



# (12) 发明专利申请

(10) 申请公布号 CN 114968776 A

(43) 申请公布日 2022. 08. 30

(21) 申请号 202210544110.9

(22) 申请日 2022.05.19

(71) 申请人 京东科技控股股份有限公司  
地址 100176 北京市北京经济技术开发区  
科创十一街18号C座2层221室

(72) 发明人 熊俊濂

(74) 专利代理机构 中原信达知识产权代理有限  
责任公司 11219  
专利代理师 李雪楠 王志远

(51) Int. Cl.

G06F 11/36 (2006.01)

G06F 9/50 (2006.01)

G06F 9/52 (2006.01)

G06F 11/30 (2006.01)

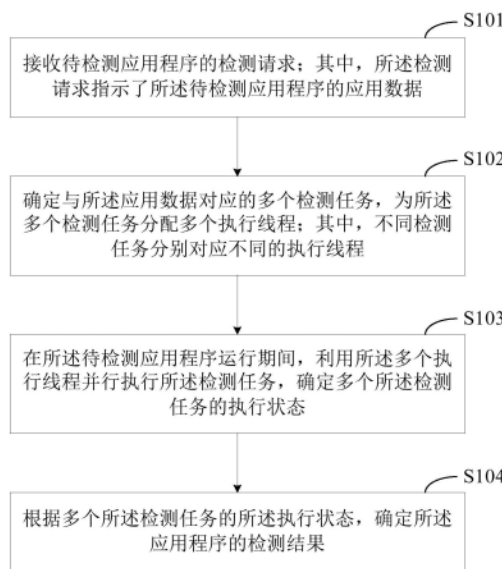
权利要求书2页 说明书11页 附图5页

## (54) 发明名称

一种应用程序的检测方法和装置

## (57) 摘要

本发明公开了一种应用程序的检测方法和装置,涉及信息安全技术领域。该方法的具体实施方式包括:接收待检测应用程序的检测请求;确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。该实施方式能够减少应用程序的检测时间,提高APP的检测效率,保护检测设备,降低检测过程对检测设备的损耗,优化检测结果的判定过程,提高检测成功率,降低控制成本和检测成本。



1. 一种应用程序的检测方法,其特征在于,包括:

接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据;

确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;

在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;

根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

2. 根据权利要求1所述的方法,其特征在于,所述确定多个所述检测任务的执行状态,包括:

分别为每一个所述执行线程分配任务状态容器;

利用所述任务状态容器对所述执行线程进行监听,实时更新多个所述检测任务的执行状态;其中,所述执行状态包括执行前、执行中、执行完成。

3. 根据权利要求2所述的方法,其特征在于,所述根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果,包括:

创建状态容器监听线程;

利用所述状态容器监听线程对多个所述任务状态容器进行监听,确定多个所述任务状态容器对应的所述检测任务的执行状态;

在确定与所述待检测应用程序对应的多个所述检测任务的执行状态全部为执行完成的情况下,确定所述应用程序的检测结果。

4. 根据权利要求2所述的方法,其特征在于,所述执行完成的执行状态指示了执行结果,所述执行结果包括执行成功和执行失败,所述确定所述应用程序的检测结果,包括:

在与所述待检测应用程序对应的多个所述检测任务的执行结果全部为执行成功的情况下,确定所述应用程序的检测结果为检测成功;

或者,

在任意一个所述检测任务的执行结果为执行失败的情况下,确定所述应用程序的检测结果为检测失败。

5. 根据权利要求2所述的方法,其特征在于,所述利用所述任务状态容器对所述执行线程进行监听,实时更新多个所述检测任务的执行状态,包括:

利用所述任务状态容器监听所述执行线程执行所述检测任务的执行结果;

确定所述执行结果是否为执行成功,如果是,存储所述检测任务的执行状态为执行完成。

6. 根据权利要求5所述的方法,其特征在于,在所述执行结果为执行失败的情况下,还包括:

A1: 重试所述执行线程;

A2: 确定所述执行线程的重试结果是否为执行成功,如果是,存储所述检测任务的执行状态为执行完成,对应的执行结果为执行成功;如果否,递增重试次数;

A3: 确定所述重试次数是否超过预设重试次数阈值,如果是,存储所述检测任务的执行状态为执行完成,对应的执行结果为执行失败;如果否,转至步骤A1。

7. 根据权利要求1至6中任一项所述的方法,其特征在于,所述应用数据包括静态检测条件参数、自启动行为检测条件参数和动态检测条件参数;

所述检测任务包括静态检测任务、自启动行为检测任务和动态检测任务。

8. 一种应用程序的检测装置,其特征在于,包括:

接收模块,用于接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据;

分配模块,用于确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;

执行模块,用于在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;

结果模块,用于根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

9. 一种应用程序的检测的电子设备,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。

## 一种应用程序的检测方法和装置

### 技术领域

[0001] 本发明涉及信息安全技术领域,尤其涉及一种应用程序的检测方法和装置。

### 背景技术

[0002] APP(Application,即应用程序)通常安装于各种智能终端,用于为用户提供搜索、通信、社交、消费等各种服务。现有的APP在上线之前,需要进行隐私合规检测以规避风险,检测步骤包括静态行为检测、自启动行为检测和动态行为检测。

[0003] 现有的APP隐私合规检测过程中,各个步骤需按序进行,确定前一个步骤执行完毕后再进行下一步骤,并且,每执行一个步骤需重启一次检测设备,根据各个步骤的检测结果确定APP是否可以上线。

[0004] 因此,现有的APP合规检测中,下一步骤依赖前一步骤的执行结果,导致检测耗时过长;反复的关闭、重启会影响检测设备的使用寿命,对应检测过程的控制成本增加;任一步骤检测失败都会导致APP无法上线,浪费成功步骤的检测成本。

### 发明内容

[0005] 有鉴于此,本发明实施例提供一种应用程序的检测方法和装置,能够减少应用程序的检测时间,提高APP的检测效率,保护检测设备,降低检测过程对检测设备的损耗,优化检测结果的判定过程,提高检测成功率,降低控制成本和检测成本。

[0006] 为实现上述目的,根据本发明实施例的一个方面,提供了一种应用程序的检测方法,包括:

[0007] 接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据;

