



(12) 发明专利

(10) 授权公告号 CN 113688043 B

(45) 授权公告日 2022.07.26

(21) 申请号 202110984169.5

G06F 9/445 (2018.01)

(22) 申请日 2021.08.25

(56) 对比文件

(65) 同一申请的已公布的文献号
申请公布号 CN 113688043 A

US 9274935 B1,2016.03.01

US 9274935 B1,2016.03.01

CN 111752817 A,2020.10.09

(43) 申请公布日 2021.11.23

US 2014253574 A1,2014.09.11

CN 111767176 A,2020.10.13

(73) 专利权人 北京三快在线科技有限公司
地址 100080 北京市海淀区北四环西路9号
2106-030

CN 112241502 A,2021.01.19

US 2016147645 A1,2016.05.26

(72) 发明人 郑晓晨 高栋南 潘文杰 陈光辉
张亚南 赵敏

宋伟.基于流量分析的Android自动化测试
方案.《中国优秀硕士学位论文全文数据库 信息
科技辑》.2017,1138-1251.

(74) 专利代理机构 北京三高永信知识产权代理
有限责任公司 11138
专利代理师 谢冬寒

审查员 张雁琳

(51) Int. Cl.

G06F 11/36 (2006.01)

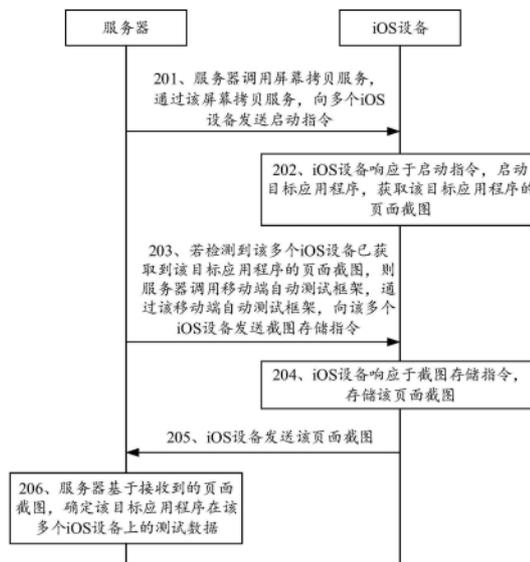
权利要求书4页 说明书19页 附图10页

(54) 发明名称

应用程序测试方法、装置、服务器、iOS设备
及介质

(57) 摘要

本申请提供了一种应用程序测试方法、装置、服务器、iOS设备及介质,属于计算机技术领域。本申请通过调用屏幕拷贝服务,以便通过屏幕拷贝服务向多个iOS设备发送启动指令,以指示多个iOS设备同时进行目标应用程序的启动,以及目标应用程序的页面截图的获取,进而在检测到各个iOS设备已获取到目标应用程序的截图时,调用移动端自动测试框架,从而通过移动端自动测试框架,向多个iOS设备发送截图存储指令,以便多个iOS设备同时进行页面截图的存储和传输,从而基于接收到的页面截图,确定目标应用程序在多个iOS设备上的测试数据,实现了同时对多个iOS设备上的目标应用程序进行测试,从而提高了目标应用程序的测试效率。



1. 一种应用程序测试方法,其特征在于,所述方法包括:

调用屏幕拷贝服务,通过所述屏幕拷贝服务,向多个iOS设备发送启动指令,所述屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能,所述启动指令用于指示启动所述目标应用程序,以及获取所述目标应用程序的页面截图;

若检测到所述多个iOS设备已获取到所述目标应用程序的页面截图,则调用移动端自动测试框架,通过所述移动端自动测试框架,向所述多个iOS设备发送截图存储指令,所述已获取到的目标应用程序的页面截图在iOS设备的缓存中,所述移动端自动测试框架用于提供应用程序测试功能,所述截图存储指令用于指示所述多个iOS设备对获取到的页面截图进行存储和传输,所述对获取到的页面截图进行存储包括:所述iOS设备运行iOS移动测试框架,将所述页面截图从缓存写入内存,所述iOS移动测试框架用于提供对所述iOS设备的远程控制功能;

基于接收到的页面截图,确定所述目标应用程序在所述多个iOS设备上的测试数据。

2. 根据权利要求1所述的方法,其特征在于,所述测试数据为启动耗时数据;

所述基于接收到的页面截图,确定所述目标应用程序在所述多个iOS设备上的测试数据,包括:

对于任一iOS设备,基于所述任一iOS设备对应的页面截图,确定应用启动起始帧和应用启动结束帧,所述应用启动起始帧为所述目标应用程序开始启动时所对应的页面截图,所述应用启动结束帧为所述目标应用程序启动完成时所对应的页面截图;

基于所述应用启动起始帧的时间戳和所述应用启动结束帧的时间戳,确定所述目标应用程序在所述任一iOS设备上的启动耗时数据。

3. 根据权利要求2所述的方法,其特征在于,所述方法还包括:

获取所述多个iOS设备的屏幕尺寸,其中,所述多个iOS设备的屏幕尺寸通过屏幕拷贝服务获取得到;

所述应用启动结束帧的确定过程,包括:

基于所述任一iOS设备对应的任一页面截图和所述任一iOS设备的屏幕尺寸,确定所述任一页面截图与所述任一iOS设备对应的其他页面截图的指纹数据的差异程度,所述指纹数据用于指示页面截图中的像素分布情况;

当所述差异程度小于第一预设阈值时,基于所述任一页面截图的像素值均值和其他页面截图的像素值均值,确定差异图像,所述差异图像为所述任一页面截图中像素值均值小于第二预设阈值的部分;

若所述差异图像的像素值标准差小于第三预设阈值,则将所述任一页面截图确定为所述应用启动结束帧。

4. 根据权利要求1所述的方法,其特征在于,所述调用屏幕拷贝服务,通过所述屏幕拷贝服务,向多个iOS设备发送启动指令之前,所述方法还包括:

读取页面配置文件,所述页面配置文件用于指示所述目标应用程序的各个页面的页面跳转顺序;

所述在所述多个iOS设备已获取到所述目标应用程序的页面截图的情况下,向所述多个iOS设备发送截图存储指令之后,所述方法还包括:

基于所述页面配置文件,向所述多个iOS设备发送页面跳转指令,所述页面跳转指令用

于指示所述多个iOS设备,按照所述页面配置文件所指示的页面跳转顺序,显示所述目标应用程序的各个页面。

5. 根据权利要求4所述的方法,其特征在于,所述测试数据包括页面加载耗时数据;

所述基于接收到的页面截图,确定所述目标应用程序在所述多个iOS设备上的测试数据,包括:

对于任一iOS设备,基于所述任一iOS设备对应的页面截图和所述任一iOS设备的屏幕尺寸,确定页面加载起始帧和页面加载结束帧,所述页面加载起始帧为所述目标应用程序在进行页面跳转前所对应的页面截图,所述页面加载结束帧为所述目标应用程序完成页面跳转时所对应的页面截图;

基于所述页面加载起始帧的时间戳和所述页面加载结束帧的时间戳,确定所述目标应用程序在所述任一iOS设备上的页面加载耗时数据。

6. 根据权利要求1所述的方法,其特征在于,所述方法还包括:

在接收到所述页面截图后,向所述多个iOS设备发送存储停止指令,所述存储停止指令用于指示所述多个iOS设备停止所述目标应用程序的页面截图的存储和传输。

7. 根据权利要求1所述的方法,其特征在于,所述调用屏幕拷贝服务,通过所述屏幕拷贝服务,向多个iOS设备发送启动指令之前,所述方法还包括:

开启定长线程池,所述定长线程池用于存储目标数量的iOS设备的截图获取任务,所述截图获取任务用于获取对应iOS设备中目标应用程序的页面截图。

8. 一种应用程序测试方法,其特征在于,所述方法包括:

响应于启动指令,启动目标应用程序,获取所述目标应用程序的页面截图,所述启动指令由服务器通过屏幕拷贝服务发送,所述屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能,其中,获取到的所述目标应用程序的页面截图在缓存中;

响应于截图存储指令,存储所述页面截图,所述截图存储指令由服务器通过移动端自动测试框架发送,所述移动端自动测试框架用于提供应用程序测试功能,所述存储页面截图包括:运行iOS移动测试框架,将所述页面截图从缓存写入内存,所述iOS移动测试框架用于提供对iOS设备的远程控制功能;

发送所述页面截图。

9. 根据权利要求8所述的方法,其特征在于,所述响应于截图存储指令,存储所述页面截图,包括:

启用iOS移动测试框架,通过所述iOS移动测试框架,响应于所述截图存储指令,存储所述页面截图。

10. 根据权利要求8所述的方法,其特征在于,所述响应于启动指令,启动目标应用程序,获取所述目标应用程序的页面截图之后,所述方法还包括:

采用队列结构缓存获取到的页面截图;

所述响应于截图存储指令,存储所述页面截图,包括:

响应于所述截图存储指令,对缓存中的页面截图进行封装;

存储封装后的页面截图。

11. 根据权利要求8所述的方法,其特征在于,所述发送所述页面截图,包括:

通过套接字接口,发送所述页面截图。

12. 根据权利要求8所述的方法,其特征在于,所述方法还包括:

响应于所述启动指令,获取iOS设备的屏幕尺寸;

发送所述屏幕尺寸;

其中,所述屏幕尺寸用于确定所述目标应用程序的应用启动结束帧和页面加载结束帧,所述应用启动结束帧为所述目标应用程序启动完成时所对应的页面截图,所述页面加载结束帧为所述目标应用程序完成页面跳转时所对应的页面截图。

13. 一种应用程序测试装置,其特征在于,所述装置包括:

调用模块,用于调用屏幕拷贝服务,所述屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能;

发送模块,用于通过所述屏幕拷贝服务,向多个iOS设备发送启动指令,所述启动指令用于指示启动所述目标应用程序,以及获取所述目标应用程序的页面截图;

所述调用模块,还用于若检测到所述多个iOS设备已获取到所述目标应用程序的页面截图,则调用移动端自动测试框架,所述已获取到的目标应用程序的页面截图在iOS设备的缓存中,所述移动端自动测试框架用于提供应用程序测试功能;

所述发送模块,还用于通过所述移动端自动测试框架,向所述多个iOS设备发送截图存储指令,所述截图存储指令用于指示所述多个iOS设备对获取到的页面截图进行存储和传输,所述对获取到的页面截图进行存储包括:所述iOS设备运行iOS移动测试框架,将所述页面截图从缓存写入内存,所述iOS移动测试框架用于提供对所述iOS设备的远程控制功能;

确定模块,用于基于接收到的页面截图,确定所述目标应用程序在所述多个iOS设备上的测试数据。

14. 一种应用程序测试装置,其特征在于,所述装置包括:

启动模块,用于响应于启动指令,启动目标应用程序,所述启动指令由服务器通过屏幕拷贝服务发送,所述屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能,其中,获取到的所述目标应用程序的页面截图在缓存中;

获取模块,用于获取所述目标应用程序的页面截图;

存储模块,用于响应于截图存储指令,存储所述页面截图,所述截图存储指令由服务器通过移动端自动测试框架发送,所述移动端自动测试框架用于提供应用程序测试功能,所述存储页面截图包括:运行iOS移动测试框架,将所述页面截图从缓存写入内存,所述iOS移动测试框架用于提供对iOS设备的远程控制功能;

发送模块,用于发送所述页面截图。

15. 一种服务器,其特征在于,所述服务器包括一个或多个处理器和一个或多个存储器,所述一个或多个存储器中存储有至少一条程序代码,所述程序代码由所述一个或多个处理器加载并执行以实现如权利要求1至权利要求7中任一项所述的应用程序测试方法所执行的操作。

16. 一种iOS设备,其特征在于,所述iOS设备包括一个或多个处理器和一个或多个存储器,所述一个或多个存储器中存储有至少一条程序代码,所述程序代码由所述一个或多个处理器加载并执行以实现如权利要求8至权利要求12中任一项所述的应用程序测试方法所执行的操作。

17. 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中存储有至少一

条程序代码,所述程序代码由处理器加载并执行以实现如权利要求1至权利要求7中任一项所述的应用程序测试方法所执行的操作;或如权利要求8至权利要求12中任一项所述的应用程序测试方法所执行的操作。

应用程序测试方法、装置、服务器、iOS设备及介质

技术领域

[0001] 本申请涉及计算机技术领域,特别涉及一种应用程序测试方法、装置、服务器、iOS设备及介质。

背景技术

[0002] 随着计算机技术的发展,各种类型的应用程序(Application,APP)层出不穷。而在APP上线前,往往要经过启动耗时测速、页面加载速度测试等专项测试过程,以实现APP运行情况的测试,从而及时发现APP存在的不足,并针对所发现的不足加以改进,以保证所上线的APP能够满足用户的使用需求。

