件中的类、类中的方法信息(方法的md5值、方法名、方法入参等)等等。通过对被测代码的类文件和差异化类文件进行比较分析,即可得到差异化代码。具体地,是在得到差异化类信息之后,将差异化类信息与被测代码的类信息进行比较,以在被测代码的类信息中映射出增加、删除或修改的类信息,进而得到差异化代码。

[0059] 然后,即可将测试执行时生成的被测代码的执行轨迹,以及差异化代码映射到被测代码中,以可视化到每一行代码的粒度,进而计算差异化覆盖率。在计算差异化覆盖率时,是通过代码覆盖率分析工具来进行计算的,其中,代码覆盖率分析工具例如是:jacoco 开源源码的分析函数。

[0060] 在获取应用代码的差异化覆盖率之后,还可以生成差异化覆盖率报告,且差异化覆盖率报告的标签包括被测代码的分支名和版本号、基线代码的分支名和版本号,以及应用标识(例如是应用名)和时间戳等。另外,由于差异化覆盖率系统的内存有限,故而根据一定的路径规则,可以将每次生成的报告上传至统一存储平台。前台查看报告只需要请求报告路径,即可查看报告。

[0061] 其中,覆盖率报告中可以包括行覆盖率、类覆盖率、方法覆盖率、分支覆盖率、指令覆盖率、圈复杂度等相关覆盖率。可以根据类名、方法名,获取一层一层的覆盖率,最后到达代码层。在代码层,会将被测代码较之基线代码中新增的代码,左侧使用“+”标记,将被测代码较之基线代码中删减的代码,左侧使用“-”标记。并且为代码行添加不同的颜色背景,以更明确地显示代码是否被执行。例如:红色背景代表本次测试执行,这一行中没有指令被执行,即没有覆盖;黄色背景代表这一行中只有一部分指令被执行,即部分执行;绿色背景代表这一行中所有指令都被覆盖,即全覆盖。下面将一一介绍这几种覆盖率的含义。

[0062] 1、行覆盖率:

[0063] 所有类文件均携带调试debug信息编译,则每行的覆盖率可计算。当至少一个指令被指定到源码行且已执行时,该源码行被认为已执行;

[0064] 2、类覆盖率:

[0065] 当类中至少有一个方法已执行,则该类被认为已执行。Jacoco中认为构造函数和静态初始化方法也当作被执行过的方法。Java接口类型若包含静态初始化方法,这种接口也被认为是可执行的类;

[0066] 3、方法覆盖率:

[0067] 每个非抽象方法至少包含一个指令。当至少一个指令被执行,该方法被认为已执行。由于Jacoco是基于字节码级别的,构造函数和静态初始化方法也被当作方法计算。其中有些方法,可能无法直接对应到源码中,比如默认构造器或常量的初始化命令;

[0068] 4、分支覆盖率:

[0069] 包括所有的if和switch语句。这个度量计算一个方法里面的总分支数,确定执行

和不执行的分支数量；

[0070] 5、指令覆盖率：

[0071] Jacoco最小的计数单元是单个java二进制代码指令。指令覆盖率提供了代码是否被执行的信息；

[0072] 6、圈复杂度：

[0073] Jacoco同样可以为每一个非抽象方法计算复杂度，最终计算出类、包和组的复杂度。根据由McCabe1996圈复杂度的定义是，在(线性)组合中，计算在一个方法里面所有可能路径的最小数目。所以复杂度可以作为度量单元测试是否有完全覆盖所有场景的一个依据。

[0074] 图2是本发明实施例的获取差异化覆盖率的方法的实现流程示意图。如图2所示，根据本发明的一个实施例，获取差异化覆盖率的方法主要包括如下的步骤：

[0075] 1、接收用户输入的被测代码和基线代码的信息。具体地，被测代码的信息例如包括被测代码的代码路径、所属项目名称、所属分支号以及版本号等信息；基线代码的信息包括基线代码的代码路径、所属项目名称、所属分支号以及版本号等信息；