[0008] 确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;

[0009] 在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;

[0010] 根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

[0011] 可选地,所述确定多个所述检测任务的执行状态,包括:

[0012] 分别为每一个所述执行线程分配任务状态容器;

[0013] 利用所述任务状态容器对所述执行线程进行监听,实时更新多个所述检测任务的执行状态;其中,所述执行状态包括执行前、执行中、执行完成。

[0014] 可选地,所述根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果,包括:

[0015] 创建状态容器监听线程;

[0016] 利用所述状态容器监听线程对多个所述任务状态容器进行监听,确定多个所述任务状态容器对应的所述检测任务的执行状态;

[0017] 在确定与所述待检测应用程序对应的多个所述检测任务的执行状态全部为执行完成的情况下,确定所述应用程序的检测结果。

[0018] 可选地,所述执行完成的执行状态指示了执行结果,所述执行结果包括执行成功和执行失败,所述确定所述应用程序的检测结果,包括:

[0019] 在与所述待检测应用程序对应的多个所述检测任务的执行结果全部为执行成功的情况下,确定所述应用程序的检测结果为检测成功;

[0020] 或者,

[0021] 在任意一个所述检测任务的执行结果为执行失败的情况下,确定所述应用程序的检测结果为检测失败。

[0022] 可选地,所述利用所述任务状态容器对所述执行线程进行监听,实时更新多个所述检测任务的执行状态,包括:

[0023] 利用所述任务状态容器监听所述执行线程执行所述检测任务的执行结果;

[0024] 确定所述执行结果是否为执行成功,如果是,存储所述检测任务的执行状态为执行完成。

[0025] 可选地,在所述执行结果为执行失败的情况下,还包括:

[0026] A1:重试所述执行线程;

[0027] A2:确定所述执行线程的重试结果是否为执行成功,如果是,存储所述检测任务的执行状态为执行完成,对应的执行结果为执行成功;如果否,递增重试次数;

[0028] A3:确定所述重试次数是否超过预设重试次数阈值,如果是,存储所述检测任务的执行状态为执行完成,对应的执行结果为执行失败;如果否,转至步骤A1。

[0029] 可选地,所述应用数据包括静态检测条件参数、自启动行为检测条件参数和动态检测条件参数;

[0030] 所述检测任务包括静态检测任务、自启动行为检测任务和动态检测任务。

[0031] 根据本发明实施例的再一个方面,提供了一种应用程序的检测装置,包括:

[0032] 接收模块,用于接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据;

[0033] 分配模块,用于确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;

[0034] 执行模块,用于在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;

[0035] 结果模块,用于根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

[0036] 根据本发明实施例的另一个方面,提供了一种应用程序的检测的电子设备,包括:

[0037] 一个或多个处理器;

[0038] 存储装置,用于存储一个或多个程序,

[0039] 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提供的应用程序的检测方法。

[0040] 根据本发明实施例的还一个方面,提供了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提供的应用程序的检测方法。

[0041] 上述发明中的一个实施例具有如下优点或有益效果:因为采用为不同的检测任务分配不同的执行线程,进而并行执行各个检测任务;利用任务状态容器监测各个检测任务的执行状态,在执行失败的情况下重试执行线程以执行检测任务;使用状态容器监听线程监听各个任务状态容器的执行状态并汇总执行结果的技术手段,所以克服了现有的APP合规检测耗时过长、反复的启闭影响检测设备的使用寿命、检测过程的控制成本高、浪费检测成本的技术问题,进而达到能够减少应用程序的检测时间,提高APP的检测效率,保护检测设备,降低检测过程对检测设备的损耗,优化检测结果的判定过程,提高检测成功率,降低控制成本和检测成本的技术效果。

[0042] 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

[0043] 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:

[0044] 图1是根据本发明实施例的应用程序的检测方法的主要流程的示意图;

[0045] 图2是根据本发明实施例的执行线程的执行状态的确定方法的主要流程的示意图;

[0046] 图3是根据本发明实施例的检测结果的确定方法的主要流程的示意图;

[0047] 图4是根据本发明实施例的应用程序的检测装置的主要模块的示意图;

[0048] 图5示出了适于应用于本发明实施例的应用程序的检测方法或应用程序的检测装置的示例性系统架构图;

[0049] 图6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

[0050] 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。

[0051] APK:Android application package,是指Android操作系统使用的一种应用程序包文件格式,用于分发和安装移动应用及中间件。

[0052] Xposed框架:Xposed Framework,是一套开源的、在Android操作系统高权限模式下运行的框架服务,可以在不修改APK文件的情况下影响程序运行/修改系统,支持创建其它功能模块,且在功能不冲突的情况下同时运作。

[0053] Hook:Windows消息处理机制的一个平台,应用程序可以在上面设置子程以监视指定窗口的消息,当消息到达后,在目标窗口处理函数之前处理它,也即,钩子机制允许应用程序截获处理window消息或特定事件。也称钩子函数,在系统的特定消息未到达目标窗口之前,钩子程序优先捕获该消息,即钩子函数先得到控制权,可以修改/改变目标窗口的执行行为、强制结束消息的传递等。

[0054] 图1是根据本发明实施例的应用程序的检测方法的主要流程的示意图,如图1所

示,本发明的应用程序的检测方法包括如下步骤:

[0055] 各大平台的移动应用(即APP)上线发版之前,通常需要进行隐私合规审核,常规审核措施是利用隐私合规检测工具(隐私合规检测工具可以是APP扫描检测系统,包括APP解压反编译及规则匹配等功能,能够检测APP源码及动态运行中产生的数据)对即将发布的APP进行检测,根据检测结果及时发现潜在的隐私合规风险,并在上线前对其进行规避。

[0056] 整个检测流程的各个检测任务依次/按顺序执行,彼此之间串行依赖,后一个检测任务的开始需要等待前一个检测任务的结束,将前一个任务的检测结果做为下一个任务开始执行的依据条件;并且,每次执行检测任务都需要重启检测设备(检测设备为检测终端,比如,手机、平板电脑等为满足检测需要而定制操作系统、网络中间代理等的终端),同时对检测设备虚拟环境进行初始化设置。