[0003] 目前,在测试APP在iOS设备上的运行情况时,主要是通过Mac服务器上触发xrecord-QuickTime指令,从而触发iOS设备进行录屏,再在Mac服务器上触发ffmpeg指令,以触发iOS设备对录制得到的视频进行分帧,并基于分帧后的结果获取相应的截图,进而根据获取到的截图确定APP的启动速度、页面加载速度等。

[0004] 在上述实现过程中,一台Mac服务器采用上述方式,仅能同时对一台iOS设备上的APP进行测试,从而导致测试效率较低。

发明内容

[0005] 本申请实施例提供了一种应用程序测试方法、装置、服务器、iOS设备及介质,能够通过多台iOS设备上的APP进行测试,从而提高测试效率。本申请的技术方案如下:

[0006] 一方面,提供了一种应用程序测试方法,该方法包括:

[0007] 调用屏幕拷贝服务,通过该屏幕拷贝服务,向多个iOS设备发送启动指令,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能,该启动指令用于指示启动该目标应用程序,以及获取该目标应用程序的页面截图;

[0008] 若检测到该多个iOS设备已获取到该目标应用程序的页面截图,则调用移动端自动测试框架,通过该移动端自动测试框架,向该多个iOS设备发送截图存储指令,该移动端自动测试框架用于提供应用程序测试功能,该截图存储指令用于指示该多个iOS设备对获取到的页面截图进行存储和传输;

[0009] 基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据。

[0010] 在一种可能的实现方式中,该测试数据为启动耗时数据;

[0011] 基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据,包括:

[0012] 对于任一iOS设备,基于该任一iOS设备对应的页面截图,确定应用启动起始帧和应用启动结束帧,该应用启动起始帧为该目标应用程序开始启动时所对应的页面截图,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图;

[0013] 基于该应用启动起始帧的时间戳和该应用启动结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的启动耗时数据。

[0014] 在一种可能的实现方式中,该方法还包括:

[0015] 获取该多个iOS设备的屏幕尺寸,其中,该多个iOS设备的屏幕尺寸通过屏幕拷贝服务获取得到;

[0016] 该应用启动结束帧的确定过程,包括:

[0017] 基于该任一iOS设备对应的任一页面截图和该任一iOS设备的屏幕尺寸,确定该任一页面截图与该任一iOS设备对应的其他页面截图的指纹数据的差异程度,该指纹数据用于指示页面截图中的像素分布情况;

[0018] 当该差异程度小于第一预设阈值时,基于该任一页面截图的像素值均值和其他页面截图的像素值均值,确定差异图像,该差异图像为该任一页面截图中像素值均值小于第二预设阈值的部分;

[0019] 若该差异图像的像素值标准差小于第三预设阈值,则将该任一页面截图确定为该应用启动结束帧。

[0020] 在一种可能的实现方式中,调用屏幕拷贝服务,通过该屏幕拷贝服务,向多个iOS设备发送启动指令之前,该方法还包括:

[0021] 读取页面配置文件,该页面配置文件用于指示该目标应用程序的各个页面的页面跳转顺序;

[0022] 在该多个iOS设备已获取到该目标应用程序的页面截图的情况下,向该多个iOS设备发送截图存储指令之后,该方法还包括:

[0023] 基于该页面配置文件,向该多个iOS设备发送页面跳转指令,该页面跳转指令用于指示该多个iOS设备,按照该页面配置文件所指示的页面跳转顺序,显示该目标应用程序的各个页面。

[0024] 在一种可能的实现方式中,该测试数据包括页面加载耗时数据;

[0025] 基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据,包括:

[0026] 对于任一iOS设备,基于该任一iOS设备对应的页面截图和该任一iOS设备的屏幕尺寸,确定页面加载起始帧和页面加载结束帧,该页面加载起始帧为该目标应用程序在进行页面跳转前所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图;

[0027] 基于该页面加载起始帧的时间戳和该页面加载结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的页面加载耗时数据。

[0028] 在一种可能的实现方式中,该方法还包括:

[0029] 在接收到该页面截图后,向该多个iOS设备发送存储停止指令,该存储停止指令用于指示该多个iOS设备停止该目标应用程序的页面截图的存储和传输。

[0030] 在一种可能的实现方式中,调用屏幕拷贝服务,通过该屏幕拷贝服务,向多个iOS设备发送启动指令之前,该方法还包括:

[0031] 开启定长线程池,该定长线程池用于存储目标数量的iOS设备的截图获取任务,该截图获取任务用于获取对应iOS设备中目标应用程序的页面截图。

[0032] 一方面,提供了一种应用程序测试方法,该方法包括:

[0033] 响应于启动指令,启动目标应用程序,获取该目标应用程序的页面截图,该启动指

令由服务器通过屏幕拷贝服务发送,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能;

[0034] 响应于截图存储指令,存储该页面截图,该截图存储指令由服务器通过移动端自动测试框架发送,该移动端自动测试框架用于提供应用程序测试功能;

[0035] 发送该页面截图。

[0036] 在一种可能的实现方式中,响应于截图存储指令,存储该页面截图,包括:

[0037] 启用iOS移动测试框架,通过该iOS移动测试框架,响应于该截图存储指令,存储该页面截图。

[0038] 在一种可能的实现方式中,响应于启动指令,启动目标应用程序,获取该目标应用程序的页面截图之后,该方法还包括:

[0039] 采用队列结构缓存获取到的页面截图;

[0040] 该响应于截图存储指令,存储该页面截图,包括:

[0041] 响应于该截图存储指令,对缓存中的页面截图进行封装;

[0042] 存储封装后的页面截图。

[0043] 在一种可能的实现方式中,发送该页面截图,包括:

[0044] 通过套接字接口,发送该页面截图。

[0045] 在一种可能的实现方式中,该方法还包括:

[0046] 响应于该启动指令,获取iOS设备的屏幕尺寸;

[0047] 发送该屏幕尺寸;

[0048] 其中,该屏幕尺寸用于确定该目标应用程序的应用启动结束帧和页面加载结束帧,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图。

[0049] 一方面,提供了一种应用程序测试装置,该装置包括:

[0050] 调用模块,用于调用屏幕拷贝服务,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能;

[0051] 发送模块,用于通过该屏幕拷贝服务,向多个iOS设备发送启动指令,该启动指令用于指示启动该目标应用程序,以及获取该目标应用程序的页面截图;

[0052] 该调用模块,还用于若检测到该多个iOS设备已获取到该目标应用程序的页面截图,则调用移动端自动测试框架,该移动端自动测试框架用于提供应用程序测试功能;

[0053] 该发送模块,还用于通过该移动端自动测试框架,向该多个iOS设备发送截图存储指令,该截图存储指令用于指示该多个iOS设备对获取到的页面截图进行存储和传输;

[0054] 确定模块,用于基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据。

[0055] 在一种可能的实现方式中,该测试数据为启动耗时数据;

[0056] 该确定模块,用于对于任一iOS设备,基于该任一iOS设备对应的页面截图,确定应用启动起始帧和应用启动结束帧,该应用启动起始帧为该目标应用程序开始启动时所对应的页面截图,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图;基于该应用启动起始帧的时间戳和该应用启动结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的启动耗时数据。

[0057] 在一种可能的实现方式中,该装置还包括:

[0058] 获取模块,用于获取该多个iOS设备的屏幕尺寸,其中,该多个iOS设备的屏幕尺寸通过屏幕拷贝服务获取得到;

[0059] 该应用启动结束帧的确定过程,包括:

[0060] 基于该任一iOS设备对应的任一页面截图和该任一iOS设备的屏幕尺寸,确定该任一页面截图与该任一iOS设备对应的其他页面截图的指纹数据的差异程度,该指纹数据用于指示页面截图中的像素分布情况;

[0061] 当该差异程度小于第一预设阈值时,基于该任一页面截图的像素值均值和其他页面截图的像素值均值,确定差异图像,该差异图像为该任一页面截图中像素值均值小于第二预设阈值的部分;

[0062] 若该差异图像的像素值标准差小于第三预设阈值,则将该任一页面截图确定为该应用启动结束帧。

[0063] 在一种可能的实现方式中,该装置还包括:

[0064] 读取模块,用于读取页面配置文件,该页面配置文件用于指示该目标应用程序的各个页面的页面跳转顺序;

[0065] 该发送模块,还用于基于该页面配置文件,向该多个iOS设备发送页面跳转指令,该页面跳转指令用于指示该多个iOS设备,按照该页面配置文件所指示的页面跳转顺序,显示该目标应用程序的各个页面。

[0066] 在一种可能的实现方式中,该测试数据包括页面加载耗时数据;

[0067] 该确定模块,用于对于任一iOS设备,基于该任一iOS设备对应的页面截图和该任一iOS设备的屏幕尺寸,确定页面加载起始帧和页面加载结束帧,该页面加载起始帧为该目标应用程序在进行页面跳转前所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图;基于该页面加载起始帧的时间戳和该页面加载结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的页面加载耗时数据。

[0068] 在一种可能的实现方式中,该发送模块,还用于在接收到该页面截图后,向该多个iOS设备发送存储停止指令,该存储停止指令用于指示该多个iOS设备停止该目标应用程序的页面截图的存储和传输。

[0069] 在一种可能的实现方式中,该装置还包括:

[0070] 开启模块,用于开启定长线程池,该定长线程池用于存储目标数量的iOS设备的截图获取任务,该截图获取任务用于获取对应iOS设备中目标应用程序的页面截图。

[0071] 一方面,提供了一种应用程序测试装置,该装置包括:

[0072] 启动模块,用于响应于启动指令,启动目标应用程序,该启动指令由服务器通过屏幕拷贝服务发送,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能;

[0073] 获取模块,用于获取该目标应用程序的页面截图;

[0074] 存储模块,用于响应于截图存储指令,存储该页面截图,该截图存储指令由服务器通过移动端自动测试框架发送,该移动端自动测试框架用于提供应用程序测试功能;

[0075] 发送模块,用于发送该页面截图。

[0076] 在一种可能的实现方式中,该存储模块,用于启用iOS移动测试框架,通过该iOS移动测试框架,响应于该截图存储指令,存储该页面截图。

- [0077] 在一种可能的实现方式中,该装置还包括:
- [0078] 缓存模块,用于采用队列结构缓存获取到的页面截图;
- [0079] 该存储模块,用于响应于该截图存储指令,对缓存中的页面截图进行封装;存储封装后的页面截图。
- [0080] 在一种可能的实现方式中,该发送模块,用于通过套接字接口,发送该页面截图。
- [0081] 在一种可能的实现方式中,该获取模块,还用于响应于该启动指令,获取iOS设备的屏幕尺寸;
- [0082] 该发送模块,还用于发送该屏幕尺寸;
- [0083] 其中,该屏幕尺寸用于确定该目标应用程序的应用启动结束帧和页面加载结束帧,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图。
- [0084] 一方面,提供了一种服务器,该服务器包括一个或多个处理器和一个或多个存储器,该一个或多个存储器中存储有至少一条程序代码,该程序代码由该一个或多个处理器加载并执行以实现该应用程序测试方法所执行的操作。
- [0085] 一方面,提供了一种iOS设备,该iOS设备包括一个或多个处理器和一个或多个存储器,该一个或多个存储器中存储有至少一条程序代码,该程序代码由该一个或多个处理器加载并执行以实现该应用程序测试方法所执行的操作。
- [0086] 一方面,提供了一种计算机可读存储介质,该计算机可读存储介质中存储有至少一条程序代码,该程序代码由处理器加载并执行以实现该应用程序测试方法所执行的操作。
- [0087] 一方面,提供了一种计算机程序产品,该计算机程序产品包括计算机代码,该计算机程序代码由处理器加载并执行以实现该应用程序测试方法所执行的操作。
- [0088] 通过调用屏幕拷贝服务,以便通过屏幕拷贝服务向多个iOS设备发送启动指令,以指示多个iOS设备同时进行目标应用程序的启动,以及目标应用程序的页面截图的获取,进而在检测到各个iOS设备已获取到目标应用程序的截图时,调用移动端自动测试框架,从而通过移动端自动测试框架,向多个iOS设备发送截图存储指令,以便多个iOS设备同时进行页面截图的存储和传输,从而基于接收到的页面截图,确定目标应用程序在多个iOS设备上的测试数据,实现了同时对多个iOS设备上的目标应用程序进行测试,从而提高了目标应用程序的测试效率。