[0076] 2、根据被测代码将应用部署到应用部署平台上，具体部署过程是依次执行以下步骤：代码编译、应用的测试环境构建、修改JVM启动参数以部署应用。其中，每个步骤在执行时，只有在其前一步骤执行成功的情况下才可执行，并且，每个步骤执行过程中均保存了执行日志，若执行失败，则可通过查询执行日志以查看具体的操作情况；

[0077] 3、基于版本控制算法，将被测代码复制到差异化覆盖率平台中；

[0078] 4、对被测代码进行编译以得到被测代码的类文件；

[0079] 5、对应用部署平台上已部署好的应用进行测试，并通过覆盖率测试工具代理jacocoagent对被测代码的执行轨迹进行监听和记录；

[0080] 6、在被测代码的类文件上运行结构分析器，以得到被测代码与基线代码的差异化代码；

[0081] 7、根据被测代码的执行轨迹和差异化代码计算差异化覆盖率；

[0082] 8、生成差异化覆盖率报告；

[0083] 9、将报告保存到统一存储平台，以及通过报告路径查看报告。

[0084] 根据本发明的技术方案，步骤5也可以是在步骤2之后，步骤3和步骤4之前执行。

[0085] 图3是根据本发明实施例的获取应用代码的差异化覆盖率的装置的主要模块示意图。如图3所示，本发明实施例的获取应用代码的差异化覆盖率的装置300主要包括测试执行模块301、代码编译模块302、差异化分析模块303和覆盖率计算模块304。

[0086] 测试执行模块301，用于基于基础镜像，根据被测代码进行应用部署，并对部署好的应用进行测试，记录所述被测代码的执行轨迹，所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计；

[0087] 代码编译模块302，用于基于版本控制算法，通过项目对象模型对所述被测代码进行编译以得到类文件；

[0088] 差异化分析模块303，用于用于在所述类文件上运行结构分析器，以得到所述被测代码和基线代码的差异化代码，所述基线代码是要与所述被测代码进行差异化比较的代码；

[0089] 覆盖率计算模块304,用于将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

[0090] 根据本发明的一个实施例,测试执行模块301还可以用于:

[0091] 对被测代码进行编译;

[0092] 基于基础镜像构建应用的测试环境;

[0093] 修改所述测试环境的启动参数以部署所述应用,所述启动参数包括部署所述应用的服务器的IP地址和端口号。

[0094] 根据本发明的另一个实施例,差异化分析模块303还可以用于:

[0095] 使用所述结构分析器得到所述被测代码和基线代码的差异化类文件;

[0096] 对所述类文件和所述差异化类文件进行比较分析以得到差异化代码。

[0097] 根据本发明的又一个实施例,代码编译模块302还可以用于:

[0098] 利用JGit提供的版本控制算法,将所述被测代码复制到差异化覆盖率系统中;

[0099] 通过所述差异化覆盖率系统包括的项目对象模型对所述被测代码进行编译。

[0100] 根据本发明的又一个实施例,获取应用代码的差异化覆盖率的装置300还可以包括轨迹存储模块(图中未示出),用于:

[0101] 在记录所述被测代码的执行轨迹之后,通过远程控制读写的方式,将所述执行轨迹存储至差异化覆盖率系统中。

[0102] 根据本发明的又一个实施例,所述差异化代码支持基于不同分支差异的差异化代码和基于同一分支的不同版本差异的差异化代码。

[0103] 根据本发明的再一个实施例,获取应用代码的差异化覆盖率的装置300还可以包括报告生成模块(图中未示出),用于:

[0104] 在获取应用代码的差异化覆盖率之后,生成差异化覆盖率报告,且所述差异化覆盖率报告的标签包括所述被测代码的分支和版本号、所述基线代码的分支和版本号,以及应用标识。