[0057] 然而,一方面,由于各个检测任务之间的串行关系,使得各个检测任务之间存在等待现象,使得检测过程耗时过长;另一方面,每一个检测任务都需要重启检测设备并初始化,使得检测过程的控制成本增加,并且频繁的关启操作会损耗检测设备的使用寿命、降低其性能;再一方面,APP的检测结果为各个检测任务的结果汇总,任意一个结果异常都会导致整个检测流程的失败,使得检测成功的检测任务失效,浪费检测成本,检测成功率低下。

[0058] 步骤S101,接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据。

[0059] 在本发明实施例中,一个Android应用程序的代码需要编译为能够被Android系统识别的文件,方可在Android设备上运行,能被Android系统识别的文件格式即为“APK”。本发明的应用程序的检测方法可以由检测服务器执行,检测服务器接收终端发送的待检测应用程序的检测请求。应用数据可以是APP的APK基础信息,包括APP名称、APP的package\_name(即APK包名,APP的唯一标识)、APK文件hash(APK文件的检测标识,具有唯一性,出现相同hash值时表明该APK已经有过扫描记录)、APK文件大小、加密类型方式、网络环境参数、自启动监听参数、检测设备的硬件参数等。

[0060] 在本发明实施例中,检测请求还指示了检测设备,检测设备上安装有待检测应用程序;其中,检测设备提供由Xposed框架搭建的服务环境,用于运行待检测应用程序对应的APK,待检测应用程序对应的APK设置检测接口,使得检测服务器可以调用检测接口以使得待检测应用程序执行检测任务。

[0061] 进一步地,检测服务器可以通过APK的检测接口向待检测应用程序发送应用数据,使得待检测应用程序从应用数据获取对应的检测条件参数以执行检测任务。

[0062] 步骤S102,确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程。

[0063] 在本发明实施例中,每一个待检测应用程序对应的多个检测任务包括静态检测任务、自启动行为检测任务和动态检测任务。检测服务器根据应用数据,确定与应用数据对应的检测任务,应用数据包括静态检测条件参数、自启动行为检测条件参数和动态检测条件参数,相应地,静态检测条件参数对应静态检测任务、自启动行为检测条件参数对应自启动行为检测任务、动态检测条件参数对应动态检测任务。

[0064] 静态检测任务是指在检测过程中,利用反编译工具对APP的APK包文件进行静态扫描分析,获取应用基本信息、敏感权限静态调用信息、第三方集成SDK信息等静态行为。其

中,敏感权限静态调用信息是指APP可以通过操作系统提供的权限API获取用户的相关隐私信息,包括但不限于通讯录、短信、相册、地理位置、登录凭证、银行卡号、就医记录、行程等隐私信息;第三方集成SDK信息是指辅助APP开发时用到的具有某些特定功能的软件开发工具包的集合,通常取自开源工具。

[0065] 自启动行为检测任务是指在检测过程中,利用安卓系统Hook程序和检测设备对APP的自启动行为进行检测,获取APP的自启动行为等自启动行为。

[0066] 动态检测任务是指在检测过程中,利用虚拟环境对APP运行时的动态进行监测,获取APP的系统权限调用、网络流量数据等动态行为。其中,系统权限调用是指APP通过操作系统调用系统权限API的相关权限信息,包括但不限于通讯录、短信、相册、地理位置等权限信息;网络流量数据是指APP运行时产生的网络请求与响应数据,比如,网络请求是否是跨境访问、响应数据是否存在IMEI识别码等敏感数据等等。

[0067] 在本发明实施例中,确定检测任务后,检测服务器可以向终端下达启动指令,使得终端的工作人员启动检测设备;或者,检测设备设置启闭接口,确定检测任务后,检测服务器调用启闭接口启动检测设备,以使得检测设备的待检测应用程序利用执行线程执行检测任务。

[0068] 在本发明实施例中,检测服务器可以通过APK的检测接口为多个检测任务分配多个执行线程,使得待检测应用程序的检测任务可以通过多个执行线程运行,不同的检测任务分别对应不同的执行线程,比如,多个执行线程包括静态执行线程、自启动执行线程和动态执行线程;其中,静态执行线程执行静态检测任务、自启动执行线程执行自启动行为检测任务、动态执行线程执行动态检测任务。

[0069] 步骤S103,在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态。

[0070] 在本发明实施例中,检测服务器用多个执行线程并行执行检测任务,相当于每一个执行线程运行一个待检测应用程序,并且多个执行线程同时运行。

[0071] 在本发明实施例中,每一个待检测应用程序执行一个检测任务,比如,静态执行线程的待检测应用程序执行静态检测任务、自启动执行线程的待检测应用程序执行自启动行为检测任务、动态执行线程的待检测应用程序执行动态检测任务。

[0072] 进一步地,检测服务器将应用数据通过APK的检测接口发送给各个执行线程,使得各个执行线程的待检测应用程序从应用数据获取对应的检测条件参数,比如,静态执行线程的待检测应用程序从应用数据获取对应的静态检测条件参数、自启动执行线程的待检测应用程序从应用数据获取对应的自启动行为检测条件参数、动态执行线程的待检测应用程序从应用数据获取对应的动态检测条件参数。

[0073] 更进一步地,检测服务器启动各个执行线程,使得各个执行线程的待检测应用程序根据对应的检测条件参数执行对应的检测任务,比如,静态执行线程的待检测应用程序根据静态检测条件参数执行静态检测任务、自启动执行线程的待检测应用程序根据自启动行为检测条件参数执行自启动行为检测任务、动态执行线程的待检测应用程序根据动态检测条件参数执行动态检测任务。

[0074] 在本发明实施例中,检测任务的执行状态包括执行前、执行中、执行完成。其中,执行前表示执行线程处于检测条件参数的获取阶段,执行线程还未开始执行检测任务;执行



中表示执行线程正在执行检测任务的检测过程中;执行完成表示执行线程已执行完毕检测任务。

[0075] 在本发明实施例中,或者,在启动检测设备之前,检测服务器控制静态执行线程的待检测应用程序从应用数据获取对应的静态检测条件参数,并根据静态检测条件参数执行静态检测任务,对非运行状态下的APP进行源码扫描。