附图说明

- [0089] 为了更清楚地说明本申请实施例中的技术方案,下面将对实施例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申请的一些实施例,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获得其他的附图。
- [0090] 图1是本申请实施例提供的一种应用程序测试方法的实施环境示意图;
- [0091] 图2是本申请实施例提供的一种应用程序测试方法的流程图;
- [0092] 图3是本申请实施例提供的一种页面截图示意图;
- [0093] 图4是本申请实施例提供的一种应用程序测试方法的流程图;

- [0094] 图5是本申请实施例提供的一种应用程序测试方法的流程图；
- [0095] 图6是本申请实施例提供的一种页面截图示意图；
- [0096] 图7是本申请实施例提供的一种应用程序测试方法的流程图；
- [0097] 图8是本申请实施例提供的一种应用程序测试方法的流程图；
- [0098] 图9是本申请实施例提供的一种应用程序测试装置的结构示意图；
- [0099] 图10是本申请实施例提供的一种应用程序测试装置的结构示意图；
- [0100] 图11是本申请实施例提供的一种iOS设备的结构示意图；
- [0101] 图12是本申请实施例提供的一种服务器的结构示意图。

具体实施方式

[0102] 为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

[0103] 图1是本申请实施例提供的一种应用程序测试方法的实施环境示意图，参见图1，该实施环境包括：服务器101和iOS设备102。

[0104] 服务器101为一台服务器、多台服务器、云计算平台和虚拟化中心中的至少一种。服务器101和iOS设备102通过有线或无线通信方式进行连接，本申请实施例对此不加以限定。服务器101能够远程控制iOS设备102启动目标应用程序，以及远程控制iOS设备102实现目标应用程序中的页面跳转，并获取iOS设备102在程序启动或页面跳转过程中的页面截图，从而基于获取到的页面截图确定目标应用程序在iOS设备102上的测试数据。可选地，上述服务器的数量更多或更少，本申请实施例对此不加以限定。当然，在更多可能的实现方式中，服务器101还包括其他功能服务器，以便提供更全面且多样化的服务。

[0105] iOS设备102为搭载有iOS操作系统的智能手机、平板电脑、智能手表、台式电脑、笔记本电脑等。iOS设备102和服务器101通过有线或无线通信方式进行连接，本申请实施例对此不加以限定。iOS设备102能够在服务器101的远程控制下，实现目标应用程序的启动以及目标应用程序中的页面跳转，并在目标应用程序的启动过程和页面跳转过程中，获取目标应用程序的页面截图，从而将获取到的页面截图发送给服务器101。

[0106] iOS设备102泛指多个iOS设备中的一个，本申请实施例仅以iOS设备102来举例说明。本领域技术人员能够知晓，在更多可能的实现方式中，上述iOS设备的数量更多或更少。比如上述iOS设备仅为几个，或者上述iOS设备为几十个或几百个，或者更多数量，本申请实施例对iOS设备102的数量和设备类型均不加以限定。

[0107] 图2是本申请实施例提供的一种应用程序测试方法的流程图，参见图2，该方法包括：

[0108] 201、服务器调用屏幕拷贝服务，通过该屏幕拷贝服务，向多个iOS设备发送启动指令，该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能，该启动指令用于指示启动该目标应用程序，以及获取该目标应用程序的页面截图。

[0109] 其中，该服务器为苹果电脑服务器(Mac Server)，多个iOS设备均能接入该服务器，可选地，服务器与各个iOS设备之间通过Usbmuxd连接，或者，服务器与各个iOS设备之间通过其他有线或无线的连接方式进行连接，本申请实施例对此不加以限定。该服务器上存储有用于进行应用程序(Application, APP)测试的测试脚本，该测试脚本包括用于指示测

试过程中待执行操作的一系列指令,以便服务器在读取该测试脚本后,能够基于该测试脚本,来对APP进行测试。

[0110] 在一种可能的实现方式中,该测试脚本由开发人员编写完成后上传至服务器,服务器在获取到开发人员上传的测试脚本后,基于获取到的测试脚本,来进行后续的测试过程,也即是,基于获取到的测试脚本,执行步骤201至步骤206。

[0111] 在测试过程中,该服务器能够提供多种类型的服务,如屏幕拷贝服务 (Minicap Server),而不同类型的服务对应于不同的线程,以便通过调用某种类型的服务,来启动相应的线程,进而提供相应的功能,以保证测试过程的顺利进行。例如,通过该屏幕拷贝服务,即能调用相应的线程,从而基于AVFoundation框架和Core Media库,来进行页面截图的获取。

[0112] 在一种可能的实现方式中,服务器调用该屏幕拷贝服务,通过该屏幕拷贝服务,调用高速截图线程,通过该高速截图线程,服务器即能向iOS设备发送指令,以便对iOS设备上的目标应用程序进行初始化,还能模拟iOS设备上的一些点击操作,并远程控制iOS设备进行截图。例如,服务器通过该屏幕拷贝服务,向iOS设备发送启动指令,以模拟对目标应用程序的图标点击操作,从而实现远程控制目标应用程序的启动,以及目标应用程序的页面截图的获取。

[0113] 其中,不同的iOS设备均对应有一个高速截图线程,而每个高速截图线程对应于一个截图端口,也即是,每个iOS设备对应有一个截图端口,以便每个iOS设备在与服务器交互时,均能通过相应的截图端口来进行信息或指令的传输,提高传输效率。

[0114] 在更多可能的实现方式中,服务器还能通过该屏幕拷贝服务,获取多个iOS设备的屏幕尺寸,该屏幕尺寸用于确定该目标应用程序的应用启动结束帧和页面加载结束帧,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图,以便后续基于该屏幕尺寸,来确定目标应用程序在各个iOS设备上的测试数据,具体过程参见下述步骤206,此处不再赘述。

[0115] 可选地,服务器在执行该步骤201之前,开启定长线程池,该定长线程池用于存储目标数量的iOS设备的截图获取任务,该截图获取任务用于获取对应iOS设备中目标应用程序的页面截图。

[0116] 在更多可能的实现方式中,服务器在其他时机开启该定长线程池,例如,在执行该步骤201后,再开启定长线程池,或者,在执行该步骤201的同时,开启定长线程池,等等,本申请实施例对此不加以限定。

[0117] 其中,该服务器中预先存储有该定长线程池的大小,该定长线程池的大小是由测试人员通过服务器预先设置好的,或者,该定长线程池的大小通过其他方式获取得到,本申请实施例对此不加以限定。该定长线程池的大小为任意正整数值,例如,该定长线程池的大小为10,也即是,服务器允许10个线程同时运行,也即是,服务器允许10个iOS设备同时进行页面截图的传输。

[0118] 通过开启定长线程池,以启动相应数量的高速截图线程,从而获取相应数量的iOS设备的页面截图。而通过设置定长线程池的大小,使得服务器同时运行的高速截图线程的数量得到限制,减少服务器的处理压力,从而能够确保服务器的稳定运行。

[0119] 202、iOS设备响应于启动指令,启动目标应用程序,获取该目标应用程序的页面截

图。

[0120] 在一种可能的实现方式中，iOS设备在获取到目标应用程序的页面截图后，采用队列结构缓存获取到的页面截图，也即是，将获取到的页面截图存储到队列中。可选地，采用其他存储结构来缓存获取到的页面截图，本申请实施例对此不加以限定。

[0121] 203、若检测到该多个iOS设备已获取到该目标应用程序的页面截图，则服务器调用移动端自动测试框架，通过该移动端自动测试框架，向该多个iOS设备发送截图存储指令，该移动端自动测试框架用于提供应用程序测试功能，该截图存储指令用于指示该多个iOS设备对获取到的页面截图进行存储和传输。

[0122] 在一种可能的实现方式中，各个iOS设备在获取到目标应用程序的页面截图后，向服务器发送获取完成信息，该获取完成信息用于指示iOS设备已完成页面截图的获取，服务器在接收到任一个iOS设备发送的获取完成信息后，即可确定该任一个iOS设备已获取到该目标应用程序的页面截图，从而实现对各个iOS设备是否已经获取到目标应用程序的页面截图的检测。

[0123] 上述仅为一种示例性的检测方式，在更多可能的实现方式中，采用其他方式来进行检测，本申请实施例对此不加以限定。

[0124] 其中，该移动端自动测试框架(Appium)能够自动化iOS设备上的应用程序，从而实现了对iOS设备的远程控制的自动化。

[0125] 在一种可能的实现方式中，服务器在检测到多个iOS设备已获取到目标应用程序的页面截图后，调用移动端自动测试框架，以便通过移动端自动测试框架，调用图片组装存储线程，从而通过图片组装存储线程，向多个iOS设备发送截图存储指令，实现下述步骤204和步骤205中页面截图的存储和传输过程。

[0126] 可选地，各个iOS设备获取到目标应用程序的页面截图的时间不同，相应地，服务器检测到各个iOS设备已获取到目标应用程序的页面截图的过程不同时发生，则服务器每检测到一个iOS设备已获取到目标应用程序的页面截图，即向这个iOS设备发送截图存储指令。例如，对于该多个iOS设备中的第一iOS设备和第二iOS设备，服务器在接收到第一iOS设备的获取完成信息后，即可确定第一iOS设备已获取到目标应用程序的页面截图，则通过移动端自动测试框架，向第一iOS设备发送截图存储指令。而若在向第一iOS设备发送截图存储指令后，服务器接收到了第二iOS设备的获取完成信息，则服务器即可确定第二iOS设备已获取到目标应用程序的页面截图，此时，通过移动端自动测试框架，向第二iOS设备发送截图存储指令。

[0127] 204、iOS设备响应于截图存储指令，存储该页面截图。

[0128] 在一种可能的实现方式中，iOS设备启用iOS移动测试框架，通过该iOS移动测试框架，响应于该截图存储指令，存储该页面截图。

[0129] 其中，iOS移动测试框架(WebDriverAgent,WDA)已预先安装在各个iOS设备上，通过iOS移动测试框架即能实现对iOS设备的远程控制。

[0130] 存储该页面截图的过程包括：iOS设备响应于该截图存储指令，对缓存中的页面截图进行封装，进而存储封装后的页面截图。其中，iOS设备在存储页面截图时，将封装后的页面截图从缓存中写入内存，从而实现页面截图的存储。

[0131] 可选地，iOS设备的内存采用队列结构存储页面截图，或者，iOS设备的内存采用其

他存储结构来存储页面截图,本申请实施例对此不加以限定。

[0132] 其中,页面截图的封装过程包括:将页面截图转化为能够被iOS设备和服务器识别的二进制数据,也即是,帧(Frame)数据,并为Frame数据封装头(Header)数据,从而实现页面截图的封装。

[0133] 上述过程以先对页面截图进行封装,进而存储封装后的页面截图为例来进行说明,在另一种可能的实现方式中,iOS设备响应于该截图存储指令,将缓存中未封装的页面截图直接写入内存,从而实现页面截图的存储。

[0134] 205、iOS设备发送该页面截图。

[0135] 在一种可能的实现方式中,若iOS设备中存储的是封装后的页面截图,则iOS设备通过套接字(Socket)接口,发送封装后的页面截图。

[0136] 在另一种可能的实现方式中,若iOS设备中存储的是未封装的页面截图,则iOS设备在发送页面截图之前,先对内存中所存储的页面截图进行封装,进而通过套接字接口,发送封装后的页面截图。

[0137] 通过采用套接字接口来进行页面截图的传输,能够提高页面截图的传输速度,从而提高目标应用程序的测试速度。在更多可能的实现方式中,采用其他协议来进行页面截图的传输,本申请实施例对此不加以限定。

[0138] 其中,上述步骤204和步骤205的顺序并不构成对这两个步骤的执行顺序的限定,在一种可能的实现方式中,iOS设备存储完所有的页面截图后,再通过步骤205逐个进行页面截图的发送;在另一种可能的实现方式中,iOS设备每存储完一个页面截图,即将已存储的页面截图发送给服务器,也即是,iOS设备实时进行页面截图的存储与发送,本申请实施例对具体采用哪种执行顺序不加以限定。

[0139] 在上述步骤203至步骤205中,通过引入截图存储指令,在未向iOS设备发送截图存储指令之前,iOS设备仅完成了页面截图的获取和缓存,但获取到的页面截图并未写入内存,在向iOS设备发送截图存储指令后,iOS设备响应于截图存储指令,将获取到的页面截图从缓存写入内存,从而实现页面截图的存储,并将已存储的页面截图发送给服务器,以便服务器继续执行下述步骤206。

[0140] 206、服务器基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据。

[0141] 可选地,服务器在接收页面截图时,采用队列结构存储接收到的页面截图,在采用队列结构存储页面截图时,每接收到一个页面截图,就会将接收到的页面截图存储至队尾,而在页面截图存储完成后,才会继续对新接收到的页面截图进行存储,从而能够有效减少传输过快造成的处理不及时,提高页面截图的存储效率。

