



(12)发明专利申请

(10)申请公布号 CN 106484622 A

(43)申请公布日 2017.03.08

(21)申请号 201610906401.2

(22)申请日 2016.10.18

(71)申请人 网易(杭州)网络有限公司

地址 310000 浙江省杭州市滨江区长河街
道网商路599号4幢7层

(72)发明人 袁定波 孙圣翔 洪一帆

(74)专利代理机构 北京博浩百睿知识产权代理
有限责任公司 11134

代理人 宋子良

(51) Int. Cl.

G06F 11/36(2006.01)

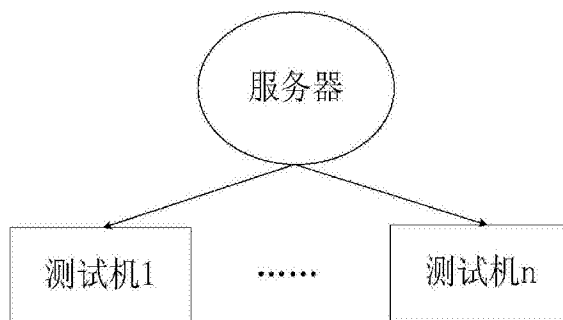
权利要求书3页 说明书11页 附图4页

(54)发明名称

测试方法、装置及系统

(57)摘要

本发明公开了一种测试方法、装置及系统。该系统包括服务器和一个或多个测试机,其中,服务器用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,网页服务提供的一个或多个网页包括以下至少之一:用于上传待测试软件包的网页、用于上传自定义测试信息的网页,服务器还用于根据自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,测试任务用于在不同测试设备上测试对应的软件包;测试机,与至少一个测试设备连接,用于在接收到来自服务器的测试任务之后,控制与其连接测试设备,以安装对应的软件包并进行测试。通过上述实施例解决了人力时间消耗较大的技术问题,提高了测试效率。



1. 一种测试系统,其特征在于,包括:服务器和一个或多个测试机,其中,

所述服务器,用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,其中,所述网页服务提供的一个或多个网页包括以下至少之一:用于上传待测试软件包的网页、用于上传自定义测试信息的网页,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;

所述服务器,还用于根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在不同测试设备上测试对应的软件包;

所述测试机,与至少一个测试设备连接,用于在接收到来自所述服务器的测试任务之后,控制与其连接测试设备,以安装对应的软件包并进行测试。

2. 根据权利要求1所述的系统,其特征在于,

所述测试机,还用于收集与其连接的测试设备的测试结果,并将所述测试结果发送给所述服务器;

所述服务器的网页服务提供的网页还包括:用于展示测试报告的网页,其中,所述测试报告是根据所述测试结果生成的。

3. 根据权利要求1或2所述的系统,其特征在于,所述测试信息包括以下至少之一:测试参数、用户自定义的第二测试脚本,其中,所述第二测试脚本用于控制所述测试设备执行软件包的安装,和/或,安装所述软件包之后控制所述软件包对应的软件执行操作,所述测试参数为所述第二测试脚本在被执行的过程中使用到的参数。

4. 根据权利要求3所述的系统,其特征在于,所述测试结果包括以下至少之一:执行测试的测试设备的信息、安装所述软件包成功或失败的信息、控制所述软件执行所述操作的信息。

5. 根据权利要求1所述的系统,其特征在于,

所述服务器,还用于获取在线的测试机,并向在线的所述测试机分发所述测试任务。

6. 根据权利要求5所述的系统,其特征在于,

所述服务器,还用于为在线的所述测试机连接的至少一个测试设备分别维护任务队列,其中,每个任务队列记录了该测试设备执行的测试任务;

所述服务器,还用于根据每个测试设备对应的任务队列中测试任务的数量向测试设备分发测试任务。

7. 根据权利要求1所述的系统,其特征在于,所述系统还包括:

一个或多个所述测试设备,与所述测试机连接,用于在所述测试机的控制下安装对应的软件包,以及控制所述软件包对应的软件进行操作。

8. 根据权利要求7所述的系统,其特征在于,所述测试设备,用于控制所述软件进行操作包括:对所述软件对应的控件进行识别和操作。

9. 根据权利要求8所述的系统,其特征在于,所述测试设备,用于根据控件类别、资源标识以及控件文本内容来对控件进行识别和操作;或者,用于根据图像匹配来获取所述控件并对获取到的控件进行操作。

10. 根据权利要求1所述的系统,其特征在于,所述不同版本的软件包对应同一款软件,所述软件的每一版本对应一个不同的发布该软件的发布渠道。

11. 一种测试方法,其特征在于,包括:

向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,

通过所述网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;

根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;

接收来自所述测试机的测试结果并通过网页展示测试报告,其中,所述测试报告是根据所述测试结果生成的。

12. 根据权利要求11所述的方法,其特征在于,所述测试信息包括以下至少之一:测试参数、用户自定义的第二测试脚本,其中,所述第二测试脚本用于控制所述测试设备执行软件包的安装,和/或,安装所述软件包之后控制所述软件包对应的软件执行操作,所述测试参数为所述第二测试脚本在被执行的过程中使用到的参数。

13. 根据权利要求12所述的方法,其特征在于,所述测试结果包括以下至少之一:执行测试的测试设备的信息、安装所述软件包成功或失败的信息、控制所述软件执行所述操作的信息。

14. 根据权利要求11所述的方法,其特征在于,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发包括:

获取在线的测试机,并向在线的所述测试机分发所述测试任务。

15. 根据权利要求14所述的方法,其特征在于,获取在线的测试机,并向在线的所述测试机分发所述测试任务包括:

获取每个测试设备已经得到的测试任务;

根据每个测试设备执行的任务获取所述每个测试设备的负荷;

根据所述每个测试设备的负荷为所述每个测试设备分发测试任务。

16. 根据权利要求15所述的方法,其特征在于,获取每个测试设备已经得到的测试任务包括:

通过为测试设备分别维护任务队列获取所述每个测试设备已经得到的测试任务。

17. 一种测试方法,其特征在于,包括:

获取来自服务器的至少一个待测试软件包以及测试信息,其中,所述测试信息为对所述至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;

根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;

接收来自所述至少一个测试设备的测试结果并发送给所述服务器。

18. 根据权利要求17所述的方法,其特征在于,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发包括:

获取所述至少一个测试设备的硬件信息、操作系统的信息和/或资源使用情况;