[0076] 在启动检测设备之后,检测服务器控制自启动执行线程的待检测应用程序从应用数据获取对应的自启动检测条件参数,并根据自启动检测条件参数执行自启动行为检测任务,并且,检测服务器控制动态执行线程的待检测应用程序从应用数据获取对应的动态检测条件参数,并根据动态检测条件参数执行动态检测任务,对运行状态的APP进行捕获劫持,并对动态行为进行监听分析。

[0077] 在本发明实施例中,如图2所示,本发明的执行线程的执行状态的确定方法包括如下步骤:

[0078] 步骤S201,分别为每一个执行线程分配任务状态容器。

[0079] 在本发明实施例中,任务状态容器用于监听并存储执行线程的执行状态,不同的执行线程对应不同的任务状态容器。

[0080] 进一步地,检测服务器也可以设置一个任务状态容器,用于监听全部执行线程的执行状态。

[0081] 步骤S202,利用任务状态容器对执行线程进行监听,实时更新多个检测任务的执行状态。

[0082] 在本发明实施例中,也可以单独为各个执行线程设置监听程序用于判断执行线程的执行状态,并利用任务状态容器存储执行线程的执行状态。

[0083] 进一步地,各个任务状态容器根据自身对执行线程的监听结果,分别实时更新各个检测任务的执行状态;或者,各个任务状态容器根据单独监听程序对执行线程的监听结果,分别实时更新各个检测任务的执行状态。

[0084] 在本发明实施例中,在各个任务状态容器确定对应的检测任务的执行状态为执行完成的的情况下,可以存储各个检测任务对应的检测过程的检测数据,以便于开发人员后续数据分析、问题定位及修正以及应用维护等。

[0085] 步骤S2021,利用任务状态容器监听执行线程执行检测任务的执行结果。

[0086] 在本发明实施例中,执行结果包括执行成功和执行失败。

[0087] 步骤S2022,确定执行结果是否为执行成功,如果是,转至步骤S2023;如果否,转至步骤S2024。

[0088] 步骤S2023,存储检测任务的执行状态为执行完成。

[0089] 在本发明实施例中,在执行线程的执行结果为执行成功、或者执行线程的重试结果为执行成功的情况下,表示执行线程对应的检测任务执行完成,并且,执行完成的执行状态对应的执行结果为执行成功。

[0090] 在本发明实施例中,在执行线程的重试结果为执行失败并且重试次数超过预设重试次数阈值的情况下,表示执行线程对应的检测任务执行完成,并且,执行完成的执行状态对应的执行结果为执行失败。

[0091] 步骤S2024,重试执行线程。

[0092] 在本发明实施例中,在执行线程的执行结果为执行失败、或者执行线程的重试结果为执行失败的情况下,重试利用执行线程再次执行对应的检测任务。

[0093] 步骤S2025,确定执行线程的重试结果是否为执行成功,如果是,转至步骤S2023; 如果否,转至步骤S2026。

[0094] 步骤S2026,递增重试次数。

[0095] 步骤S2027,确定重试次数是否超过预设重试次数阈值,如果是,转至步骤S2023; 如果否,转至步骤S2024。

[0096] 在本发明实施例中,通过本发明的执行线程的执行状态的确定方法,能够通过任务状态容器对各个执行线程进行实时监听,从而监测并实时更新执行线程的执行状态,便于后续检测结果的确,由于检测任务并行执行,而且对执行失败的检测任务进行多次重试,可以大大提高检测速度和检测任务的成功率,提升了检测效率。

[0097] 步骤S104,根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

[0098] 在本发明实施例中,应用程序的检测结果包括检测成功和检测失败。在与待检测应用程序对应的多个检测任务的执行结果全部为执行成功的情况下,确定应用程序的检测结果为检测成功;

[0099] 或者,在任意一个检测任务的执行结果为执行失败的情况下,确定应用程序的检测结果为检测失败

[0100] 在本发明实施例中,如图3所示,本发明的检测结果的确定方法包括如下步骤:

[0101] 步骤S301,创建状态容器监听线程。

[0102] 在本发明实施例中,状态容器监听线程用于对多个检测任务对应的多个任务状态容器进行监听,从而可以实时获取执行线程的执行状态。

[0103] 步骤S302,利用状态容器监听线程对多个任务状态容器进行监听,确定多个任务状态容器对应的检测任务的执行状态。

[0104] 步骤S303,在确定与待检测应用程序对应的多个检测任务的执行状态全部为执行完成的情况下,确定应用程序的检测结果。

[0105] 在本发明实施例中,执行完成的执行状态指示了执行结果。

[0106] 在本发明实施例中,调用状态容器监听线程,判断各个检测任务是否执行完成,只有在全部检测任务均为执行完成的情况下,才会确定待检测应用程序的检测结果,否则,状态容器监听线程将保持监听状态,监听执行状态为执行前或者执行中的各个执行线程。

[0107] 步骤S3031,在与待检测应用程序对应的多个检测任务的执行结果全部为执行成功的情况下,确定应用程序的检测结果为检测成功。

[0108] 在本发明实施例中,检测服务器根据状态容器监听线程汇总的各个检测任务的执行结果,判断各个检测任务是否执行成功,只有在全部检测任务的执行结果均为执行成功的情况下,才会确定待检测应用程序的检测结果为检测成功。

[0109] 步骤S3032,在任意一个检测任务的执行结果为执行失败的情况下,确定应用程序的检测结果为检测失败。

[0110] 在本发明实施例中,检测服务器在任一检测任务的执行结果为执行失败时,即确定待检测应用程序的检测结果为检测失败。

[0111] 在本发明实施例中,通过本发明的检测结果的确定方法,能够利用状态容器监听线程对多个任务状态容器进行监听,从而根据对应的检测任务的执行状态确定应用程序的检测结果,可以大大提高检测速度,提升检测效率。

[0112] 在本发明实施例中,进一步地,还可以根据检测结果,输出检测报告,将检测报告发送给所述终端以响应所述检测请求。各个检测任务的执行状态由检测服务器统一汇总,任何一个检测任务的执行状态更新为执行完成时,都会向状态容器监听线程发送当前检测任务的完成信号,状态容器监听线程收到信号后查看其他检测任务的执行状态,若其他检测任务的执行状态未完成,说明存在未结束的检测任务正在执行中,状态容器监听线程将继续保持监听;当最后一个检测任务完成并向状态容器监听线程发送完成信号时,表示全部检测任务的执行状态皆更新为完成态,状态容器监听线程结束监听,汇总多个检测任务的执行状态,使得检测服务器确定应用程序的检测结果并输出。