[0142] 在一种可能的实现方式中,服务器关联有一个页面截图数据库,用于存储接收到的页面截图,以便服务器在接收到页面截图后,通过该页面截图数据库对接收到的页面截图进行存储。在该页面截图数据库中,不同的测试类型对应有不同的存储区域,在存储接收到的页面截图时,服务器基于测试脚本所执行的指令对应的测试类型,将接收到的页面截图存储至对应的存储区域中。例如,不同的测试类型对应有不同的文件夹,则服务器基于测试脚本所执行的指令对应的测试类型,将接收到的页面截图存储至对应的文件夹中。参见图3,图3是本申请实施例提供的一种页面截图示意图,图3所示的服务器包括多个存储区域

(也即是文件夹),用于存储不同测试类型的页面截图。

[0143] 可选地,服务器在接收到该页面截图后,向该多个iOS设备发送存储停止指令,该存储停止指令用于指示该多个iOS设备停止该目标应用程序的页面截图的存储和传输。

[0144] 通过向多个iOS设备发送存储停止指令,以便iOS设备响应于接收到的截图存储指令,通过目标应用程序的页面截图的存储和传输,减少iOS设备的处理压力,相应地,服务器也无需接收和存储页面截图,减少服务器的处理压力。

[0145] 其中,该测试数据为启动耗时数据,相应地,该测试数据的确定过程包括:对于任一iOS设备,服务器基于该任一iOS设备对应的页面截图,确定应用启动起始帧和应用启动结束帧,该应用启动起始帧为该目标应用程序开始启动时所对应的页面截图,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图;基于该应用启动起始帧的时间戳和该应用启动结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的启动耗时数据。

[0146] 其中,基于应用启动起始帧的时间戳和该应用启动结束帧的时间戳,确定启动耗时数据的过程包括:确定应用启动起始帧的时间戳和应用启动结束帧的时间戳对应的时间间隔,作为该启动耗时数据,也即是,确定应用启动结束帧的时间戳与应用启动起始帧的时间戳之间的差值,作为该启动耗时数据。

[0147] 该应用启动起始帧的确定过程包括:服务器基于获取到的页面截图的时间戳,确定时间戳最小的页面截图,将时间戳最小的页面截图确定为该应用启动起始帧。

[0148] 该应用启动结束帧的确定过程包括:服务器基于该任一iOS设备对应的任一页面截图和该任一iOS设备的屏幕尺寸,确定该任一页面截图与该任一iOS设备对应的其他页面截图的指纹数据的差异程度,该指纹数据用于指示页面截图中的像素分布情况;当该差异程度小于第一预设阈值时,基于该任一页面截图的像素值均值和其他页面截图的像素值均值,确定差异图像,该差异图像为该任一页面截图中像素值均值小于第二预设阈值的部分;若该差异图像的像素值标准差小于第三预设阈值,则将该任一页面截图确定为该应用启动结束帧。其中,该第一预设阈值、第二预设阈值和第三预设阈值均为任意取值,本申请实施例对此不加以限定。

[0149] 其中,该任一页面截图的指纹数据的获取过程包括:服务器将该任一iOS设备的屏幕尺寸,确定为该任一页面截图的尺寸,不同的尺寸对应有不同的分辨率,从而基于该任一页面截图的尺寸,确定该任一页面截图的分辨率,以便基于分辨率确定该任一页面截图的像素点数量,从而基于确定出的像素点数量以及各个像素点的像素值,确定出一个特征值,作为该任一页面截图的指纹数据,每个页面截图对应于唯一的指纹数据。上述仅以任一页面截图的指纹数据的获取过程为例来进行说明,其他页面截图的指纹数据的获取过程与之同理,此处不再赘述。

[0150] 其他页面截图为时间戳在该任一页面截图的时间戳之前的n个页面截图,其中,n为任意正整数值,以n为3为例,该应用启动结束帧的确定过程包括:若该任一iOS设备的屏幕尺寸为5.8英寸,则服务器确定该任一页面截图的尺寸为5.8英寸,而5.8英寸对应的分辨率为 1125×2436 ,也即是,该任一页面截图在水平方向上有1125个像素点,在竖直方向上有2436个像素点,从而确定出了该任一页面截图的像素点数量,以便基于确定出的像素点数量以及各个像素点的像素值,确定任一页面截图的指纹数据,相应地,3个其他页面截图的指纹数据的确定过程与之同理;在确定出该任一页面截图和3个其他页面截图的指纹数据

后,确定该任一页面截图的指纹数据和3个其他页面截图的指纹数据的差异程度,当该差异程度小于第一预设阈值时,确定该任一页面截图和3个其他页面截图中的像素值均值,将该任一页面截图中像素值均值小于第二预设阈值的部分,确定为差异图像;进而确定该差异图像的像素值标准差,若确定出的像素值标准差小于第三预设阈值,则将该任一页面截图确定为该应用启动结束帧。

[0151] 上述过程仅以目标应用程序在一个iOS设备上的启动耗时数据的确定过程为例来进行说明,目标应用程序在其他iOS设备上的启动耗时数据的确定过程与上述过程同理,此处不再赘述。

[0152] 可选地,服务器还关联有一个测试结果数据库,用于存储目标应用程序在各个iOS设备上的测试数据,则服务器在确定出目标应用程序在各个iOS设备上的启动耗时数据后,将确定出的启动耗时数据存储至测试结果数据库中,以便后续能够从测试结果数据库中获取到相应的启动耗时数据。

[0153] 对于上述步骤201至步骤206,服务器与多个iOS设备的交互过程中iOS设备的处理过程是以一个iOS设备为例来进行说明的,其他iOS设备的处理过程与之同理,此处不再赘述。

[0154] 上述步骤201至步骤206的过程参见图4,图4是本申请实施例提供的一种应用程序测试方法的流程图,服务器启动测试脚本,来进行目标应用程序的测试,再调用Minicap Server,来进行APP初始化和线程管理,也即是,通过Minicap Server,来初始化iOS设备上的目标应用程序,并通过线程管理来调用高度截图线程,来触发iOS设备获取目标应用程序的页面截图,进而在服务器检测到iOS设备获取到页面截图后,服务器启动Appium,以开启图片组装存储线程,进而通过图片组装存储线程,来触发iOS设备进行页面截图的存储和发送,从而在接收到页面截图后,关闭截图存储线程,进而通过分析接收到的页面截图,确定测试结果,也即是启动耗时数据,并将确定出的启动耗时数据存储至数据库。

[0155] 本申请实施例所提供的方案,通过调用屏幕拷贝服务,以便通过屏幕拷贝服务向多个iOS设备发送启动指令,以指示多个iOS设备同时进行目标应用程序的启动,以及目标应用程序的页面截图的获取,进而在检测到各个iOS设备已获取到目标应用程序的截图时,调用移动端自动测试框架,从而通过移动端自动测试框架,向多个iOS设备发送截图存储指令,以便多个iOS设备同时进行页面截图的存储和传输,从而基于接收到的页面截图,确定目标应用程序在多个iOS设备上的测试数据,实现了同时对多个iOS设备上的目标应用程序进行测试,从而提高了目标应用程序的测试效率。此外,通过为每个iOS设备设置相应的截图端口,以减少因多个iOS设备使用同一端口产生冲突的问题的出现,保证测试过程的顺利进行。而且,通过本申请提供的方案,无需对iOS设备进行录屏,也无需对录制得到的视频进行分帧,测试过程更加简单,从而能够提高测试过程的执行效率。

[0156] 上述过程是以确定目标应用程序的启动耗时数据的过程为例来进行说明的,在更多可能的实现方式中,本申请所提供的方案,还能够用于确定目标应用程序的页面加载耗时数据,图5是本申请实施例提供的一种应用测试程序方法的流程图,参见图5,该方法包括:

[0157] 501、服务器读取页面配置文件,该页面配置文件用于指示该目标应用程序的各个页面的页面跳转顺序。

[0158] 其中,该页面配置(Scheme)文件包括用于指示测试过程中进行页面跳转操作的一系列指令,以便服务器在读取到该页面配置文件后,能够基于该页面配置文件,来远程控制目标应用程序进行页面跳转。

[0159] 该页面配置文件由测试人员编写完成后上传至服务器,服务器在获取到该页面配置文件后,对该页面配置文件进行校验,以确定该页面配置文件的格式是否为配置文件对应的格式,在页面配置文件校验通过时,继续执行后续步骤502至步骤509,否则直接结束此次测试过程。

[0160] 502、服务器调用屏幕拷贝服务,通过该屏幕拷贝服务,向多个iOS设备发送启动指令,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能,该启动指令用于指示启动该目标应用程序,以及获取该目标应用程序的页面截图。

[0161] 503、iOS设备响应于启动指令,启动目标应用程序,获取该目标应用程序的页面截图。

[0162] 504、若检测到该多个iOS设备已获取到该目标应用程序的页面截图,则服务器调用移动端自动测试框架,通过该移动端自动测试框架,向该多个iOS设备发送截图存储指令,该移动端自动测试框架用于提供应用程序测试功能,该截图存储指令用于指示该多个iOS设备对获取到的页面截图进行存储和传输。

[0163] 步骤502至步骤504与上述步骤201至步骤203同理,此处不再赘述。

[0164] 505、服务器基于该页面配置文件,向该多个iOS设备发送页面跳转指令,该页面跳转指令用于指示该多个iOS设备,按照该页面配置文件所指示的页面跳转顺序,显示该目标应用程序的各个页面。

[0165] 在一种可能的实现方式中,服务器在检测到该多个iOS设备已获取到目标应用程序的页面截图后,调用移动端自动测试框架,以便通过移动端自动测试框架,调用配置页面跳转线程,从而通过配置页面跳转线程,来远程控制iOS设备进行页面跳转。

[0166] 其中,不同的iOS设备均对应有一个配置页面跳转线程,而每个配置页面跳转线程对应于一个页面跳转端口,也即是,每个iOS设备对应有一个页面跳转端口,以便在服务器指示各个iOS设备进行页面跳转时,能通过相应的页面跳转端口来进行指令的传输,提高传输效率。

[0167] 该步骤505的标号并不构成对步骤505的执行顺序的限定,在更多可能的实现方式中,服务器在检测到该多个iOS设备已获取到目标应用程序的页面截图后,同时向该多个iOS设备发送该截图存储指令和页面跳转指令,本申请实施例对此不加以限定。

[0168] 506、iOS设备响应于页面跳转指令,跳转至相应的页面,并获取相应的页面截图。

[0169] 在一种可能的实现方式中,iOS设备响应于该页面跳转指令,开始页面跳转过程,在页面跳转过程中,实时获取页面跳转过程中的页面截图,并缓存获取到的页面截图,该页面截图的缓存过程参见上述步骤202,此处不再赘述。

[0170] 507、iOS设备响应于截图存储指令,存储获取到的页面截图。

[0171] 508、iOS设备发送获取到的页面截图。

[0172] 在上述步骤504至步骤508中,通过引入截图存储指令,在未向iOS设备发送截图存储指令之前,iOS设备仅完成了页面截图的获取和缓存,但获取到的页面截图并未写入内存,此时iOS设备上的WDA框架并未运行,而在向iOS设备发送截图存储指令后,iOS设备响应

于截图存储指令,开始运行WDA框架,将获取到的页面截图从缓存写入内存,从而实现页面截图的存储,并将已存储的页面截图发送给服务器,从而能够减少测试过程中Minicap Sever和WDA框架的冲突,保证测试过程的顺利进行。

[0173] 509、服务器基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据。

[0174] 步骤507至步骤509与上述步骤204至步骤206同理,此处不再赘述。

[0175] 其中,页面跳转过程中获取到的页面截图参见图6,图6是本申请实施例提供的一种页面截图示意图,图6所示的服务器包括多个存储区域(也即是文件夹),用于存储在页面跳转过程中,不同的页面跳转过程对应的页面截图。

[0176] 上述步骤501至步骤509的过程参见图7,图7是本申请实施例提供的一种应用程序测试方法的流程图,服务器启动测试脚本,来进行目标应用程序的测试,再调用Minicap Server,来进行APP初始化和线程管理,也即是,通过Minicap Server,来初始化iOS设备上的目标应用程序,并通过线程管理来调用高度截图线程,来触发iOS设备获取目标应用程序的页面截图,进而在服务器检测到iOS设备获取到页面截图后,服务器启动Appium,以开启图片组装存储线程,进而调用Scheme页面跳转线程,通过Scheme页面跳转线程,来远程控制iOS设备进行页面跳转,进而通过图片组装存储线程,来触发iOS设备进行页面截图的存储和发送,从而在接收到页面截图后,关闭截图存储线程,进而通过分析接收到的页面截图,确定测试结果,也即是页面加载耗时数据,并将确定出的页面加载耗时数据存储至数据库。