根据所述测试信息中的对测试设备的要求向符合所述要求的测试机分发所述测试任

务。

19. 一种测试装置,其特征在于,包括:

提供模块,用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本;

第一获取模块,通过所述网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;

第一分发模块,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;

展示模块,用于接收来自所述测试机的测试结果并通过网页展示测试报告,其中,所述测试报告是根据所述测试结果生成的。

20. 一种测试装置,其特征在于,包括:

第二获取模块,用于获取来自服务器的至少一个待测试软件包以及测试信息,其中,所述测试信息为对所述至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;

第二分发模块,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;

发送模块,用于接收来自所述至少一个测试设备的测试结果并发送给所述服务器。

测试方法、装置及系统

技术领域

[0001] 本发明涉及软件测试,具体而言,涉及测试方法、装置及系统。

背景技术

[0002] 目前,对于某个特定的软件(或者称为应用)需要根据系统的不同制作不同的版本(称为软件包或者应用包),对于每个版本均需要进行测试,例如,目前比较流行的安卓(Android)系统由于其系统的开放性,导致其发布渠道种类很多。据保守统计,目前主流的发布渠道至少有五十多个。因此,在应用(例如,游戏)的开发和运营过程中,不同的渠道包均需要进行测试(例如,可以测试安装、启动、登录、支付等)。

[0003] 为了保证各个游戏渠道包功能正常,在现有技术中,需要人工地将每一个游戏渠道包安装在不同的设备上,然后人为地完成各种功能的测试。这一过程需要耗费大量的人力和时间。

[0004] 针对上述的问题,目前尚未提出有效的解决方案。

发明内容

[0005] 本发明实施例提供了测试方法、装置及系统,以至少解决现有技术不同应用版本需要人为进行测试所导致的人力时间消耗较大技术问题。

[0006] 根据本发明实施例的一个方面,提供了一种测试系统,包括:服务器和一个或多个测试机,其中,所述服务器,用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,其中,所述网页服务提供的一个或多个网页包括以下至少之一:用于上传待测试软件包的网页、用于上传自定义测试信息的网页,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;所述服务器,还用于根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在不同测试设备上测试对应的软件包;所述测试机,与至少一个测试设备连接,用于在接收到来自所述服务器的测试任务之后,控制与其连接测试设备,以安装对应的软件包并进行测试。

[0007] 进一步地,所述测试机,还用于收集与其连接的测试设备的测试结果,并将所述测试结果发送给所述服务器;所述服务器的网页服务提供的网页还包括:用于展示测试报告的网页,其中,所述测试报告是根据所述测试结果生成的。

[0008] 进一步地,所述测试信息包括以下至少之一:测试参数、用户自定义的第二测试脚本,其中,所述第二测试脚本用于控制所述测试设备执行软件包的安装,和/或,安装所述软件包之后控制所述软件包对应的软件执行操作,所述测试参数为所述第二测试脚本在被执行的过程中使用到的参数。

[0009] 进一步地,所述测试结果包括以下至少之一:执行测试的测试设备的信息、安装所述软件包成功或失败的信息、控制所述软件执行所述操作的信息。

[0010] 进一步地,所述服务器,还用于获取在线的测试机,并向在线的所述测试机分发所

述测试任务。

[0011] 进一步地,所述服务器,还用于为在线的所述测试机连接的至少一个测试设备分别维护任务队列,其中,每个任务队列记录了该测试设备执行的测试任务;所述服务器,还用于根据每个测试设备对应的任务队列中测试任务的数量向测试设备分发测试任务。

[0012] 进一步地,所述系统还包括:一个或多个所述测试设备,与所述测试机连接,用于在所述测试机的控制下安装对应的软件包,以及控制所述软件包对应的软件进行操作。

[0013] 进一步地,所述测试设备,用于控制所述软件进行操作包括:对所述软件对应的控件进行识别和操作。

[0014] 进一步地,所述测试设备,用于根据控件类别、资源标识以及控件文本内容来对控件进行识别和操作;或者,用于根据图像匹配来获取所述控件并对获取到的控件进行操作。

[0015] 进一步地,所述不同版本的软件包对应同一款软件,所述软件的每一版本对应一个不同的发布该软件的发布渠道。

[0016] 根据本实施例的另一个方面,还提供了一种测试方法,包括:向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,通过所述网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;接收来自所述测试机的测试结果并通过网页展示测试报告,其中,所述测试报告是根据所述测试结果生成的。

[0017] 进一步地,所述测试信息包括以下至少之一:测试参数、用户自定义的第二测试脚本,其中,所述第二测试脚本用于控制所述测试设备执行软件包的安装,和/或,安装所述软件包之后控制所述软件包对应的软件执行操作,所述测试参数为所述第二测试脚本在被执行的过程中使用到的参数。

[0018] 进一步地,所述测试结果包括以下至少之一:执行测试的测试设备的信息、安装所述软件包成功或失败的信息、控制所述软件执行所述操作的信息。

[0019] 进一步地,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发包括:获取在线的测试机,并向在线的所述测试机分发所述测试任务。

[0020] 进一步地,获取在线的测试机,并向在线的所述测试机分发所述测试任务包括:获取每个测试设备已经得到的测试任务;根据每个测试设备执行的任务获取所述每个测试设备的负荷;根据所述每个测试设备的负荷为所述每个测试设备分发测试任务。

[0021] 进一步地,获取每个测试设备已经得到的测试任务包括:通过为测试设备分别维护任务队列获取所述每个测试设备已经得到的测试任务。

[0022] 根据本发明实施例的另一个方面,还提供了一种测试方法,包括:获取来自服务器的至少一个待测试软件包以及测试信息,其中,所述测试信息为对所述至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;接收来自所述至少一个测试设备的测试结果并发送给所述服务器。

[0023] 进一步地,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发包括:获取所述至少一个测试设备的硬件信息、操作系统的信息和/或资源使用情况;根据所述测试信息中的对测试设备的要求向符合所述要求的测试机分发所述测试任务。

[0024] 根据本发明实施例的另一个方面,还提供了一种测试装置,包括:提供模块,用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本;第一获取模块,通过所述网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,所述测试信息为对所述待测试软件包进行测试所需要使用的用户自定义信息;第一分发模块,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;展示模块,用于接收来自所述测试机的测试结果并通过网页展示测试报告,其中,所述测试报告是根据所述测试结果生成的。