[0105] 根据本发明实施例的技术方案,通过基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录被测代码的执行轨迹,执行轨迹由基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对被测代码进行编译以得到类文件;在类文件上运行结构分析器,以得到被测代码和基线代码的差异化代码;将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率,可以实现对于不同版本源码的差异化代码进行差异化覆盖率分析,支持解析单源码、多源码的不同分支或同一分支不同版本号的差异化代码和获取对应的差异化覆盖率报告,无需对被测试代码进行添加或者修改配置,减少了测试人员对测试用例和代码的维护工作量,较大地提升了软件研发和测试的效率。通过根据自定义的路径规则,将每次生成的报告上传至统一存储平台中,通过请求统一存储平台的报告路径,来查看报告,实现了对报告的统一管理,便于查看和比较,进一步提高了软件研发和测试的效率。另外,根据本发明的技术方案,还可以实现一键部署测试环境,简化了测试操作,提高了测试效率。

[0106] 图4示出了可以应用本发明实施例的获取应用代码的差异化覆盖率的方法或获取应用代码的差异化覆盖率的装置的示例性系统架构400。

[0107] 如图4所示,系统架构400可以包括终端设备401、402、403,网络404和服务器405。

网络404用以在终端设备401、402、403和服务器405之间提供通信链路的介质。网络404可以包括各种连接类型,例如有线、无线通信链路或者光纤电缆等等。

[0108] 用户可以使用终端设备401、402、403通过网络404与服务器405交互,以接收或发送消息等。终端设备401、402、403上可以安装有各种客户端应用,例如网页浏览器应用、被测应用、交互类应用等(仅为示例)。

[0109] 终端设备401、402、403可以是具有显示屏并且支持网页浏览的各种电子设备,包括但不限于智能手机、平板电脑、膝上型便携计算机和台式计算机等等。

[0110] 服务器405可以是提供各种服务的服务器,例如对用户利用终端设备401、402、403所发送请求进行响应提供支持的后台管理服务器(仅为示例)。后台管理服务器可以对接收到的应用测试请求等数据进行分析等处理,并将处理结果反馈给终端设备。

[0111] 需要说明的是,本发明实施例所提供的获取应用代码的差异化覆盖率的方法一般由服务器405执行,相应地,获取应用代码的差异化覆盖率的装置一般设置于服务器405中。

[0112] 应该理解,图4中的终端设备、网络和服务器的数目仅仅是示意性的。根据实现需要,可以具有任意数目的终端设备、网络和服务器。

[0113] 下面参考图5,其示出了适于用来实现本发明实施例的终端设备或服务器的计算机系统500的结构示意图。图5示出的终端设备或服务器仅仅是一个示例,不应对本发明实施例的功能和使用范围带来任何限制。

[0114] 如图5所示,计算机系统500包括中央处理单元(CPU)501,其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM 503中,还存储有系统500操作所需的各种程序和数据。CPU 501、ROM 502以及RAM 503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0115] 以下部件连接至I/O接口505:包括键盘、鼠标等的输入部分506;包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分507;包括硬盘等的存储部分508;以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分509。通信部分509经由诸如因特网的网络执行通信处理。驱动器510也根据需要连接至I/O接口505。可拆卸介质511,诸如磁盘、光盘、磁光盘、半导体存储器等等,根据需要安装在驱动器510上,以便于从其上读出的计算机程序根据需要被安装入存储部分508。

[0116] 特别地,根据本发明公开的实施例,上文参考流程图描述的过程可以被实现为计算机软件程序。例如,本发明公开的实施例包括一种计算机程序产品,其包括承载在计算机可读介质上的计算机程序,该计算机程序包含用于执行流程图所示的方法的程序代码。在这样的实施例中,该计算机程序可以通过通信部分509从网络上被下载和安装,和/或从可拆卸介质511被安装。在该计算机程序被中央处理单元(CPU)501执行时,执行本发明的系统中限定的上述功能。