[0177] 本申请实施例提供的方案,通过读取页面配置文件,进而调用屏幕拷贝服务,以便通过屏幕拷贝服务向多个iOS设备发送启动指令,以指示多个iOS设备同时进行目标应用程序的启动,以及目标应用程序的页面截图的获取,进而在检测到各个iOS设备已获取到目标应用程序的截图时,调用移动端自动测试框架,从而通过移动端自动测试框架,向多个iOS设备发送截图存储指令和页面跳转指令,以便多个iOS设备同时页面跳转,并在页面跳转过程中,进行页面截图的存储和传输,从而基于接收到的页面截图,确定目标应用程序在多个iOS设备上的测试数据,实现了同时对多个iOS设备上的目标应用程序进行测试,从而提高了目标应用程序的测试效率。此外,通过为每个iOS设备设置相应的截图端口和页面跳转端口,以减少因多个iOS设备使用同一端口产生冲突的问题的出现,保证测试过程的顺利进行。而且,通过本申请提供的方案,无需对iOS设备进行录屏,也无需对录制得到的视频进行分帧,测试过程更加简单,从而能够提高测试过程的执行效率。

[0178] 在更多可能的实现方式中,服务器在多个iOS设备上同时确定目标应用程序的启动耗时数据和页面加载耗时数据,参见图8,图8是本申请实施例提供的一种应用程序测试方法的流程图,服务器(也即是Mac Sever)通过Usbmuxd连接有5个iOS设备,在对这5个iOS设备上的目标应用程序进行测试时,服务器调用Minicap Sever,通过Minicap Sever,来初始化目标应用程序,并基于AVFoundation框架和Core Media库,来进行页面截图的获取。在获取页面截图时,服务器调用Appium,以便通过Appium,控制各个iOS设备分别执行启动目标应用程序、在目标应用程序中进行页面跳转等自动化测试过程,使得各个iOS设备能够获取到目标应用程序的页面截图。在多个iOS设备均获取到页面截图后,将页面截图转化为iOS设备和服务器能够识别的二进制数据(包括Header数据和Frame数据),进而在接收到服务器的截图存储指令后,将Header数据和Frame数据通过相应的端口(Port),传输到服务器

上,以便服务器在接收到Header数据和Frame数据后,基于Header数据和Frame数据能够组装出完整的页面截图,从而保存页面截图,以便基于页面截图进行分析,以获取到目标应用程序在这个5个iOS设备上的测试数据。

[0179] 图8中仅以5个iOS设备为例来进行说明,在更多可能的实现方式中,iOS设备的数量更多或更少,本申请实施例对此不加以限定。此外,上述图8所示仅为一种示例性的流程说明,具体实现过程参见上述图2和图5所示的过程,此处不再赘述。

[0180] 上述所有可选技术方案,能够采用任意结合形成本申请的可选实施例,在此不再一一赘述。

[0181] 图9是本申请实施例提供的一种应用程序测试装置的结构示意图,参见图9,该装置包括:

[0182] 调用模块901,用于调用屏幕拷贝服务,该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能;

[0183] 发送模块902,用于通过该屏幕拷贝服务,向多个iOS设备发送启动指令,该启动指令用于指示启动该目标应用程序,以及获取该目标应用程序的页面截图;

[0184] 该调用模块901,还用于若检测到该多个iOS设备已获取到该目标应用程序的页面截图,则调用移动端自动测试框架,该移动端自动测试框架用于提供应用程序测试功能;

[0185] 该发送模块902,还用于通过该移动端自动测试框架,向该多个iOS设备发送截图存储指令,该截图存储指令用于指示该多个iOS设备对获取到的页面截图进行存储和传输;

[0186] 确定模块903,用于基于接收到的页面截图,确定该目标应用程序在该多个iOS设备上的测试数据。

[0187] 本申请实施例所提供的装置,通过调用屏幕拷贝服务,以便通过屏幕拷贝服务向多个iOS设备发送启动指令,以指示多个iOS设备同时进行目标应用程序的启动,以及目标应用程序的页面截图的获取,进而在检测到各个iOS设备已获取到目标应用程序的截图时,调用移动端自动测试框架,从而通过移动端自动测试框架,向多个iOS设备发送截图存储指令,以便多个iOS设备同时进行页面截图的存储和传输,从而基于接收到的页面截图,确定目标应用程序在多个iOS设备上的测试数据,实现了同时对多个iOS设备上的目标应用程序进行测试,从而提高了目标应用程序的测试效率。

[0188] 在一种可能的实现方式中,该测试数据为启动耗时数据;

[0189] 该确定模块903,用于对于任一iOS设备,基于该任一iOS设备对应的页面截图,确定应用启动起始帧和应用启动结束帧,该应用启动起始帧为该目标应用程序开始启动时所对应的页面截图,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图;基于该应用启动起始帧的时间戳和该应用启动结束帧的时间戳,确定该目标应用程序在该任一iOS设备上的启动耗时数据。

[0190] 在一种可能的实现方式中,该装置还包括:

[0191] 获取模块,用于获取该多个iOS设备的屏幕尺寸,其中,该多个iOS设备的屏幕尺寸通过屏幕拷贝服务获取得到;

[0192] 该应用启动结束帧的确定过程,包括:

[0193] 基于该任一iOS设备对应的任一页面截图和该任一iOS设备的屏幕尺寸,确定该任一页面截图与该任一iOS设备对应的其他页面截图的指纹数据的差异程度,该指纹数据用

于指示页面截图中的像素分布情况；

[0194] 当该差异程度小于第一预设阈值时，基于该任一页面截图的像素值均值和其他页面截图的像素值均值，确定差异图像，该差异图像为该任一页面截图中像素值均值小于第二预设阈值的部分；

[0195] 若该差异图像的像素值标准差小于第三预设阈值，则将该任一页面截图确定为该应用启动结束帧。

[0196] 在一种可能的实现方式中，该装置还包括：

[0197] 读取模块，用于读取页面配置文件，该页面配置文件用于指示该目标应用程序的各个页面的页面跳转顺序；

[0198] 该发送模块902，还用于基于该页面配置文件，向该多个iOS设备发送页面跳转指令，该页面跳转指令用于指示该多个iOS设备，按照该页面配置文件所指示的页面跳转顺序，显示该目标应用程序的各个页面。

[0199] 在一种可能的实现方式中，该测试数据包括页面加载耗时数据；

[0200] 该确定模块903，用于对于任一iOS设备，基于该任一iOS设备对应的页面截图和该任一iOS设备的屏幕尺寸，确定页面加载起始帧和页面加载结束帧，该页面加载起始帧为该目标应用程序在进行页面跳转前所对应的页面截图，该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图；基于该页面加载起始帧的时间戳和该页面加载结束帧的时间戳，确定该目标应用程序在该任一iOS设备上的页面加载耗时数据。

[0201] 在一种可能的实现方式中，该发送模块902，还用于在接收到该页面截图后，向该多个iOS设备发送存储停止指令，该存储停止指令用于指示该多个iOS设备停止该目标应用程序的页面截图的存储和传输。

[0202] 在一种可能的实现方式中，该装置还包括：

[0203] 开启模块，用于开启定长线程池，该定长线程池用于存储目标数量的iOS设备的截图获取任务，该截图获取任务用于获取对应iOS设备中目标应用程序的页面截图。

[0204] 图10是本申请实施例提供的一种应用程序测试装置的结构示意图，参见图10，该装置包括：

[0205] 启动模块1001，用于响应于启动指令，启动目标应用程序，该启动指令由服务器通过屏幕拷贝服务发送，该屏幕拷贝服务用于提供目标应用程序的页面截图的获取功能；

[0206] 获取模块1002，用于获取该目标应用程序的页面截图；

[0207] 存储模块1003，用于响应于截图存储指令，存储该页面截图，该截图存储指令由服务器通过移动端自动测试框架发送，该移动端自动测试框架用于提供应用程序测试功能；

[0208] 发送模块1004，用于发送该页面截图。

[0209] 本申请实施例提供的装置，通过读取页面配置文件，进而调用屏幕拷贝服务，以便通过屏幕拷贝服务向多个iOS设备发送启动指令，以指示多个iOS设备同时进行目标应用程序的启动，以及目标应用程序的页面截图的获取，进而在检测到各个iOS设备已获取到目标应用程序的截图时，调用移动端自动测试框架，从而通过移动端自动测试框架，向多个iOS设备发送截图存储指令和页面跳转指令，以便多个iOS设备同时页面跳转，并在页面跳转过程中，进行页面截图的存储和传输，从而基于接收到的页面截图，确定目标应用程序在多个iOS设备上的测试数据，实现了同时对多个iOS设备上的目标应用程序进行测试，从而提高

了目标应用程序的测试效率。

[0210] 在一种可能的实现方式中,该存储模块1003,用于启用iOS移动测试框架,通过该iOS移动测试框架,响应于该截图存储指令,存储该页面截图。

[0211] 在一种可能的实现方式中,该装置还包括:

[0212] 缓存模块,用于采用队列结构缓存获取到的页面截图;

[0213] 该存储模块1003,用于响应于该截图存储指令,对缓存中的页面截图进行封装;存储封装后的页面截图。

[0214] 在一种可能的实现方式中,该发送模块1004,用于通过套接字接口,发送该页面截图。

[0215] 在一种可能的实现方式中,该获取模块1002,还用于响应于该启动指令,获取iOS设备的屏幕尺寸;

[0216] 该发送模块1004,还用于发送该屏幕尺寸;

[0217] 其中,该屏幕尺寸用于确定该目标应用程序的应用启动结束帧和页面加载结束帧,该应用启动结束帧为该目标应用程序启动完成时所对应的页面截图,该页面加载结束帧为该目标应用程序完成页面跳转时所对应的页面截图。

[0218] 需要说明的是:上述实施例提供的应用程序测试装置在对目标应用程序进行测试时,仅以上述各功能模块的划分进行举例说明,实际应用中,可以根据需要而将上述功能分配由不同的功能模块完成,即将服务器或iOS设备的内部结构划分成不同的功能模块,以完成以上描述的全部或者部分功能。另外,上述实施例提供的应用程序测试装置与应用程序测试方法实施例属于同一构思,其具体实现过程详见方法实施例,这里不再赘述。

[0219] 图11是本申请实施例提供的一种iOS设备的结构示意图。该iOS设备1100可以是:搭载有iOS操作系统的智能手机、平板电脑、智能手表、台式电脑、笔记本电脑等。iOS设备1100还可能被称为用户设备、便携式终端、膝上型终端、台式终端等其他名称。

[0220] 通常,iOS设备1100包括有:一个或多个处理器1101和一个或多个存储器1102。

[0221] 处理器1101可以包括一个或多个处理核心,比如4核心处理器、8核心处理器等。处理器1101可以采用DSP (Digital Signal Processing,数字信号处理)、FPGA (Field-Programmable Gate Array,现场可编程门阵列)、PLA (Programmable Logic Array,可编程逻辑阵列)中的至少一种硬件形式来实现。处理器1101也可以包括主处理器和协处理器,主处理器是用于对在唤醒状态下的数据进行处理的处理器,也称CPU (Central Processing Unit,中央处理器);协处理器是用于对在待机状态下的数据进行处理的低功耗处理器。在一些实施例中,处理器1101可以集成有GPU (Graphics Processing Unit,图像处理器),GPU用于负责显示屏所需要显示的内容的渲染和绘制。一些实施例中,处理器1101还可以包括AI (Artificial Intelligence,人工智能) 处理器,该AI处理器用于处理有关机器学习的计算操作。

[0222] 存储器1102可以包括一个或多个计算机可读存储介质,该计算机可读存储介质可以是非暂态的。存储器1102还可包括高速随机存取存储器,以及非易失性存储器,比如一个或多个磁盘存储设备、闪存存储设备。在一些实施例中,存储器1102中的非暂态的计算机可读存储介质用于存储至少一个程序代码,该至少一个程序代码用于被处理器1101所执行以实现本申请中方法实施例提供的应用程序测试方法。

[0223] 在一些实施例中，iOS设备1100还可选包括有：外围设备接口1103和至少一个外围设备。处理器1101、存储器1102和外围设备接口1103之间可以通过总线或信号线相连。各个外围设备可以通过总线、信号线或电路板与外围设备接口1103相连。具体地，外围设备包括：射频电路1104、显示屏1105、摄像头组件1106、音频电路1107、定位组件1108和电源1109中的至少一种。

[0224] 外围设备接口1103可被用于将I/O (Input/Output, 输入/输出) 相关的至少一个外围设备连接到处理器1101和存储器1102。在一些实施例中，处理器1101、存储器1102和外围设备接口1103被集成在同一芯片或电路板上；在一些其他实施例中，处理器1101、存储器1102和外围设备接口1103中的任意一个或两个可以在单独的芯片或电路板上实现，本实施例对此不加以限定。