[0025] 根据本发明实施例的另一个方面,还提供了一种测试装置,包括:第二获取模块,用于获取来自服务器的至少一个待测试软件包以及测试信息,其中,所述测试信息为对所述至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;第二分发模块,根据所述自定义测试信息及与所述软件包对应的第一测试脚本生成测试任务并向所述测试机进行分发,其中,所述测试任务用于在所述测试机连接的不同测试设备上测试对应的软件包;发送模块,用于接收来自所述至少一个测试设备的测试结果并发送给所述服务器。

[0026] 在本发明实施例中,采用服务器和一个或多个测试机,其中,所述服务器,用于向用户提供网页服务,其中,所述网页服务提供的一个或多个网页包括以下至少之一:用于上传待测试软件包的网页、用于上传测试信息的网页,所述测试信息为对所述待测试软件包进行测试所使用的信息;所述服务器,还用于向所述测试机分发测试任务,其中,所述测试任务用于在不同测试设备上测试对应的软件包;所述测试机,与至少一个测试设备连接,用于在接收到来自所述服务器的测试任务之后,控制与其连接测试设备,以安装对应的软件包并进行测试。通过上述实施例解决了人力时间消耗较大的技术问题,提高了测试效率。

附图说明

[0027] 此处所说明的附图用来提供对本发明的进一步理解,构成本申请的一部分,本发明的示意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。在附图中:

[0028] 图1是根据本发明实施例的测试系统的示意图;

[0029] 图2是根据本发明实施例的测试方法的流程图;

[0030] 图3是根据本发明实施例的另一测试方法的流程图;

[0031] 图4是根据本发明实施例的测试装置的结构框图;

[0032] 图5是根据本发明实施例的另一测试装置的结构框图;

[0033] 图6是根据本发明实施例的可选的游戏测试示意图;

[0034] 图7是根据本发明实施例的任务分发的示意图;

[0035] 图8是根据本发明实施例的执行测试任务的示意图;以及

[0036] 图9是根据本发明实施例的Server-Agent的分布式架构示意图。

具体实施方式

[0037] 为了使本技术领域的人员更好地理解本发明方案,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分的实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本发明保护的范围。

[0038] 需要说明的是,本发明的说明书和权利要求书及上述附图中的术语“第一”、“第二”等是用于区别类似的对象,而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换,以便这里描述的本发明的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外,术语“包括”和“具有”以及他们的任何变形,意图在于覆盖不排他的包含,例如,包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元,而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0039] 需要说明的是,在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行,并且,虽然在流程图中示出了逻辑顺序,但是在某些情况下,可以以不同于此处的顺序执行所示出或描述的步骤。

[0040] 在本实施例中,提供了一种测试系统,图1是根据本发明实施例的测试系统的示意图,如图1所示,本测试系统包括:服务器和一个或多个测试机。下面对此进行说明。

[0041] 本实施例中的服务至少具有两个功能:

[0042] 功能一:

[0043] 向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,其中,网页服务提供的一个或多个网页包括以下至少之一:用于上传待测试软件包的网页、用于上传自定义测试信息的网页,该测试信息为对待测试软件包进行测试所需要使用的用户自定义信息。

[0044] 提供网页服务可以采用多种形式来进行,例如,可以使用开源的web服务器,也可以使用windows系统提供的网页服务。提供网页服务的优势在于,可以让用户(即测试人员)通过比较便利的方式上传软件包以及测试所需要使用的信息。

[0045] 功能二:

[0046] 根据所述自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,测试任务用于在不同测试设备上测试对应的软件包。

[0047] 同一款软件可能对应不同的软件包,这些软件包可以安装在不同的设备上,此时,服务器还承担了向测试机分发测试任务的功能。

[0048] 由于涉及了多个测试设备,在本系统中,使用了测试机于服务和测试设备连接。即每一台测试机都可以与至少一个测试设备连接,用于在接收到来自服务器的测试任务之后,控制与其连接测试设备,以安装对应的软件包并进行测试。

[0049] 在一个可选的实施方式中,不同版本的软件包对应同一款软件,这一款软件的每一版本对应一个不同的发布该软件的发布渠道,对渠道包测试时,测试脚本包括两部分:第一部分是web服务提供各个渠道的测试脚本,用于模拟渠道登陆的各个操作步骤,例如:点

击哪个图标进行登陆、点击哪个控件进行用户名/密码输入等等,第一部分是核心的测试脚本,也即第一测试脚本;第二部分是用户自定义的测试脚本,也即第二测试脚本,主要用于模拟在进入登陆界面前的点击操作,例如,有的游戏在登陆界面之前,需要点击用户电子协议、游戏公告等等。

[0050] 网页服务器会提供各个渠道的游戏登录、渠道支付、卸载包体、回发测试结果等核心功能的测试脚本,用户通过网页选择指定的渠道即可。

[0051] 游戏软件包括游戏逻辑主体部分和用于实现登陆、支付等功能的SDK,同一个游戏由于接入不同的渠道SDK就会形成不同的渠道包,(例如:华为渠道包、腾讯渠道包),接不同的渠道SDK游戏的登陆、付费方式会有差异,例如:腾讯SDK一般可用微信、QQ登陆,而华为SDK用户可用华为账号登陆。正是由于不同SDK登陆、付费的步骤、需点击的控件位置等存在差异,不同渠道包登陆、付费的测试脚本就会有差异,而常规的游戏渠道有50-100家,如果每个游戏要自行测试,工作量巨大,本发明实施例的web服务器会提供不同渠道的测试脚本,用户只需上传自定义的部分并选择在web服务器提供的渠道列表上选择渠道包对应的渠道测试脚本即可。

[0052] 在一个可选的实施方式中,由于服务器连接多个测试机,为了更加便于分发测试任务,服务器可以获取在线的测试机,并向在线的测试机分发测试任务。服务器可以通过向测试机发送消息看其是否有响应消息来判断测试机是否在线,或者,测试机也可以通过心跳消息来向服务器说明在线。在另外一个实施方式中,服务器还可以通过测试机得知其连接的测试设备的情况,然后记录每个测试设备正在执行的测试,即服务器,可以为在线的测试机连接的至少一个测试设备分别维护任务队列,其中,每个任务队列记录了该测试设备执行的测试任务,由于记录了每个测试设备已经被分发的测试任务,从而可以根据每个测试设备对应的任务队列中测试任务的数量向测试设备分发测试任务。通过该可选实施方式可以做到负载均衡。