[0113] 在本发明实施例中,进一步地,检测任务的执行状态的执行前还可以进一步区分,包括未设置和未开始。

[0114] 更进一步地,检测任务的执行状态可以包括执行前、执行中、检测成功和检测失败,检测服务器无需为各个执行线程设置任务状态容器,只需要设置单独状态容器监听线程对各个执行线程的执行状态进行监听即可确定检测结果。

[0115] 在本发明实施例中,通过获取待检测应用程序的应用数据;确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果等步骤,能够减少应用程序的检测时间,提高APP的检测效率,保护检测设备,降低检测过程对检测设备的损耗,优化检测结果的判定过程,提高检测成功率,降低控制成本和检测成本。

[0116] 图4是根据本发明实施例的应用程序的检测装置的主要模块的示意图,如图4所示,本发明的应用程序的检测装置400包括:

[0117] 接收模块401,用于接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据。

[0118] 在本发明实施例中,检测服务器的所述接收模块401接收终端发送的待检测应用程序的检测请求。

[0119] 分配模块402,用于确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程。

[0120] 在本发明实施例中,检测服务器确定与应用数据的静态检测条件参数对应的静态检测任务、与应用数据的自启动行为检测条件参数对应的自启动行为检测任务、与应用数据的动态检测条件参数对应的动态检测任务,利用分配模块402分别为静态检测任务、自启动行为检测任务和动态检测任务分配静态执行线程、自启动执行线程和动态执行线程。

[0121] 执行模块403,用于在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态。

[0122] 在本发明实施例中,检测服务器的所述执行模块403利用静态执行线程、自启动执行线程和动态执行线程并行静态检测任务、自启动行为检测任务和动态检测任务,静态执

行线程、自启动执行线程和动态执行线程同时运行，并分别确定静态检测任务、自启动行为检测任务和动态检测任务的执行状态。

[0123] 结果模块404，用于根据多个所述检测任务的所述执行状态，确定所述应用程序的检测结果。

[0124] 在本发明实施例中，检测服务器的所述结果模块404根据静态检测任务、自启动行为检测任务和动态检测任务的执行状态，确定待检测应用程序的检测结果。

[0125] 在本发明实施例中，通过接收模块、分配模块、执行模块和结果模块等模块，能够减少应用程序的检测时间，提高APP的检测效率，保护检测设备，降低检测过程对检测设备的损耗，优化检测结果的判定过程，提高检测成功率，降低控制成本和检测成本。

[0126] 图5示出了适于应用于本发明实施例的应用程序的检测方法或应用程序的检测装置的示例性系统架构图，如图5所示，本发明实施例的应用程序的检测方法或应用程序的检测装置的示例性系统架构包括：

[0127] 如图5所示，系统架构500可以包括终端设备501、502、503，网络504和服务器505。网络504用以在终端设备501、502、503和服务器105之间提供通信链路的介质。网络504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。

[0128] 用户可以使用终端设备501、502、503通过网络504与服务器505交互，以接收或发送消息等。终端设备501、502、503上可以安装有各种通讯客户端应用，例如购物类应用、网页浏览器应用、搜索类应用、即时通信工具、邮箱客户端、社交平台软件等。

[0129] 终端设备501、502、503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。

[0130] 服务器505可以是提供各种服务的服务器，例如对用户利用终端设备501、502、503所浏览的购物类网站提供支持的后台管理服务器。后台管理服务器可以对接收到的应用程序的检测请求等数据进行分析等处理，并将处理结果（例如应用程序的检测报告）反馈给终端设备501、502、503。

[0131] 需要说明的是，本发明实施例所提供的应用程序的检测方法一般由服务器505执行，相应地，应用程序的检测装置一般设置于服务器505中。

[0132] 应该理解，图5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。

[0133] 图6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，如图6所示，本发明实施例的终端设备或服务器的计算机系统600包括：

[0134] 中央处理单元（CPU）601，其可以根据存储在只读存储器（ROM）602中的程序或者从存储部分608加载到随机访问存储器（RAM）603中的程序而执行各种适当的动作和处理。在RAM603中，还存储有系统600操作所需的各种程序和数据。CPU601、ROM602以及RAM603通过总线604彼此相连。输入/输出（I/O）接口605也连接至总线604。

[0135] 以下部件连接至I/O接口605：包括键盘、鼠标等的输入部分606；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607；包括硬盘等的存储部分608；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609。通信部分609经由诸如因特网的网络执行通信处理。驱动器610也根据需要连接至I/O接口605。可拆卸介质611，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610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608。

[0136] 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609从网络上被下载和安装,和/或从可拆卸介质611被安装。在该计算机程序被中央处理单元(CPU)601执行时,执行本发明的系统中限定的上述功能。

[0137] 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——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。

[0138] 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。

[0139] 描述于本发明实施例中所涉及到的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:一种处理器包括接收模块、分配模块、执行模块和结果模块。其中,这些模块的名称在某种情况下并不构成对该模块本身的限定,例如,分配模块还可以被描述为“确定与所述应用数据对应的多个检测任务并为所述多个检测任务分配多个执行线程的模块”。

[0140] 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:接收待检测应用程序的检测请求;其中,所述检测请求指示了所述待检测应用程序的应用数据;确定与所述应用数据对应的多个检测任务,为所述多个检测任务分配多个执行线程;其中,不同检测任务分别对应不同的执行线程;在所述待检测应用程序运行期间,利用所述多个执行线程并行执行所述检测任务,确定多个所述检测任务的执行状态;根据多个所述检测任务的所述执行状态,确定所述应用程序的检测结果。