[0225] 射频电路1104用于接收和发射RF (Radio Frequency, 射频) 信号，也称电磁信号。射频电路1104通过电磁信号与通信网络以及其他通信设备进行通信。射频电路1104将电信号转换为电磁信号进行发送，或者，将接收到的电磁信号转换为电信号。可选地，射频电路1104包括：天线系统、RF收发器、一个或多个放大器、调谐器、振荡器、数字信号处理器、编解码芯片组、用户身份模块卡等等。射频电路1104可以通过至少一种无线通信协议来与其它iOS设备进行通信。该无线通信协议包括但不限于：城域网、各代移动通信网络(2G、3G、4G及5G)、无线局域网和/或WiFi (Wireless Fidelity, 无线保真) 网络。在一些实施例中，射频电路1104还可以包括NFC (Near Field Communication, 近距离无线通信) 有关的电路，本申请对此不加以限定。

[0226] 显示屏1105用于显示UI (User Interface, 用户界面)。该UI可以包括图形、文本、图标、视频及其它们的任意组合。当显示屏1105是触摸显示屏时，显示屏1105还具有采集在显示屏1105的表面或表面上方的触摸信号的能力。该触摸信号可以作为控制信号输入至处理器1101进行处理。此时，显示屏1105还可以用于提供虚拟按钮和/或虚拟键盘，也称软按钮和/或软键盘。在一些实施例中，显示屏1105可以为一个，设置在iOS设备1100的前面板；在另一些实施例中，显示屏1105可以为至少两个，分别设置在iOS设备1100的不同表面或呈折叠设计；在另一些实施例中，显示屏1105可以是柔性显示屏，设置在iOS设备1100的弯曲表面上或折叠面上。甚至，显示屏1105还可以设置成非矩形的不规则图形，也即异形屏。显示屏1105可以采用LCD (Liquid Crystal Display, 液晶显示屏)、OLED (Organic Light-Emitting Diode, 有机发光二极管) 等材质制备。

[0227] 摄像头组件1106用于采集图像或视频。可选地，摄像头组件1106包括前置摄像头和后置摄像头。通常，前置摄像头设置在iOS设备的前面板，后置摄像头设置在iOS设备的背面。在一些实施例中，后置摄像头为至少两个，分别为主摄像头、景深摄像头、广角摄像头、长焦摄像头中的任意一种，以实现主摄像头和景深摄像头融合实现背景虚化功能、主摄像头和广角摄像头融合实现全景拍摄以及VR (Virtual Reality, 虚拟现实) 拍摄功能或者其他融合拍摄功能。在一些实施例中，摄像头组件1106还可以包括闪光灯。闪光灯可以是单色温闪光灯，也可以是双色温闪光灯。双色温闪光灯是指暖光闪光灯和冷光闪光灯的组合，可以用于不同色温下的光线补偿。

[0228] 音频电路1107可以包括麦克风和扬声器。麦克风用于采集用户及环境的声波，并将声波转换为电信号输入至处理器1101进行处理，或者输入至射频电路1104以实现语音通

信。出于立体声采集或降噪的目的,麦克风可以为多个,分别设置在iOS设备1100的不同部位。麦克风还可以是阵列麦克风或全向采集型麦克风。扬声器则用于将来自处理器1101或射频电路1104的电信号转换为声波。扬声器可以是传统的薄膜扬声器,也可以是压电陶瓷扬声器。当扬声器是压电陶瓷扬声器时,不仅可以将电信号转换为人类可听见的声波,也可以将电信号转换为人类听不见的声波以进行测距等用途。在一些实施例中,音频电路1107还可以包括耳机插孔。

[0229] 定位组件1108用于定位iOS设备1100的当前地理位置,以实现导航或LBS(Location Based Service,基于位置的服务)。定位组件1108可以是基于美国的GPS(Global Positioning System,全球定位系统)、中国的北斗系统、俄罗斯的格雷纳斯系统或欧盟的伽利略系统的定位组件。

[0230] 电源1109用于为iOS设备1100中的各个组件进行供电。电源1109可以是交流电、直流电、一次性电池或可充电电池。当电源1109包括可充电电池时,该可充电电池可以支持有线充电或无线充电。该可充电电池还可以用于支持快充技术。

[0231] 在一些实施例中,iOS设备1100还包括有一个或多个传感器1110。该一个或多个传感器1110包括但不限于:加速度传感器1111、陀螺仪传感器1112、压力传感器1113、指纹传感器1114、光学传感器1115以及接近传感器1116。

[0232] 加速度传感器1111可以检测以iOS设备1100建立的坐标系的三个坐标轴上的加速度大小。比如,加速度传感器1111可以用于检测重力加速度在三个坐标轴上的分量。处理器1101可以根据加速度传感器1111采集的重力加速度信号,控制显示屏1105以横向视图或纵向视图进行用户界面的显示。加速度传感器1111还可以用于游戏或者用户的运动数据的采集。

[0233] 陀螺仪传感器1112可以检测iOS设备1100的机体方向及转动角度,陀螺仪传感器1112可以与加速度传感器1111协同采集用户对iOS设备1100的3D动作。处理器1101根据陀螺仪传感器1112采集的数据,可以实现如下功能:动作感应(比如根据用户的倾斜操作来改变UI)、拍摄时的图像稳定、游戏控制以及惯性导航。

[0234] 压力传感器1113可以设置在iOS设备1100的侧边框和/或显示屏1105的下层。当压力传感器1113设置在iOS设备1100的侧边框时,可以检测用户对iOS设备1100的握持信号,由处理器1101根据压力传感器1113采集的握持信号进行左右手识别或快捷操作。当压力传感器1113设置在显示屏1105的下层时,由处理器1101根据用户对显示屏1105的压力操作,实现对UI界面上的可操作性控件进行控制。可操作性控件包括按钮控件、滚动条控件、图标控件、菜单控件中的至少一种。

[0235] 指纹传感器1114用于采集用户的指纹,由处理器1101根据指纹传感器1114采集到的指纹识别用户的身份,或者,由指纹传感器1114根据采集到的指纹识别用户的身份。在识别出用户的身份为可信身份时,由处理器1101授权该用户执行相关的敏感操作,该敏感操作包括解锁屏幕、查看加密信息、下载软件、支付及更改设置等。指纹传感器1114可以被设置在iOS设备1100的正面、背面或侧面。当iOS设备1100上设置有物理按键或厂商Logo时,指纹传感器1114可以与物理按键或厂商Logo集成在一起。

[0236] 光学传感器1115用于采集环境光强度。在一个实施例中,处理器1101可以根据光学传感器1115采集的环境光强度,控制显示屏1105的显示亮度。具体地,当环境光强度较高

时,调高显示屏1105的显示亮度;当环境光强度较低时,调低显示屏1105的显示亮度。在另一个实施例中,处理器1101还可以根据光学传感器1115采集的环境光强度,动态调整摄像头组件1106的拍摄参数。

[0237] 接近传感器1116,也称距离传感器,通常设置在iOS设备1100的前面板。接近传感器1116用于采集用户与iOS设备1100的正面之间的距离。在一个实施例中,当接近传感器1116检测到用户与iOS设备1100的正面之间的距离逐渐变小时,由处理器1101控制显示屏1105从亮屏状态切换为息屏状态;当接近传感器1116检测到用户与iOS设备1100的正面之间的距离逐渐变大时,由处理器1101控制显示屏1105从息屏状态切换为亮屏状态。

[0238] 本领域技术人员可以理解,图11中示出的结构并不构成对iOS设备1100的限定,可以包括比图示更多或更少的组件,或者组合某些组件,或者采用不同的组件布置。

[0239] 图12是本申请实施例提供的一种服务器的结构示意图,该服务器1200可因配置或性能不同而产生比较大的差异,可以包括一个或多个处理器(Central Processing Units, CPU) 1001和一个或多个的存储器1202,其中,该一个或多个存储器1202中存储有至少一条程序代码,该至少一条程序代码由该一个或多个处理器1201加载并执行以实现上述各个方法实施例提供的方法。当然,该服务器1200还可以具有有线或无线网络接口、键盘以及输入输出接口等部件,以便进行输入输出,该服务器1200还可以包括其他用于实现设备功能的部件,在此不做赘述。

[0240] 在示例性实施例中,还提供了一种计算机可读存储介质,例如包括程序代码的存储器,上述程序代码可由处理器执行以完成上述实施例中的应用程序测试方法。例如,该计算机可读存储介质可以是只读存储器(Read-Only Memory, ROM)、随机存取存储器(Random Access Memory, RAM)、只读光盘(Compact Disc Read-Only Memory, CD-ROM)、磁带、软盘和光数据存储设备等。

[0241] 在示例性实施例中,还提供了一种计算机程序产品,该计算机程序产品包括计算机程序代码,该计算机程序代码由服务器的处理器加载并执行,以完成上述实施例中提供的应用程序测试方法的方法步骤。

[0242] 本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成,也可以通过程序来程序代码相关的硬件完成,该程序可以存储于一种计算机可读存储介质中,上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0243] 上述仅为本申请的可选实施例,并不用以限制本申请,凡在本申请的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本申请的保护范围之内。