[0117] 需要说明的是,本发明所示的计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质或者是上述两者的任意组合。计算机可读存储介质例如可以是一——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件,或者任意以上的组合。计算机可读存储介质的更具体的例子可以包括但不限于:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机访问存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储

器 (EPROM或闪存)、光纤、便携式紧凑磁盘只读存储器 (CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本发明中,计算机可读存储介质可以是任何包含或存储程序的有形介质,该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。而在本发明中,计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号,其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式,包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质,该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。计算机可读介质上包含的程序代码可以用任何适当的介质传输,包括但不限于:无线、电线、光缆、RF等等,或者上述的任意合适的组合。

[0118] 附图中的流程图和框图,图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分,上述模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意,在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个接连地表示的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意,框图或流程图中的每个方框、以及框图或流程图中的方框的组合,可以用执行规定的功能或操作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0119] 描述于本发明实施例中所涉及到的单元或模块可以通过软件的方式实现,也可以通过硬件的方式来实现。所描述的单元或模块也可以设置在处理器中,例如,可以描述为:一种处理器包括测试执行模块、代码编译模块、差异化分析模块和覆盖率计算模块。其中,这些单元或模块的名称在某种情况下并不构成对该单元或模块本身的限定,例如,测试执行模块还可以被描述为“用于基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计的模块”。

[0120] 作为另一方面,本发明还提供了一种计算机可读介质,该计算机可读介质可以是上述实施例中描述的设备中所包含的;也可以是单独存在,而未装配入该设备中。上述计算机可读介质承载有一个或者多个程序,当上述一个或者多个程序被一个该设备执行时,使得该设备包括:基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录所述被测代码的执行轨迹,所述执行轨迹由所述基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对所述被测代码进行编译以得到类文件;在所述类文件上运行结构分析器,以得到所述被测代码和基线代码的差异化代码,所述基线代码是要与所述被测代码进行差异化比较的代码;将所述差异化代码和所述执行轨迹映射到所述被测代码中,以获取应用代码的差异化覆盖率。

[0121] 根据本发明实施例的技术方案,通过基于基础镜像,根据被测代码进行应用部署,并对部署好的应用进行测试,记录被测代码的执行轨迹,执行轨迹由基础镜像中包括的覆盖率测试工具代理进行监听统计;基于版本控制算法,通过项目对象模型对被测代码进行编译以得到类文件;在类文件上运行结构分析器,以得到被测代码和基线代码的差异化代

码;将差异化代码和执行轨迹映射到被测代码中,以获取应用代码的差异化覆盖率,可以实现对于不同版本源码的差异化代码进行差异化覆盖率分析,支持解析单源码、多源码的不同分支或同一分支不同版本号的差异化代码和获取对应的差异化覆盖率报告,无需对被测试代码进行添加或者修改配置,减少了测试人员对测试用例和代码的维护工作量,较大地提升了软件研发和测试的效率。通过根据自定义的路径规则,将每次生成的报告上传至统一存储平台中,通过请求统一存储平台的报告路径,来查看报告,实现了对报告的统一管理,便于查看和比较,进一步提高了软件研发和测试的效率。另外,根据本发明的技术方案,还可以实现一键部署测试环境,简化了测试操作,提高了测试效率。

[0122] 上述具体实施方式,并不构成对本发明保护范围的限制。本领域技术人员应该明白的是,取决于设计要求和因素,可以发生各种各样的修改、组合、子组合和替代。任何在本发明的精神和原则之内所作的修改、等同替换和改进等,均应包含在本发明保护范围之内。