[0141] 根据本发明实施例的技术方案,还可以将检测流程中的每个检测任务拆解到一台检测设备的沙盒运行环境中执行,从而利用多台检测设备完成检测流程。

[0142] 根据本发明实施例的技术方案,分别为各个检测任务分配独立的执行线程,彼此之间的执行状态互不影响,运行时各个检测任务之间不存在串行依赖关系,不产生前后依赖,缩短了检测时间;通过应用数据作为公共数据源,各个检测任务从公共数据源获取检测参数条件,无需依赖、等待前一个检测任务的执行结果,提高了检测效率;检测设备无需重启及再次初始化,提高了检测设备的使用寿命,提升了检测设备的检测性能;任务状态容器能够实时掌控执行线程的执行状态,在执行结果为执行失败的情况下控制执行线程进行重试,提高了执行线程对应的检测任务的成功率,增强了检测流程执行过程中的稳定性、以及检测流程的成功率。

[0143] 根据本发明实施例的技术方案,能够减少应用程序的检测时间,提高APP的检测效率,保护检测设备,降低检测过程对检测设备的损耗,优化检测结果的判定过程,提高检测成功率,降低控制成本和检测成本。

[0144] 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。

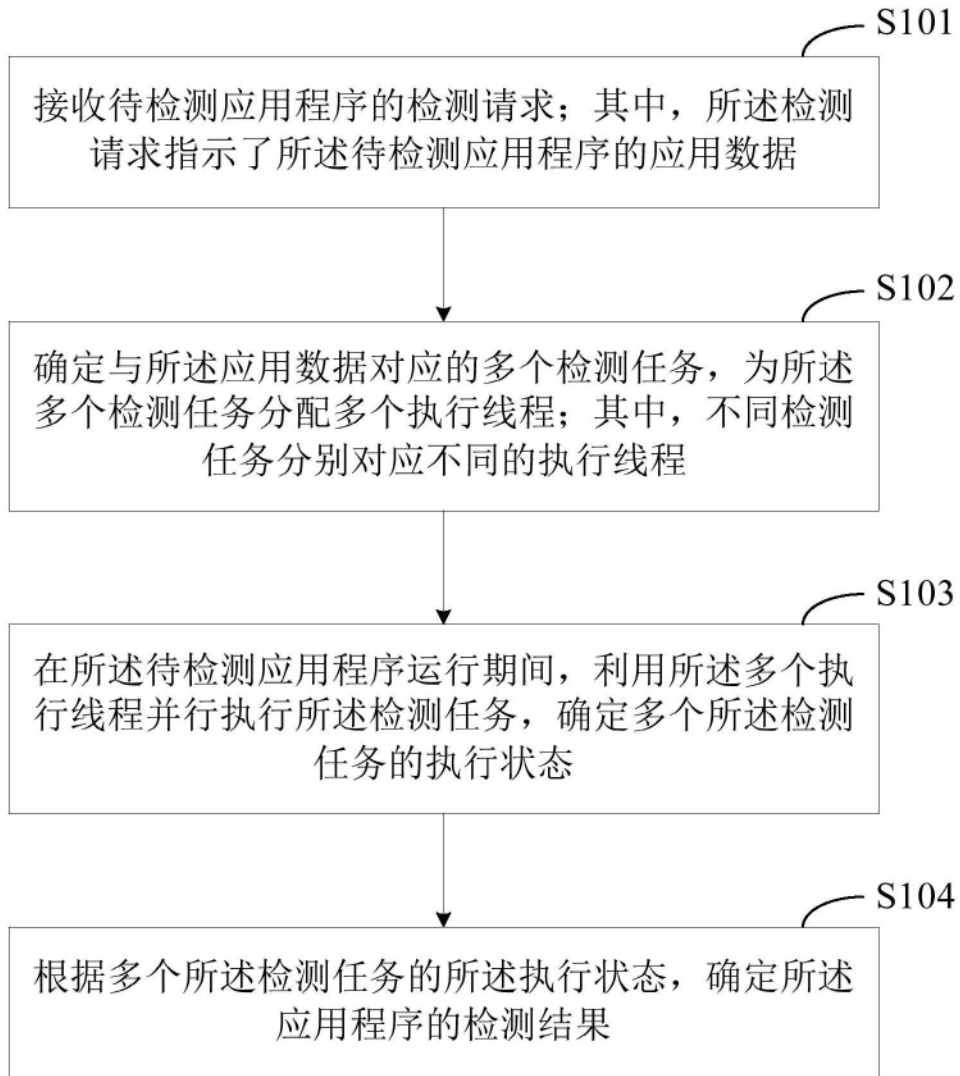


图1

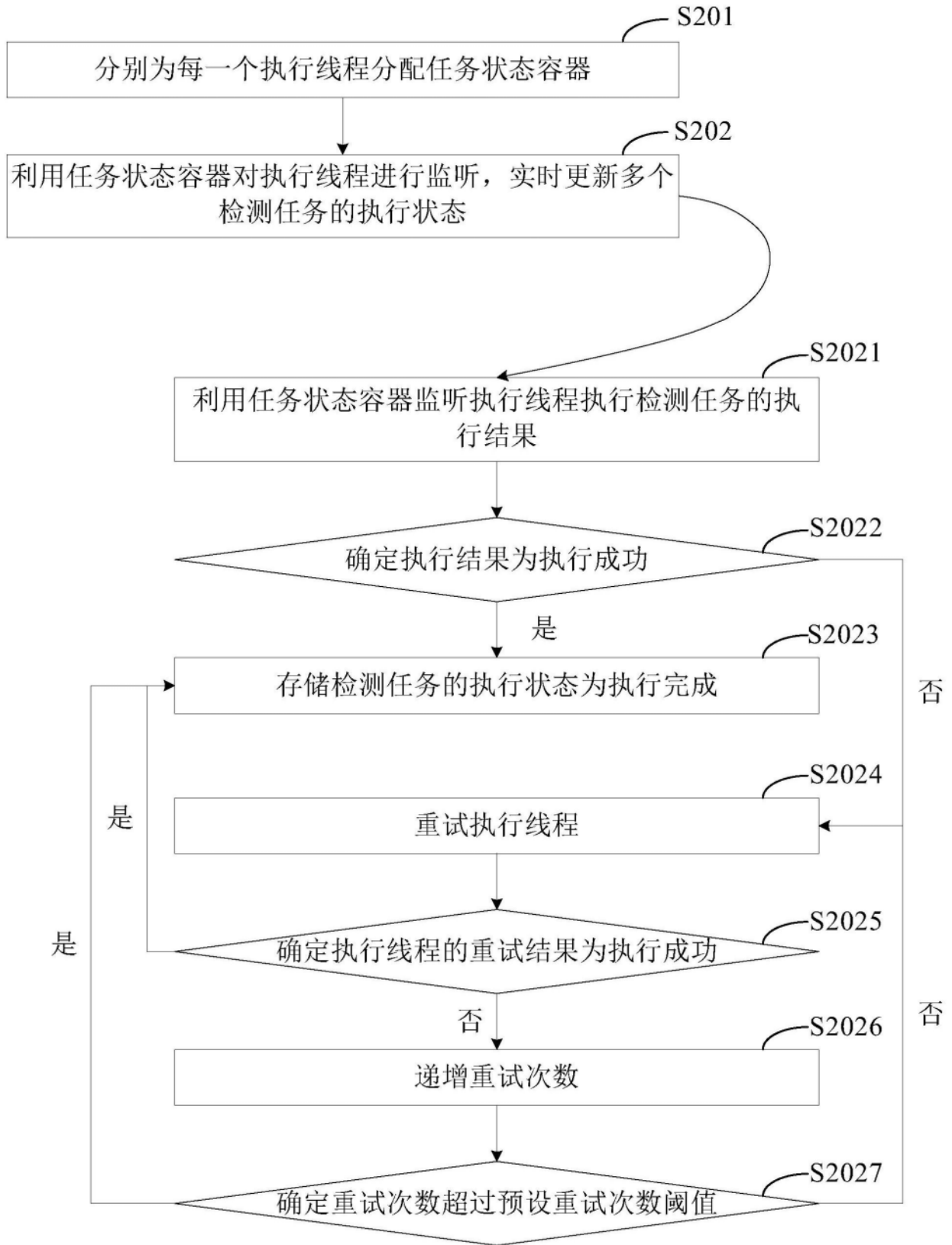


图2



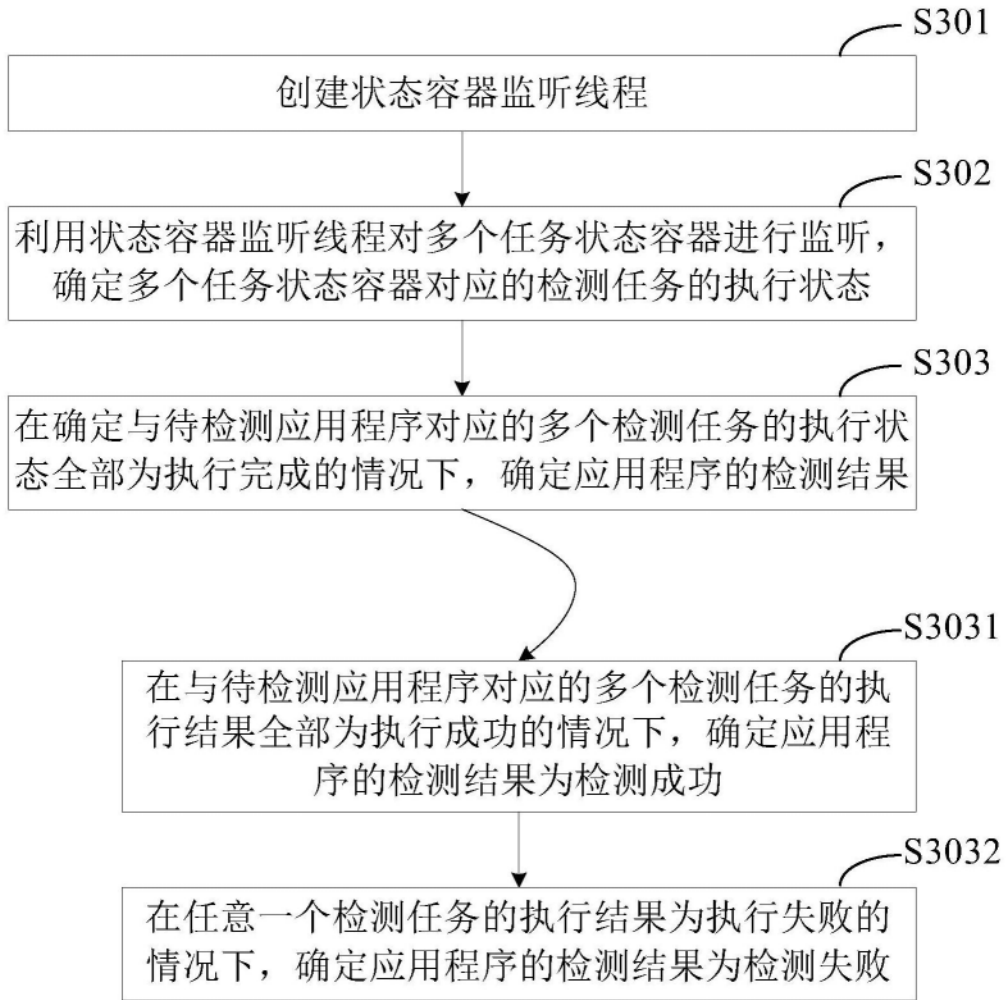


图3

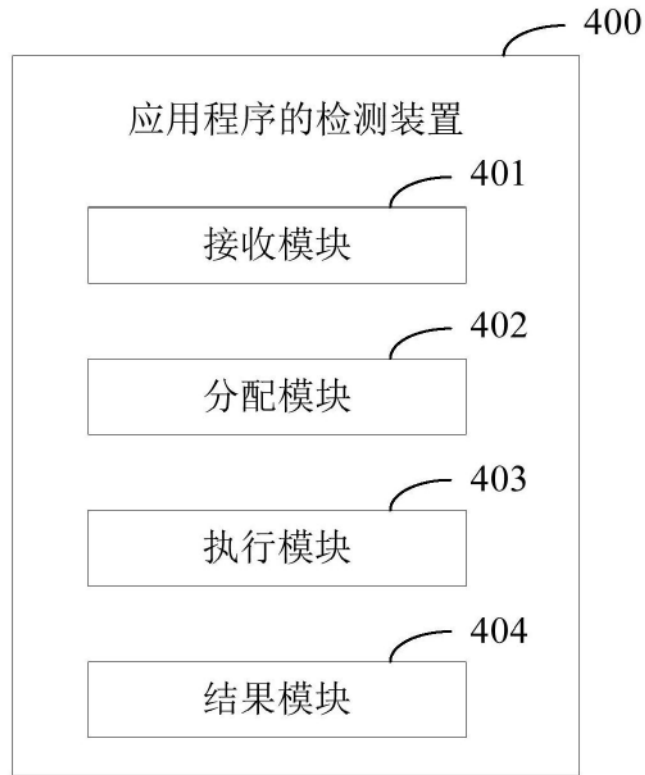


图4

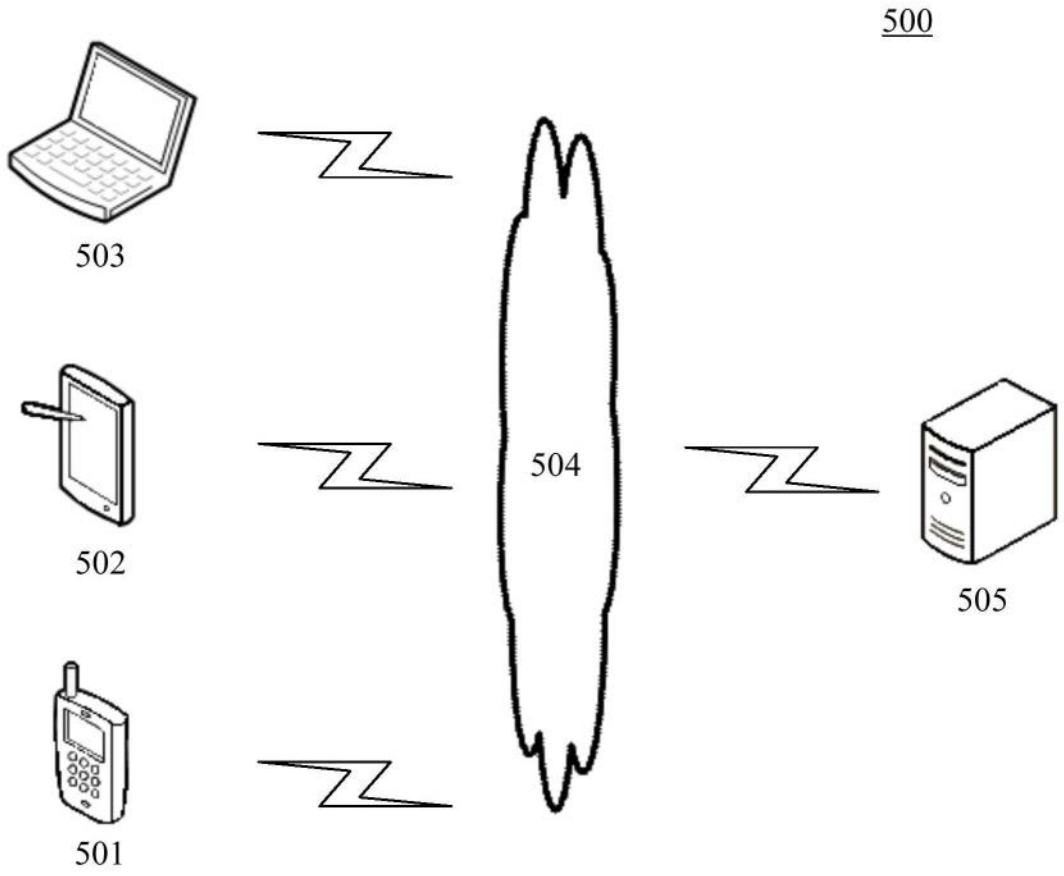


图5

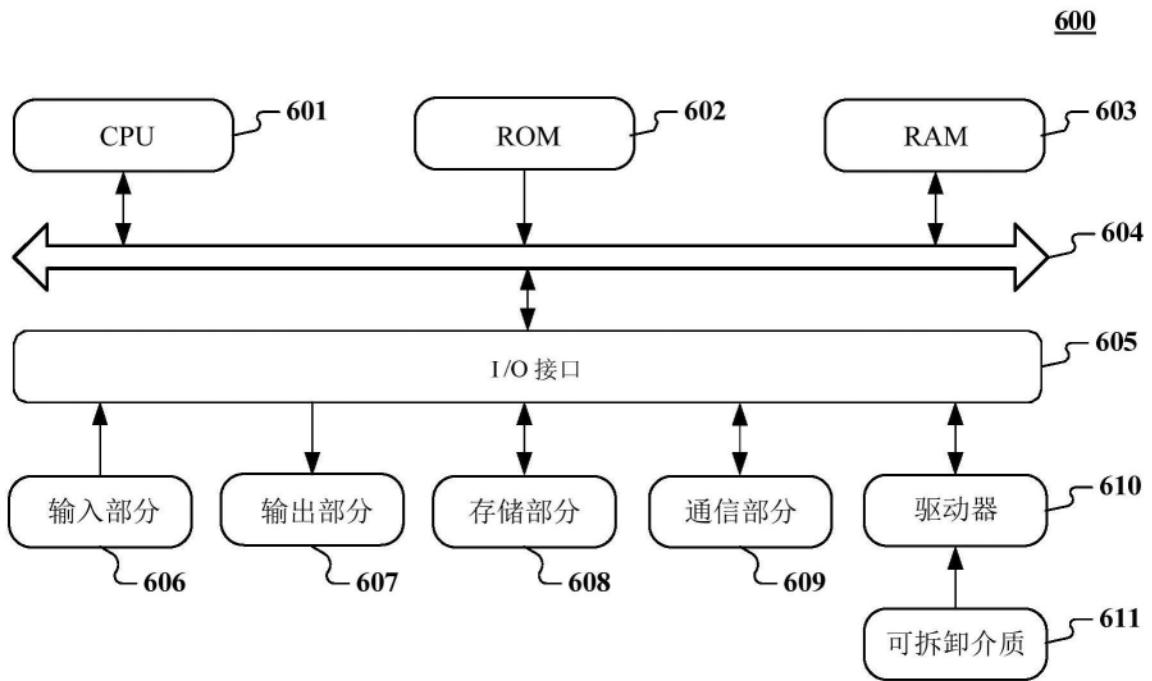


图6