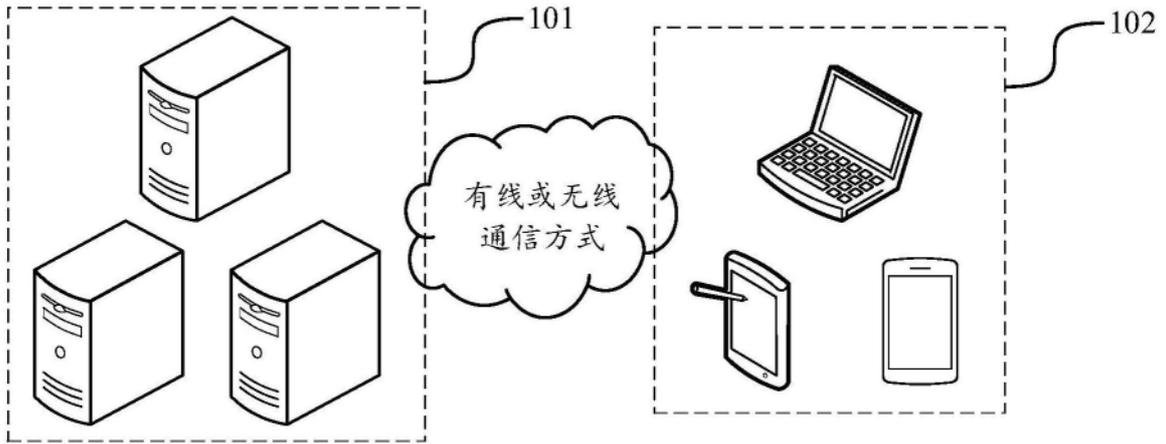


图1

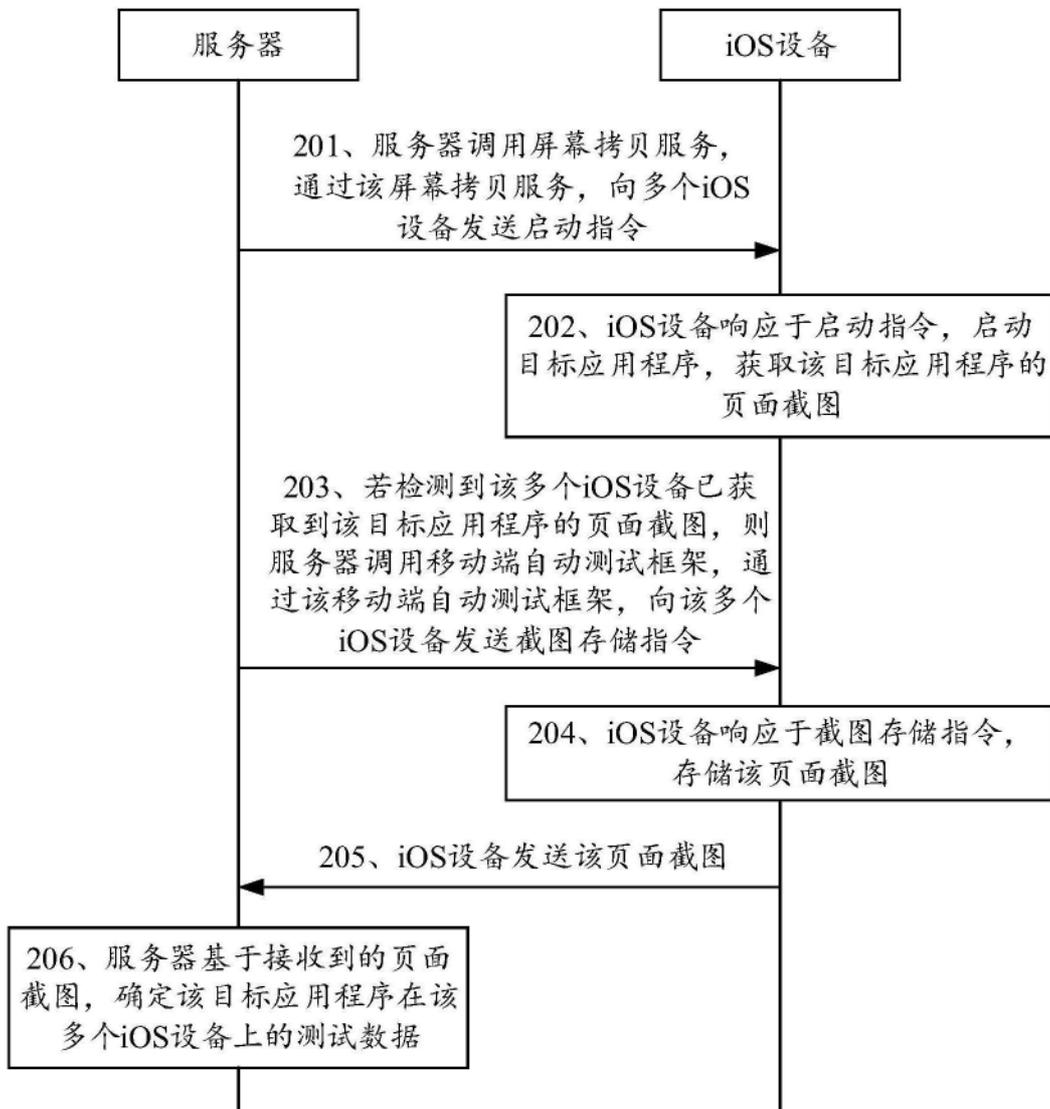


图2

启动耗时测试截图



图3

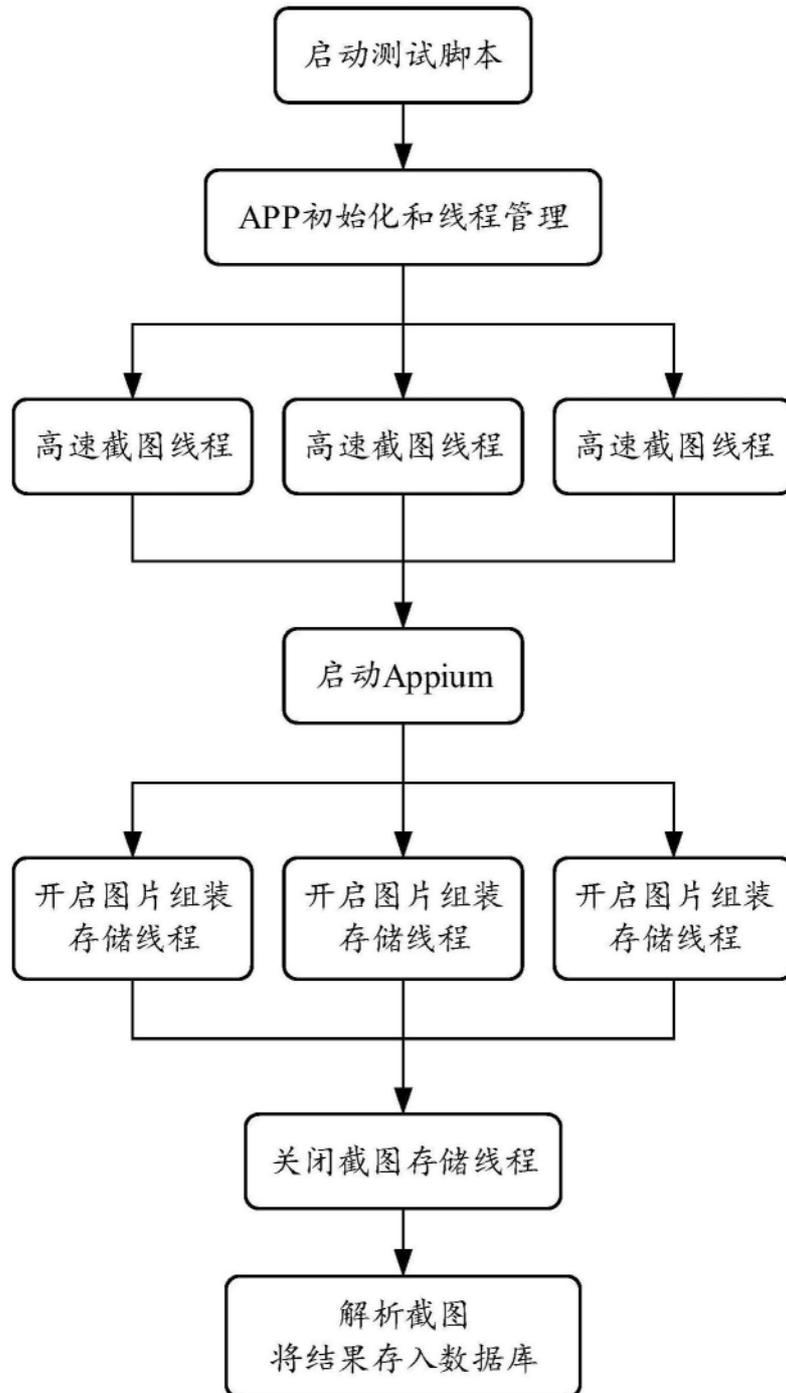


图4

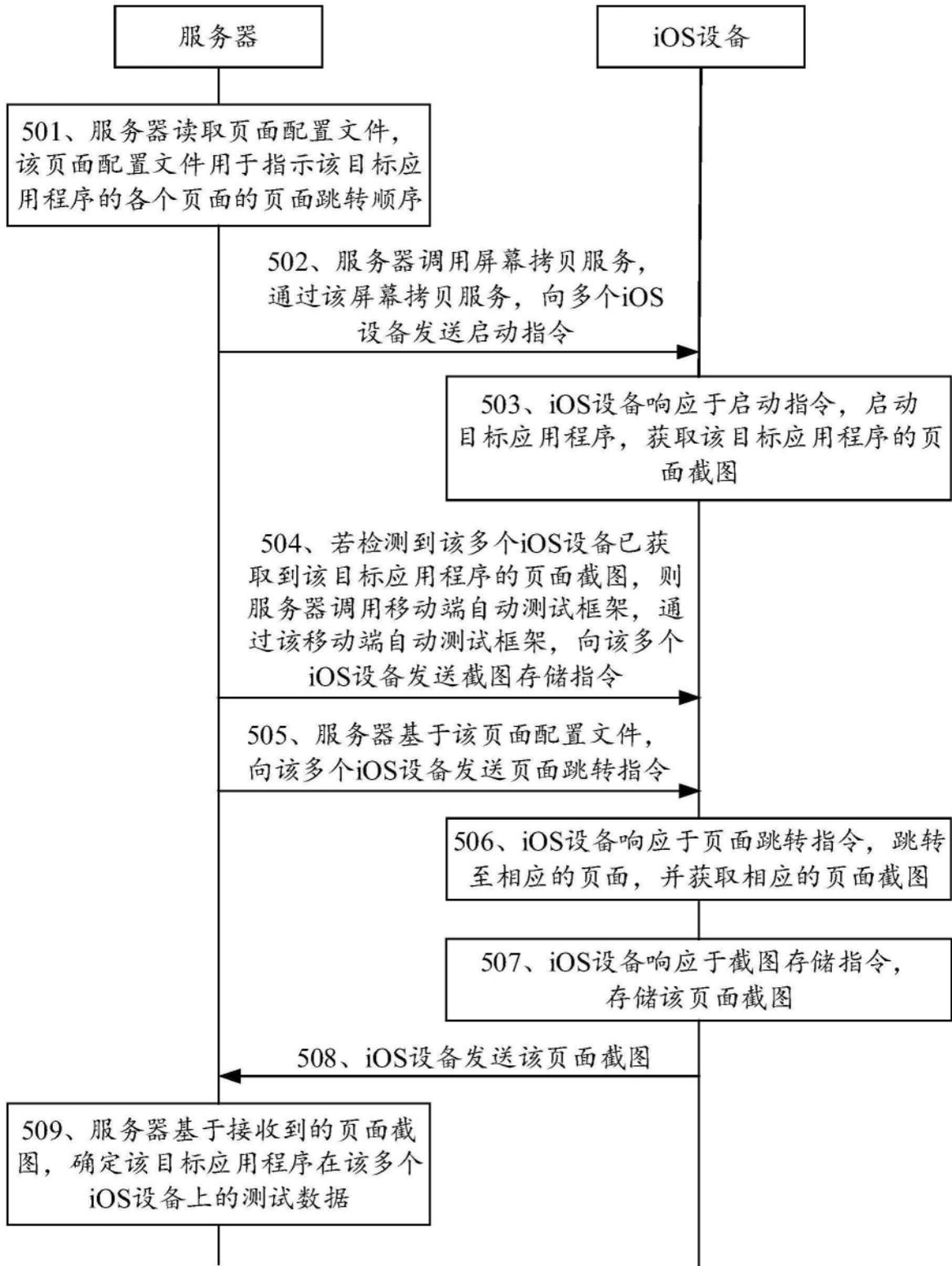


图5

页面加载耗时测试截图



图6

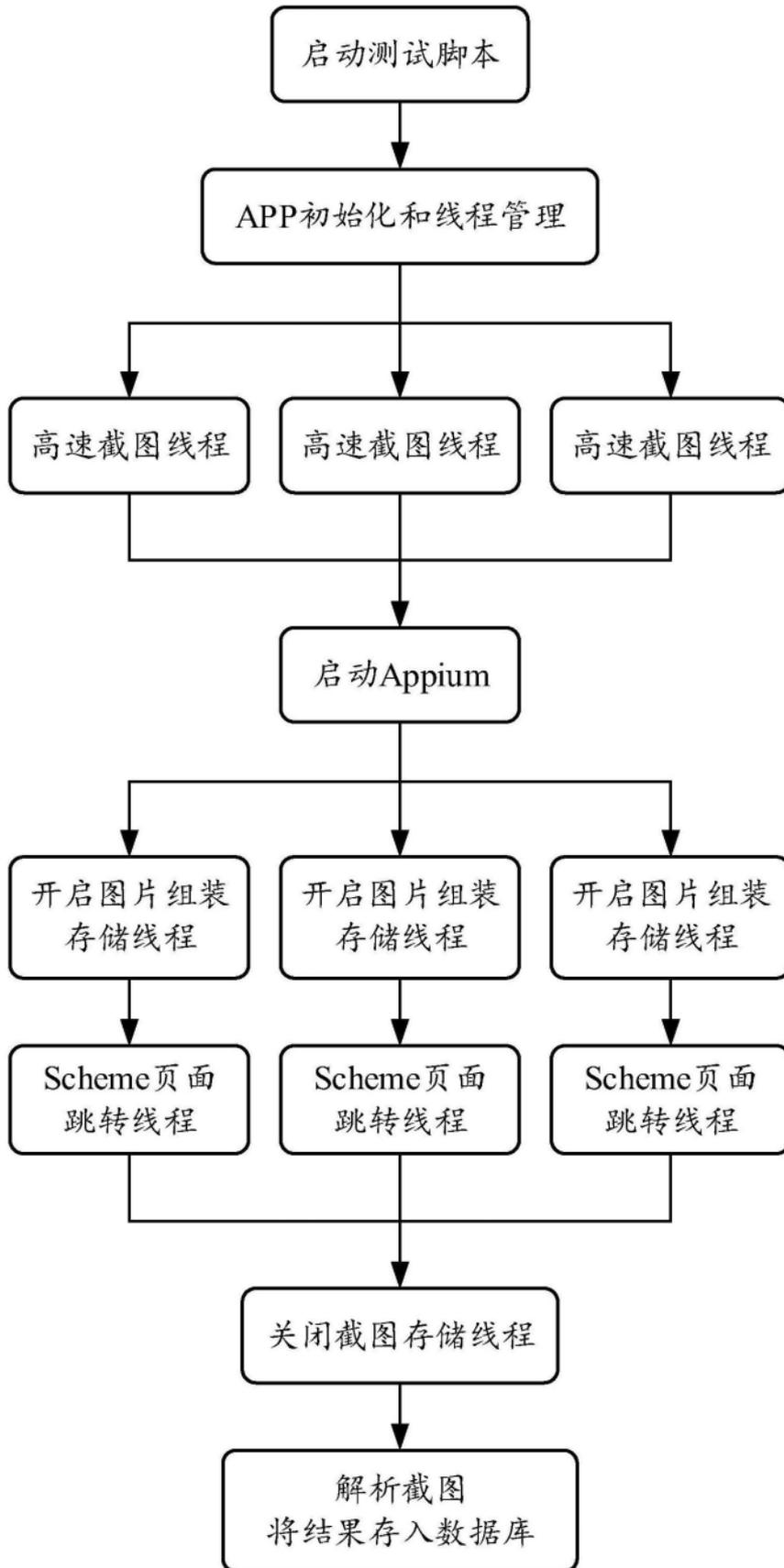