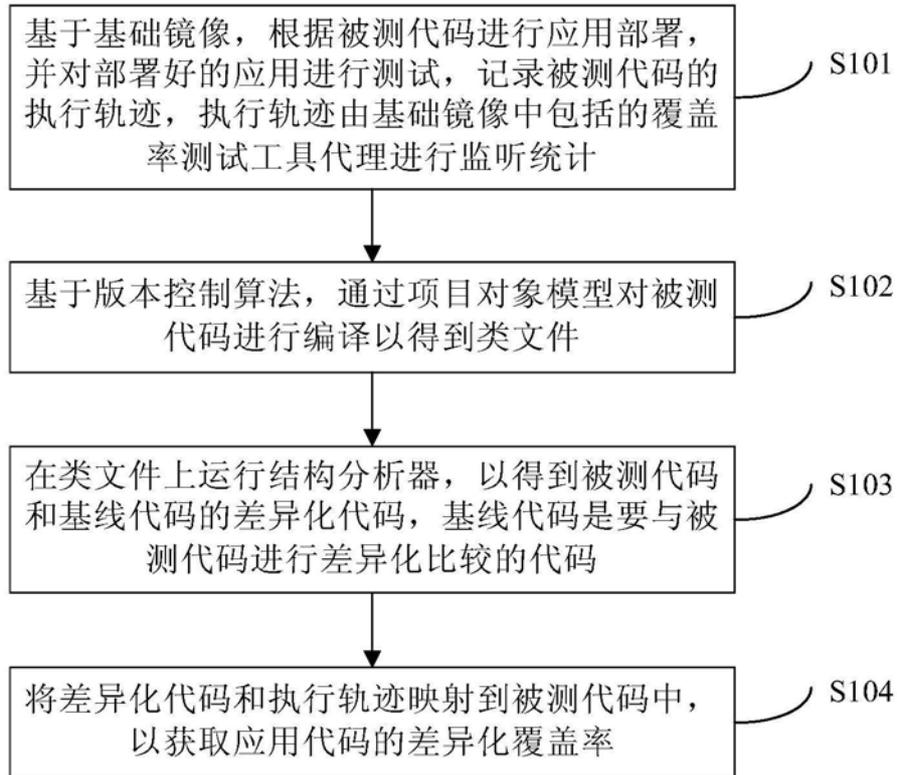


图1

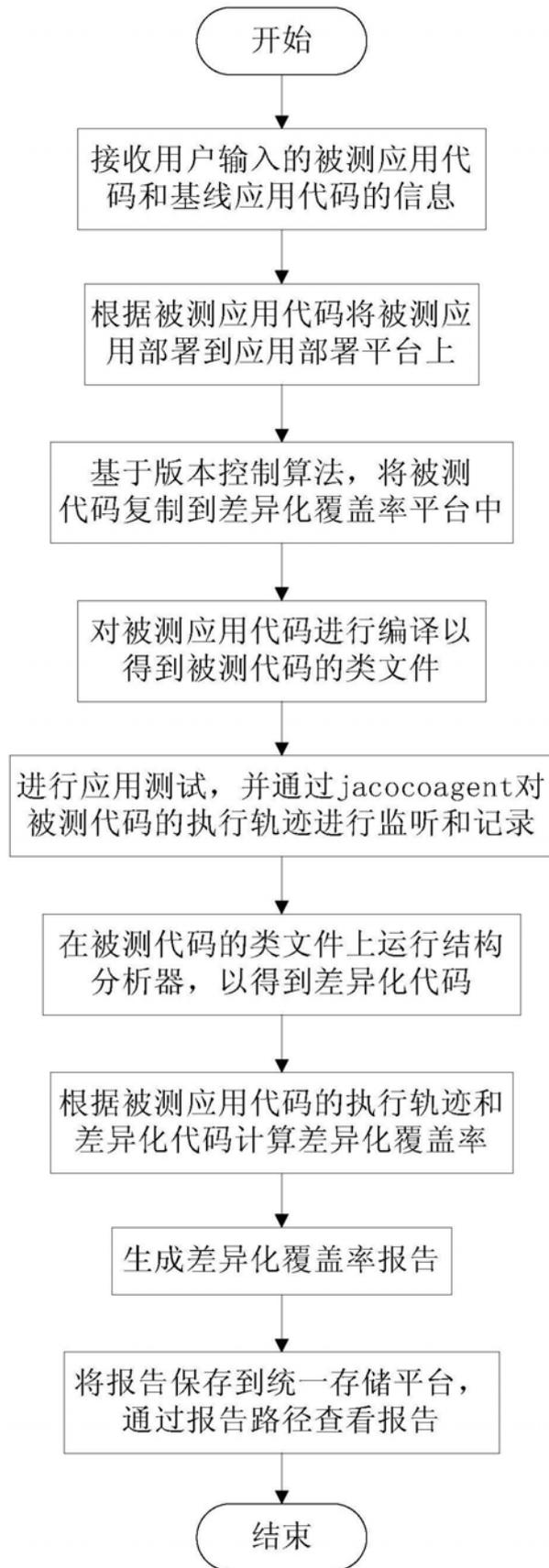