[0053] 通过上述系统中的服务器和测试机,测试人员只需要通过网页选择指定的渠道,而网页服务器提供各个渠道的游戏登录、渠道支付、卸载包体、回发测试结果等核心功能的测试脚本,服务器和测试机相互配合既可以完成测试。相比于现有技术中需要人工进行安装测试包并进行测试的技术方案,解决了人力时间消耗较大的问题,提高了测试效率。

[0054] 在一个可选的实施方式中,不仅仅可以做到测试任务的自动执行,还可以自动得到测试结果,在本可选实施方式中,每个测试机,还可以用于收集与其连接的测试设备的测试结果,并将测试结果发送给服务器。此时为了便于测试人员查看测试结果,服务器的网页服务还可以提供用于展示测试报告的网页,其中,测试报告是根据测试结果生成的。测试报告可以由服务器根据各个测试机上报的测试结果生成的,或者,也可以是由测试机直接生成测试报告,服务器仅仅用于展示测试上报的测试报告即可。

[0055] 为了便于测试人员使用,测试人员可以通过网页选择指定的渠道,无需上传测试使用的脚本,该脚本可以理解为需要控制测试设备所执行的动作,例如,控制测试设备执行软件包的安装,和/或,安装软件包之后控制软件包对应的软件执行操作。在执行用户自定义的第二测试脚本的时候还有可能使用到相应的参数,这些测试参数可以为测试脚本在执行过程中所使用的参数。测试参数和用户自定义的第二测试脚本都可以理解为是测试信息的一部分。

[0056] 作为另一个可选的实施例,测试结果中包括的内容也可以由测试人员进行配置,例如,测试结果可以包括以下至少之一:执行测试的测试设备的信息、安装软件包成功或失败的信息、控制软件执行操作的信息。测试结果的定制可以根据目前软件包的状态来配置,例如,在软件包刚开始进行测试的,需要展示更多的测试结果;而在软件包邻近发布的时候,可以配置展示相对少一点的内容。为了便于测试人员的使用,可以提供结果模板供测试人员选择,不同的测试模板展示了不同的测试结果包括的内容。这样的一种处理方式使测试人员对测试结果可以更加灵活的配置。

[0057] 对于测试设备而言,其可以随时连接到测试机上,这些测试设备可以至专门用于测试的,也可以是通常所使用的设备,仅仅为了测试的需要而连接到测试机上的。因此,作为测试系统而言,该可以包括一个或多个测试设备,与测试机连接,用于在测试机的控制下安装对应的软件包,以及控制软件包对应的软件进行操作。这些测试设备也可以不包括系统中,每个测试人员可以随时连接自己需要的设备到测试机上即可。测试机可以自动识别连接到其上的测试设备,也可以由测试人员在服务器或者测试机上进行配置以添加新的测试设备。

[0058] 控制测试设备在对软件执行操作的过程中,软件中如果涉及到控件需要进行操作,则需要对软件中的控件进行识别和操作。进行识别和操作的方式有很多种,例如,可以根据控件类别、资源标识以及控件文本内容来对控件进行识别和操作;或者,也可以根据图像匹配来获取控件并对获取到的控件进行操作。

[0059] 作为一种可选的实施方式,不同版本的软件包对应同一款软件,软件的每一版本对应一个不同的发布该软件的发布渠道。

[0060] 在上述系统中,服务器执行了一种测试的方法,图2是根据本发明实施例的测试方法的流程图,如图2所示,该方法包括如下步骤:

[0061] 步骤S202,向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本,通过网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,测试信息为对待测试软件包进行测试所需要使用的用户自定义信息;

[0062] 步骤S204,根据自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,测试任务用于在测试机连接的不同测试设备上测试对应的软件包;

[0063] 步骤S206,接收来自测试机的测试结果并通过网页展示测试报告,其中,测试报告是根据测试结果生成的。

[0064] 通过上述步骤,可以做到自动的测试,上述执行的步骤可以是服务器,也可以是其他的装置或者设备,只要能够做到上述步骤就可以解决本发明所解决的技术问题。

[0065] 作为一个可选的实施方式,步骤S204,在分发测试任务的时候,可以获取每个测试设备已经得到的测试任务;根据每个测试设备执行的任务获取每个测试设备的负荷;根据每个测试设备的负荷为每个测试设备分发测试任务。作为另一个可选的实施方式,测试设备的测试任务可以通过任务队列的方式来进行维护。

[0066] 在本实施例中还提供了一种测试方法,该测试方式可以在测试机中执行,当然,也可以在其他设备或装置中执行,无论在什么设备或装置中执行,只要执行了以下步骤就可以实现相应的功能。图3是根据本发明实施例的另一种测试方法的流程图,如图3所示,该流

程包括如下步骤：

[0067] 步骤S302,获取来自服务器的至少一个待测试软件包以及测试信息,其中,测试信息为对至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;

[0068] 步骤S304,根据自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,测试任务用于在测试机连接的不同测试设备上测试对应的软件包;

[0069] 步骤S306,接收来自至少一个测试设备的测试结果并发送给服务器。

[0070] 步骤S204和步骤S304均涉及到测试任务的分发,测试任务的分发可以由服务器来进行,服务器将测试任务直接分发给测试设备,也可以将测试任务分发给测试机,然后由测试机向测试设备分发。同理,服务器也可以不执行任务分发,而由测试机来进行分发,测试机获取至少一个测试设备的硬件信息、操作系统的信息和/或资源使用情况;根据测试信息中的对测试设备的要求向符合要求的测试机分发测试任务。不同的测试机之间还可以进行消息的互通,测试机可以将原本自己执行的任务分发给其他的测试机。如果分发给其他测试机的任务是服务器分发的,此时可以告知服务器任务被分发给其他测试机了。通过该可选的实施方式可以使测试更加灵活。

[0071] 对应于上述图2中的步骤,在本实施例中还提供了一种测试装置,图4是根据本发明实施例的测试装置的结构框图,如图4所示,该装置可以包括如下模块:

[0072] 提供模块42,用于向用户提供网页服务以及提供预设的不同版本的软件包对应的第一测试脚本;

[0073] 第一获取模块44,通过网页服务提供的一个或多个网页获取用户上传待测试软件包以及自定义测试信息,其中,测试信息为对待测试软件包进行测试所需要使用的用户自定义信息;

[0074] 第一分发模块46,根据自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,测试任务用于在测试机连接的不同测试设备上测试对应的软件包;