图7

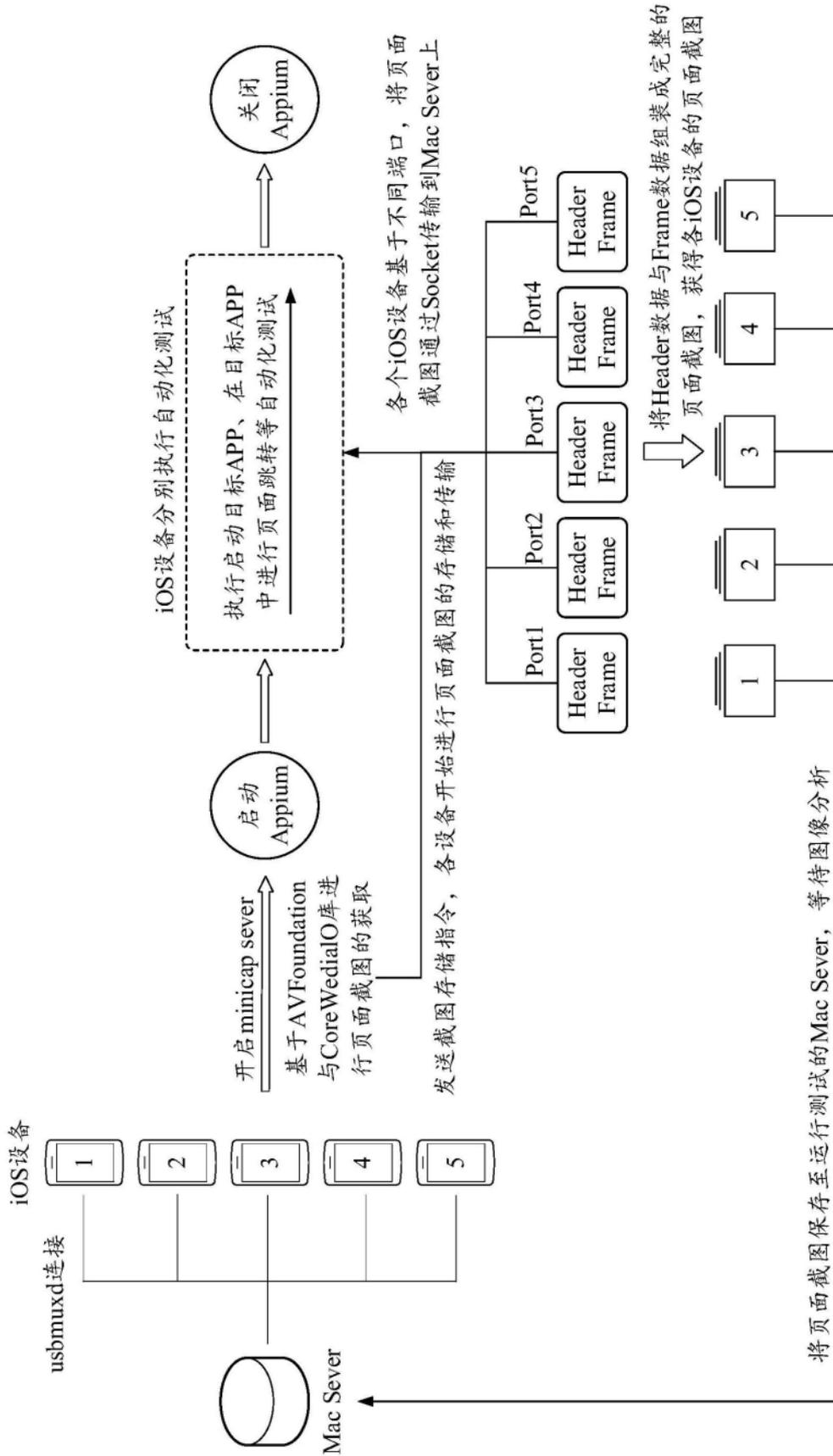


图8

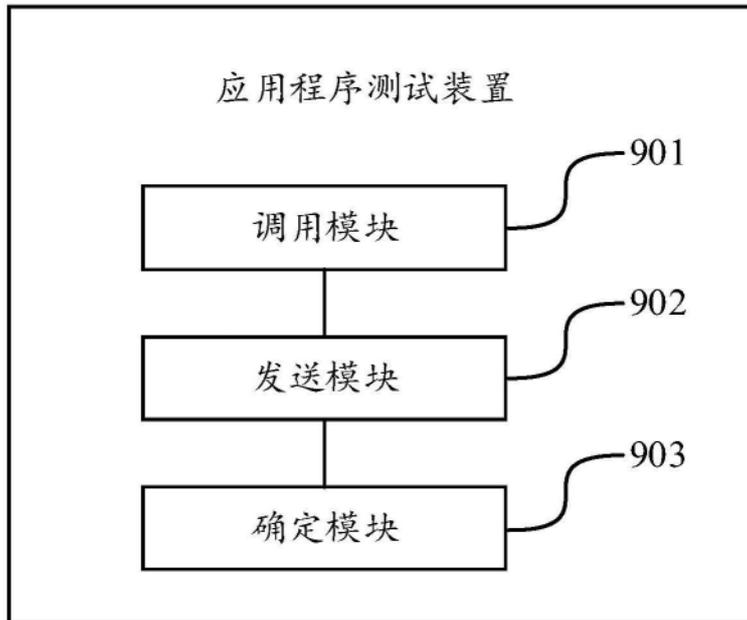


图9

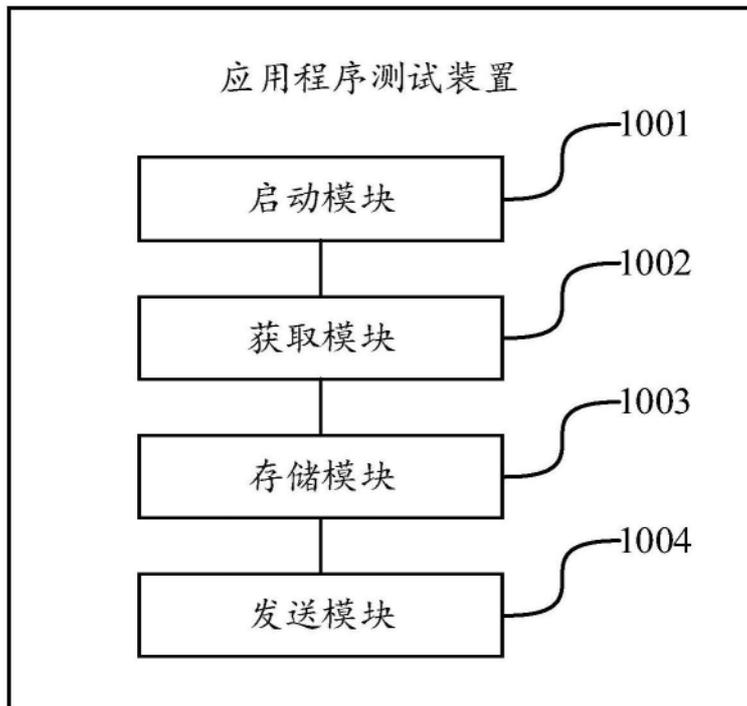


图10

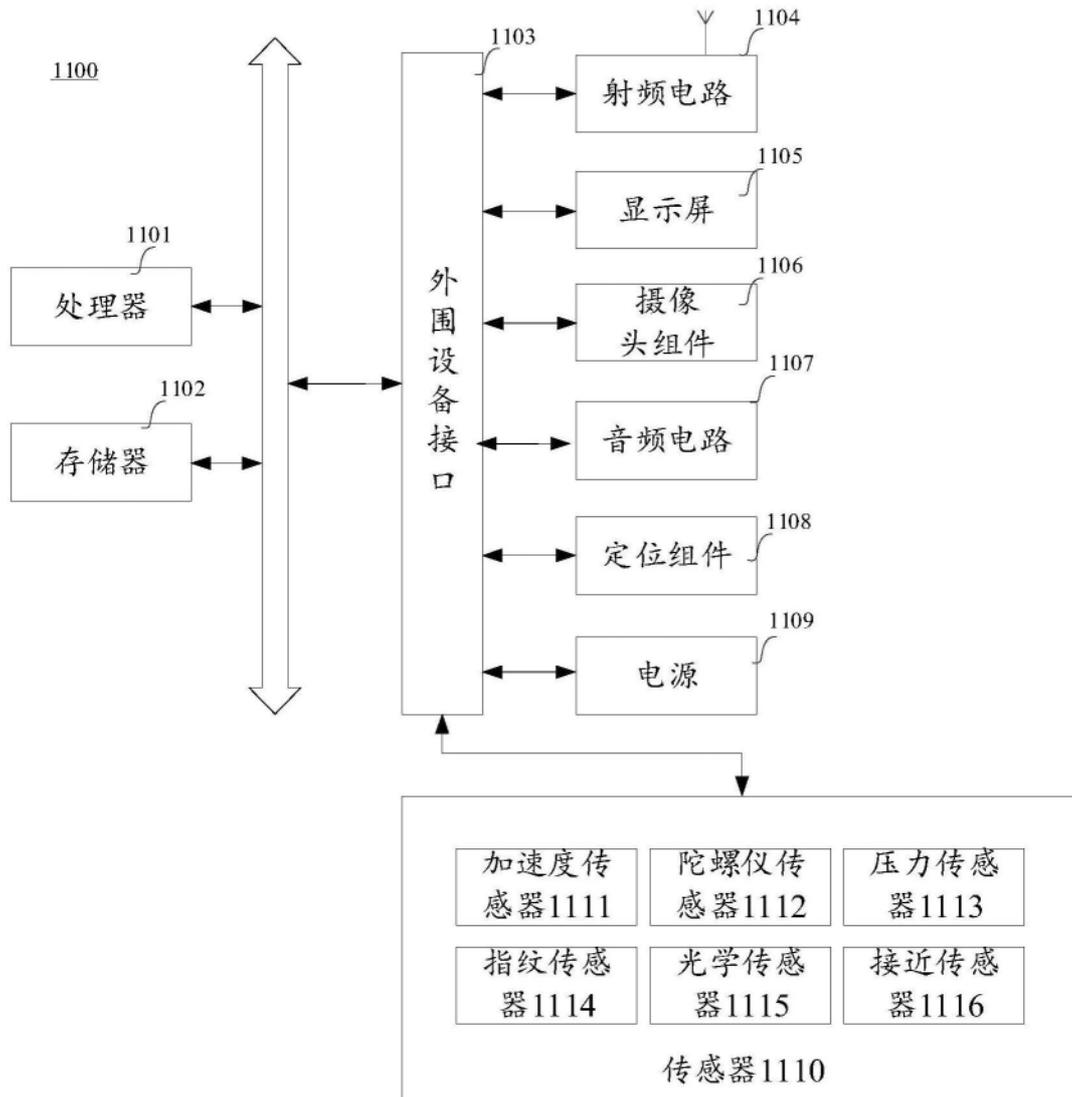


图11

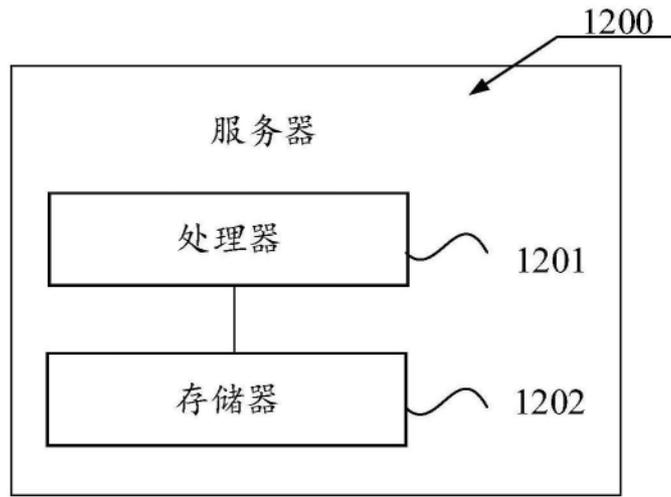


图12