图2

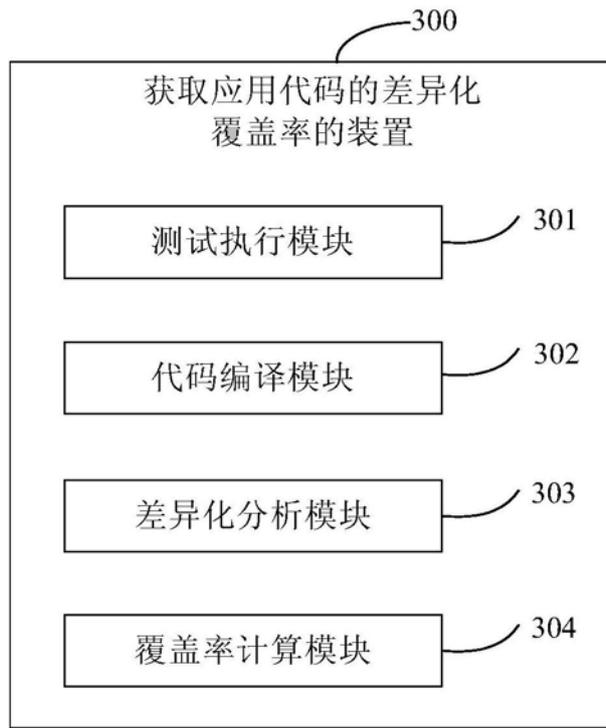


图3