[0075] 展示模块48,用于接收来自测试机的测试结果并通过网页展示测试报告,其中,测试报告是根据测试结果生成的。

[0076] 对应于上述图3中的步骤,在本实施例中还提供了一种测试装置,图5是根据本发明实施例的另一测试装置的结构框图,如图5所示,该装置可以包括如下模块:

[0077] 第二获取模块52,用于获取来自服务器的至少一个待测试软件包以及测试信息,其中,测试信息为对至少一个待测试软件包中的每个待测试包进行测试所需要使用的用户自定义信息;

[0078] 第二分发模块54,根据自定义测试信息及与软件包对应的第一测试脚本生成测试任务并向测试机进行分发,其中,测试任务用于在测试机连接的不同测试设备上测试对应的软件包;

[0079] 发送模块56,用于接收来自至少一个测试设备的测试结果并发送给服务器。

[0080] 通过上述实施例解决了人力时间消耗较大的技术问题,提高了测试效率。下面以游戏测试为例进行说明。

[0081] 在本可选实施例中提供了一种渠道自动化测试的技术并设计了一个渠道自动化

测试的系统。该渠道自动化测试系统可以基于该渠道自动化测试技术对用户上传的游戏渠道包进行安装、启动、渠道登录、渠道支付的自动化测试,并形成详细的报告呈现给用户。

[0082] 图6是根据本发明实施例的可选的游戏测试示意图,如图6所示,可以将待测游戏渠道包自动分配到不同地待测设备上,自动完成游戏渠道包的安装、启动、渠道登录、渠道支付测试流程,并自动生成详细完整的渠道测试报告。

[0083] 在本实施例中,可以利用服务器(或称为web平台)对待测的渠道包、测试设备、测试脚本、测试结果进行控制和管理。用户(测试人员)通过访问web平台,上传待测的游戏渠道包,用户自定义的脚本以及渠道测试参数(渠道登录用户名、密码、支付密码等);web服务端自动将待测任务分发到各个测试设备上,并开启自动测试流程。

[0084] web服务端从各个测试机获取现有的所有在线测试设备,维护任务分发队列,将渠道测试任务分发到各个测试机所对应的测试设备上。服务端为每个测试机上接入的测试设备(例如,手机)维护一个任务队列。如果用户指定测试设备,则将测试任务分发至相应的测试设备;如果用户没有指定测试设备,则将任务分发至测试任务最小的测试设备。

[0085] 测试完成后,web服务端从测试设备获取测试数据,生成测试报告并web展示。一方面,测试报告包含了待测设备的详细信息和所有渠道在相应测试设备上的测试统计结果,包括安装统计信息(成功、失败)、启动统计信息(成功、失败)、渠道登录统计信息(渠道登录是否成功、失败的渠道和对应的机型)、渠道支付统计信息、卸载统计信息;另一方面,测试报告还包含了每一个测试任务(一个渠道包在一个测试设备上的执行过程)的详细执行信息:平台既提供了任务执行的LOG供用户查看,同时提供了执行过程中的截图供用户查看执行过程及相应画面表现。对用户而言,只需上传待测的游戏渠道包、渠道测试参数、以及自定义的测试脚本,无需关心测试设备的选择和具体测试流程。

[0086] 下面对本实施例中涉及到的技术方案按照步骤进行说明。

[0087] 步骤一,上传待测游戏渠道包、测试参数(渠道登录用户名、密码、支付密码等)以及用户自定义的测试脚本。上传待测游戏渠道包时需标定渠道名称(渠道名称列表由web平台提供,用户选择即可),测试参数按照平台规定的格式填写即可。自定义测试脚本中主要完成游戏交互弹框的点击操作以及调用web平台的渠道登录测试接口。

[0088] 步骤二,渠道测试任务创建与分发。根据步骤一中用户上传的游戏渠道包、测试参数和测试脚本,渠道测试web平台从各个测试机获取现有的所有在线测试设备,维护任务分发队列,将渠道测试任务分发到各个测试机所对应的测试设备上。图7是根据本发明实施例的任务分发的示意图,如图7所示,服务端为每个测试机上接入的测试设备(例如,手机)维护一个任务队列。如果用户指定测试设备,则将测试任务分发至相应的测试设备;如果用户没有指定测试设备,则将任务分发至测试任务最小的测试设备。而每台测试机会为其所连接的测试设备设置任务超时时间,并维护设备状态(空闲,任务中)。

[0089] 步骤三渠道测试任务的执行。测试机在接收到来自渠道测试web平台的测试任务后,开始执行渠道测试流程。图8是根据本发明实施例的执行测试任务的示意图,如图8所示,首先其从测试设备获取设备的参数,包括品牌、内存、cpu、系统版本等信息(其会在测试报告中显示);然后,逐步完成包体的解析、包体安装、包体的启动;当有系统弹框时,平台根据用户自定义脚本点击弹框,进入到渠道登录界面;通过平台提供的渠道自动化登录技术,平台会自动识别渠道登录界面的用户名输入框和密码输入框,然后自动填充用户提供的用

用户名和密码,最后点击相关按钮完成渠道登录;继而,根据用户自定义脚本完成选服等操作,登录到游戏服务器;通过平台提供的渠道自动化测试技术,完成渠道支付测试,具体实现原理为:利用Automator测试框架,对于Android原生的UI,根据控件类别(Class Name),资源ID(Resource ID)以及控件文字(Text)来实现控件的识别和点击,对于非Android原生的UI,根据模板匹配和SIFT匹配两种图像匹配方法来完成控件的识别和点击,从而可以自动完成渠道支付的流程;最后,卸载包体,将上述测试结果从测试机回发至web平台服务器。

[0090] 步骤四,渠道测试结果报告生成。Web平台在收到步骤三回发的测试结果后,会对测试结果进行统计处理,形成完整的测试报告。用户通过Web前端即可浏览相应的测试报告。一方面,测试报告包含了待测设备的详细信息和所有渠道在相应测试设备上的测试统计结果,包括安装统计信息(成功、失败)、启动统计信息(成功、失败)、渠道登录统计信息(渠道登录是否成功、失败的渠道和对应的机型)、渠道支付统计信息、卸载统计信息;另一方面,测试报告还包含了每一个测试任务(一个渠道包在一个测试设备上的执行过程)的详细执行信息:平台既提供了任务执行的LOG供用户查看,同时提供了执行过程中的截图供用户查看执行过程及相应画面表现。

[0091] 利用步骤二,可以方便快捷地将渠道测试任务分发至不同型号的测试设备上,以达到渠道兼容性测试的目的。

[0092] 利用步骤三,可以自动化地进行渠道的安装、启动、登录、支付、卸载测试,脱离人工,可以节省大量的人力成本和时间成本,提高测试效率。

[0093] 利用步骤四,可以给用户呈现完整且直观的测试报告。用户仅仅需要查看测试报告即可知道渠道测试的整体测试概况,同时又可以查看失败渠道测试任务的具体失败详情,可协助开发者及时快速地定位bug。

[0094] 在本实施例中,整个平台可以采用Server-Agent的分布式架构。图9是根据本发明实施例的Server-Agent的分布式架构示意图,如图9所示,采用分布式架构有利于横向扩展,当需要接入更多的测试设备时,可以通过扩展Agent节点来完成这一功能。用户在Web平台上创建渠道测试任务后,Web平台通过http协议将任务分发至Agent节点,Agent节点再通过任务执行模块调用各个测试设备执行渠道测试任务。任务完成后,Agent节点通过http协议将测试结果回发至Server。Server通过报告生成模块生成最终供用户查看的报告。整个平台的核心模块包括任务分发模块、任务执行模块、设备管理模块、数据回发模块、报告生成模块。

[0095] 各模块实现方式可以如下:

[0096] 任务分发模块:该模块主要用于将用户创建的渠道测试任务分发到各个Agent,其基本原理见步骤二,渠道测试任务创建与分发。该模块定时从各Agent获取各个测试设备的状态(空闲、忙碌)并为每个测试设备维护一个任务队列。其定时从Server的数据库中读取任务并通过http协议将任务分发至各个Agent。即Server端实时获取Agent上所接入的测试设备状态并为每个测试设备维护一个任务队列,将用户创建的渠道测试任务按要求分发至各Agent。

[0097] 任务执行模块:该模块主要用于接收来自Server的调度任务并执行渠道测试任务。同一个Agent可能并行执行多个渠道测试任务,该模块的核心功能是弹框点击、渠道登录测试和渠道支付测试。三者的具体实现原理为:对于Android原生的UI,通过UI控件的控

件类别(Class Name)、资源ID(Resource ID)、以及控件上的文字(Text)三个属性的正则匹配来寻找UI的位置,从而完成相应的点击、滑动操作。对于游戏内非Android原生的UI,通过图像匹配的方式来获取UI的位置,从而完成相应的点击、滑动操作。图像匹配采用模板卷积匹配和SIFT算法两种方式。

[0098] 设备管理模块:该模块主要是各个Agent用来管理与其连接的所有测试设备。Agent通过adb来与连接设备进行通信,获取在线设备参数、设备串号,同时维护设备状态:当设备接收测试任务时,设备状态置为忙碌;当设备完成测试任务时,设备状态置为空闲;当设备测试任务超时的时候,设备状态置为空闲;当已连接设备断开时,设备状态置为断开。

[0099] 数据回发模块:该模块主要将各Agent上的渠道测试结果采用http协议回发至Server并存入数据库。回发的渠道测试结果包括设备基本信息、测试时间信息、安装、启动、渠道登录、渠道支付信息。同时还包括整个测试任务执行过程中所产生的LOG文件以及测试设备截图。

[0100] 报告生成模块:该模块主要将回发至Server的测试结果信息进行统计处理,形成测试报告,通过web前端展示。利用web前端技术如jQuery,ajax,AngularJS以及echart等开源插件将测试结果以图形化的形式展现出来。用户既可以看到各渠道在各测试设备上的测试概况,同时也可以查看每个测试任务的详细数据,包括:测试任务的执行日志,测试任务每一步的执行情况及相应的截图。

[0101] 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。

[0102] 在本发明的上述实施例中,对各个实施例的描述都各有侧重,某个实施例中沒有详述的部分,可以参见其他实施例的相关描述。

[0103] 在本申请所提供的几个实施例中,应该理解到,所揭露的技术内容,可通过其它的方式实现。其中,以上所描述的装置实施例仅仅是示意性的,例如所述单元的划分,可以为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,单元或模块的间接耦合或通信连接,可以是电性或其它的形式。

[0104] 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单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0105] 另外,在本发明各个实施例中的各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理存在,也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现,也可以采用软件功能单元的形式实现。

[0106] 所述集成的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时,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,本发明的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或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一个存储介质中,包括若干指令用以使得一台计算机设备(可为个人计算机、服务器或者网络设备)执行本发明各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、只读存储器(ROM,Read-Only Memory)、随机存取存

储器 (RAM, Random Access Memory)、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