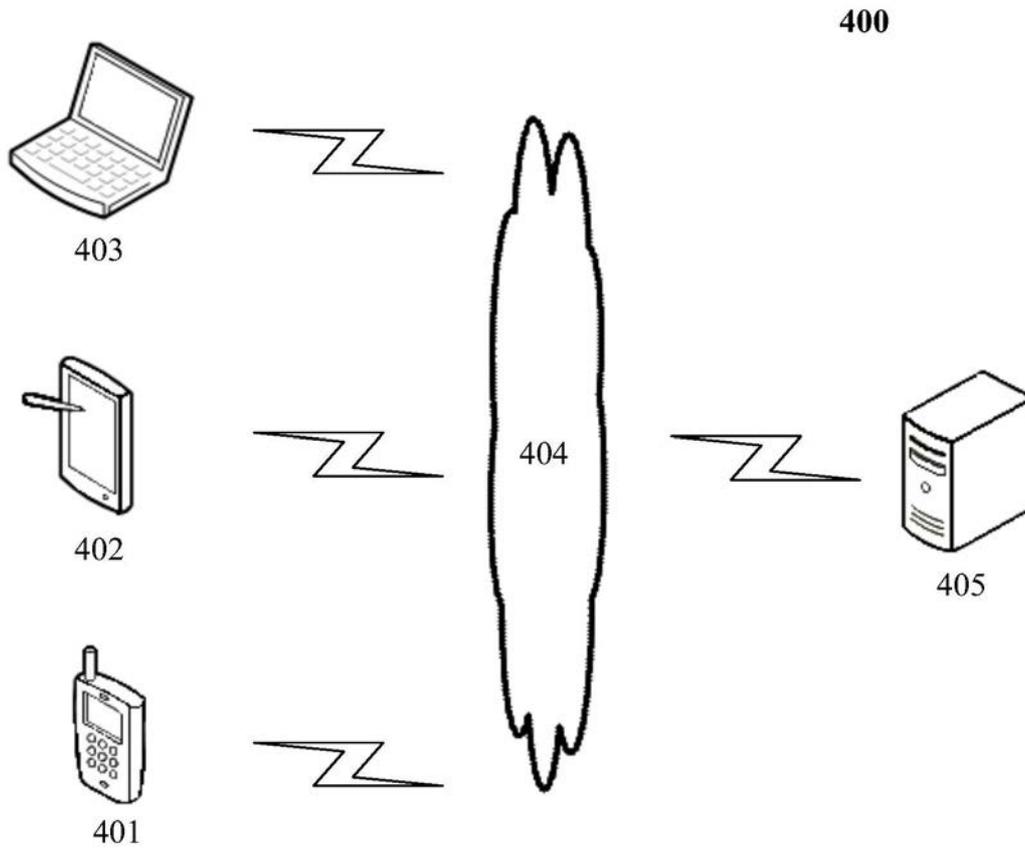


图4

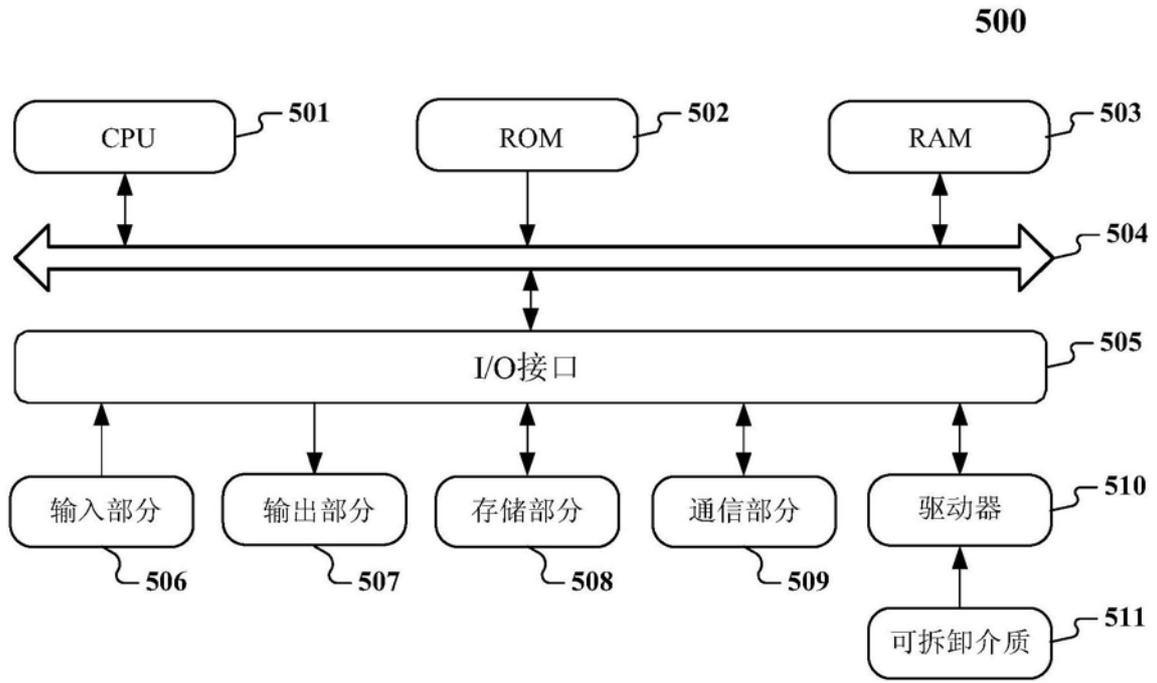


图5