[0107] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

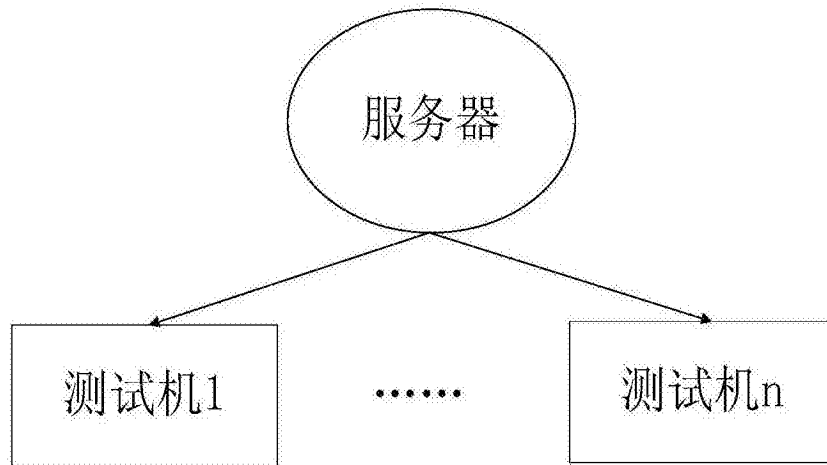


图1

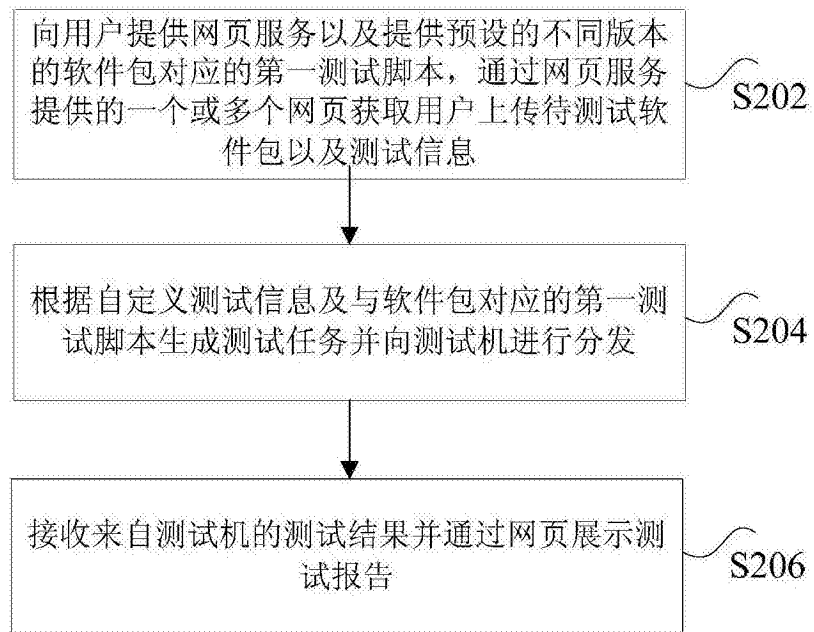


图2

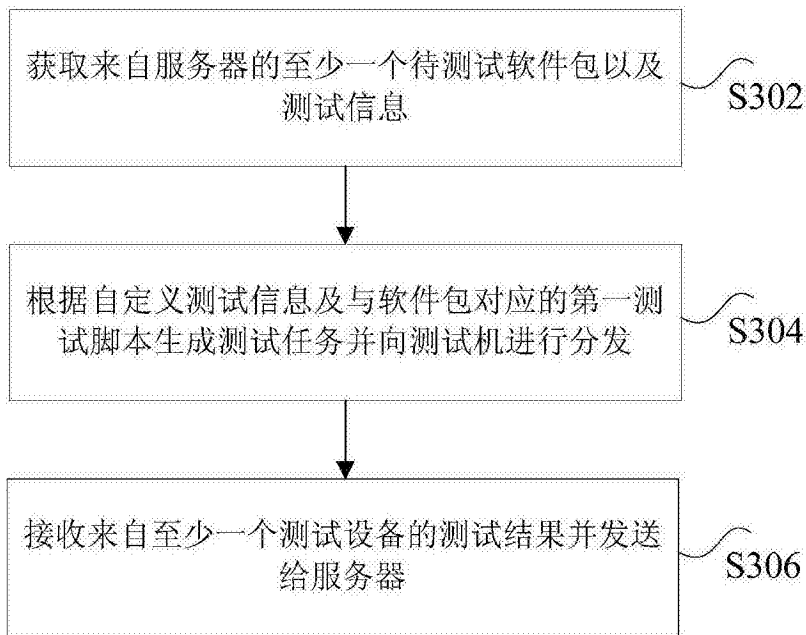


图3

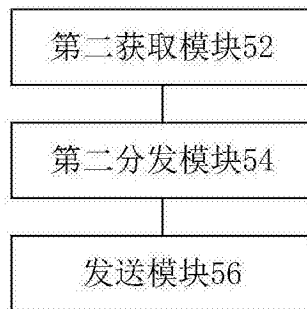


图5

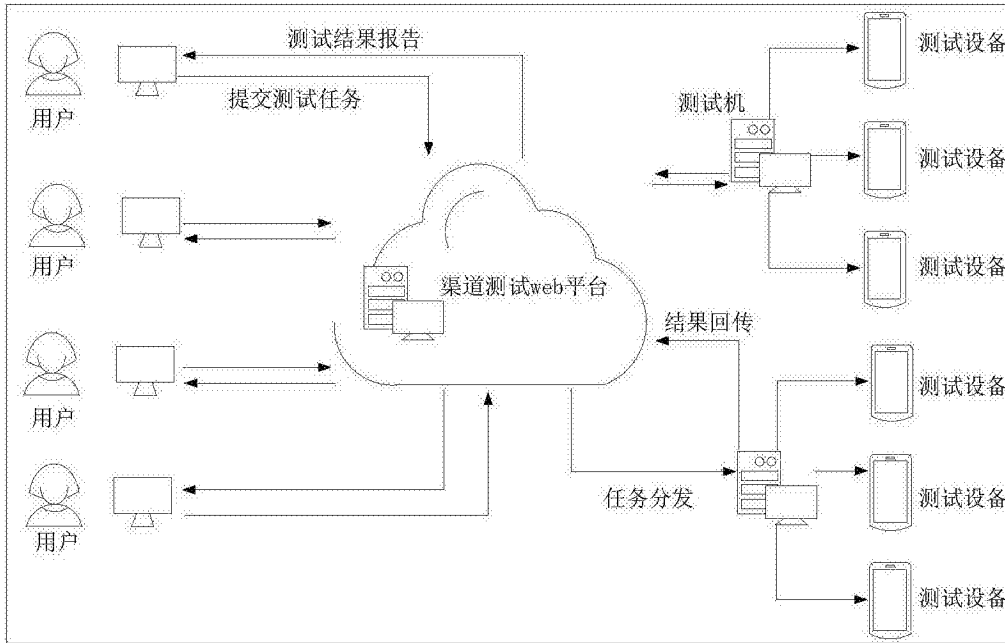


图6

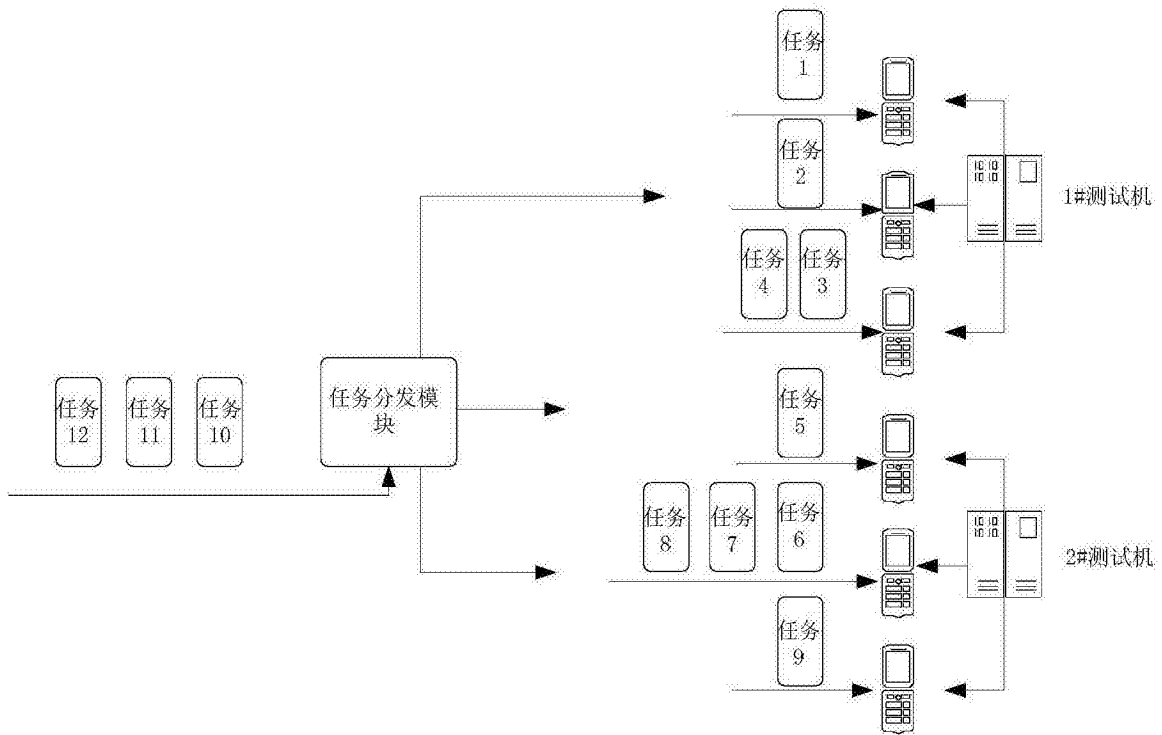


图7

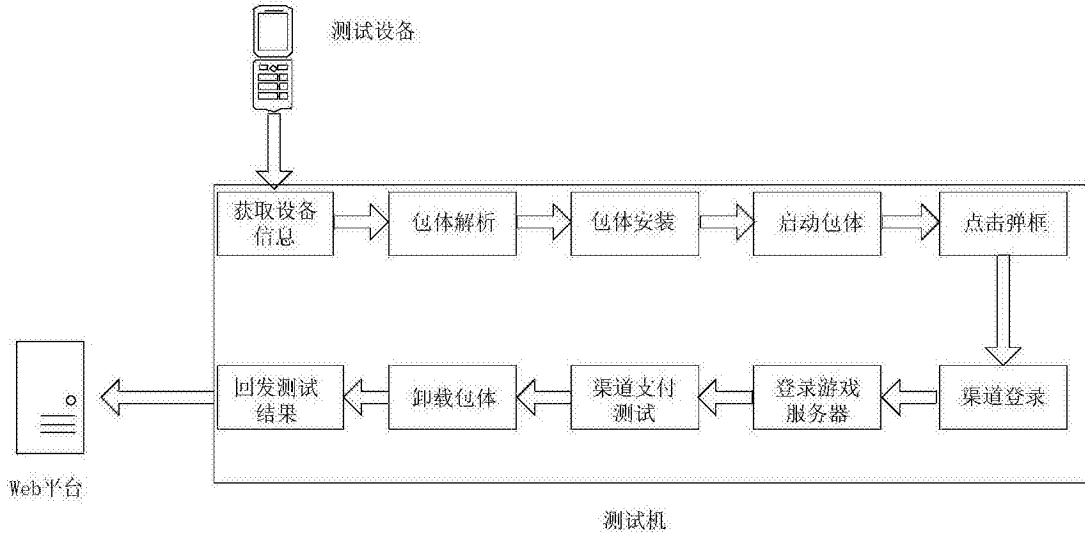


图8

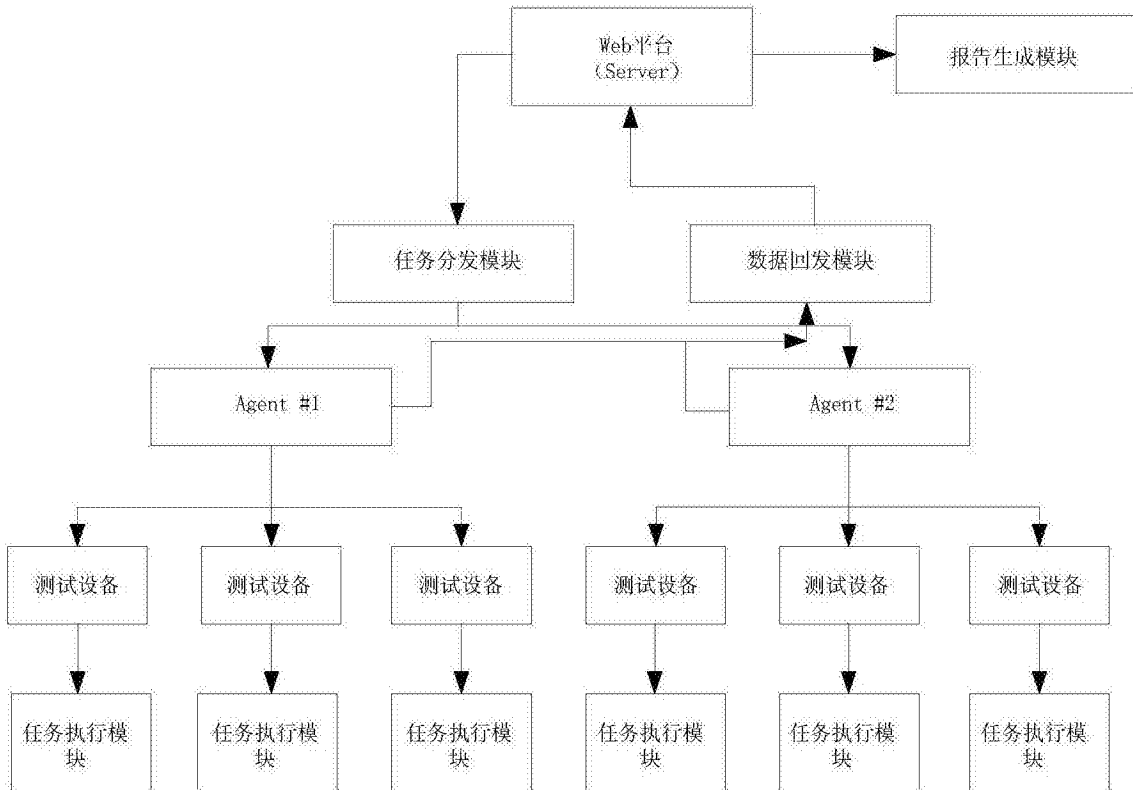


图9