(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110825456 B
(45)授权公告日 2024.01.23

(21)申请号 201911060726.3
(22)申请日 2019.11.01
(65)同一申请的已公布的文献号
申请公布号 CN 110825456 A
(43)申请公布日 2020.02.21
(73)专利权人 北京博睿宏远数据科技股份有限公司
地址 100027 北京市东城区东中街46号鸿基大厦4层
(72)发明人 邓小志 张彦春 刘科冲
(74)专利代理机构 北京品源专利代理有限公司 11332
专利代理师 孟金喆
(51)Int.Cl.
G06F 9/445(2018.01)
(56)对比文件
CN 107562610 A,2018.01.09
CN 108228463 A,2018.06.29
CN 109446095 A,2019.03.08
US 2017046254 A1,2017.02.16
US 2019304214 A1,2019.10.03
CN 110333903 A,2019.10.15
CN 109299000 A,2019.02.01
CN 102761467 A,2012.10.31
马雪琴 等.基于韦伯-费希纳定律的网页访问质量评价方法.《计算机与数字工程》.2016,第44卷(第01期),68-72.
吴清扬 等.基于完全渲染的网页首屏性能测量设计.《电子制作》.2018,(第16期),42-44.
马征.浅析资讯类App发展现状及央视新闻移动网客户端使用分析.《现代电视技术》.2019,(第06期),103-106.
王靖.面向Web加速的HTTP协议优化机制的研究与设计.《 CNKI优秀硕士学位论文全文库》.2015,(第04期),I139-296.
Murat Birinci, Serkan Kiranyaz.A perceptual scheme for fully automatic video shot boundary detection.《Signal Processing: Image Communication》.2014,第29卷(第3期),410-423.
审查员 李傲宇

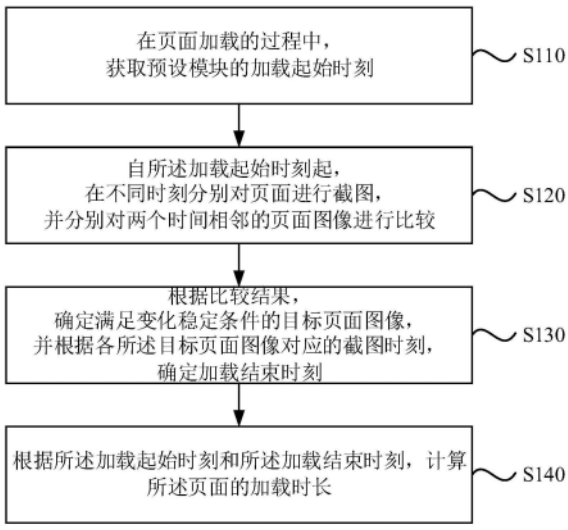
权利要求书2页 说明书9页 附图5页

(54)发明名称

加载时间计算方法、装置、计算机设备及存储介质

(57)摘要

本发明实施例公开了一种加载时间计算方法、装置、计算机设备及存储介质。所述方法包括:在页面加载的过程中,获取预设模块的加载起始时刻;自所述加载起始时刻起,在不同时刻分别对页面进行截图,并分别对两个时间相邻的页面图像进行比较;根据比较结果,确定满足变化稳定条件的目标页面图像,并根据各所述目标页面图像对应的截图时刻,确定加载结束时刻;根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。本发明实施例可以准确计算加载时间,从而准确评估应用程序性能。



CN 110825456 B

1.一种加载时间计算方法,其特征在于,包括:

在页面加载的过程中,获取预设模块的加载起始时刻;其中,所述加载的过程包括广告启动加载阶段、本地资源加载阶段、页面跳转阶段、网络资源加载阶段和页面绘制阶段;所述预设模块包括采用Hook技术注入的代码包;

自所述加载起始时刻起,在不同时刻分别对页面进行截图,包括:对页面中的稳定区域进行截图作为页面图像;其中,所述稳定区域中展示内容的变化频率满足低频条件;

并按照时间顺序分别对两个时间相邻的页面图像进行比较,直至首次出现相同结果的两个页面图像,包括:在所述页面加载完成之后,从所述至少两个截取的页面图像中,分别计算两个时间相邻的页面图像之间的相似度;如果所述两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值,确定所述两个目标页面图像的比较结果为相同结果;

将所述比较结果为相同结果的两个相邻时间的页面图像作为满足变化稳定条件的目标页面图像;获取各所述目标页面图像对应的截图时刻;将与所述加载起始时刻最近的截图时刻作为加载结束时刻;其中,所述变化稳定条件用于判断两个所述目标页面图像是否相同或相似;

根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。

2.根据权利要求1所述的方法,其特征在于,所述相似度阈值可以通过服务器或用户指令更新。

3.根据权利要求1所述的方法,其特征在于,所述页面包括应用程序首页。

4.一种加载时间计算装置,其特征在于,包括:

加载起始时刻获取模块,用于在页面加载的过程中,获取预设模块的加载起始时刻;其中,所述加载的过程包括广告启动加载阶段、本地资源加载阶段、页面跳转阶段、网络资源加载阶段和页面绘制阶段;所述预设模块包括采用Hook技术注入的代码包;

截图比较模块,用于自所述加载起始时刻起,在不同时刻分别对页面进行截图,并按照时间顺序分别对两个时间相邻的页面图像进行比较,直至出现相同结果的两个页面图像;

所述截图比较模块包括区域截图单元和图像相似度计算单元;

所述区域截图单元,用于对页面中的稳定区域进行截图作为页面图像;其中,所述稳定区域中展示内容的变化频率满足低频条件;

所述图像相似度计算单元,用于在所述页面加载完成之后,从所述至少两个截取的页面图像中,分别计算两个时间相邻的页面图像之间的相似度;如果所述两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值,确定所述两个目标页面图像的比较结果为相同结果;

加载结束时刻确定模块,用于将所述比较结果为相同结果的两个相邻时间的页面图像作为满足变化稳定条件的目标页面图像;获取各所述目标页面图像对应的截图时刻;将与所述加载起始时刻最近的截图时刻作为加载结束时刻;其中,所述变化稳定条件用于判断两个所述目标页面图像是否相同或相似;

加载时长计算模块,用于根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。

5.一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计

算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1-3中任一所述的加载时间计算方法。

6.一种计算机可读存储介质,其上存储有计算机程序,其特征在于,该程序被处理器执行时实现如权利要求1-3中任一所述的加载时间计算方法。

加载时间计算方法、装置、计算机设备及存储介质

技术领域

[0001] 本发明实施例涉及计算机网络领域，尤其涉及一种加载时间计算方法、装置、计算机设备及存储介质。

背景技术

[0002] 随着应用程序的发展和普及，用户对应用程序性能的要求越来越高。目前可以通过监控应用软件执行不同操作的时长，评估应用程序的性能。

[0003] 在应用程序开发的过程中，技术人员通常会对应用程序启动时间进行统计，以实现应用程序的测试，并基于测试结果对应用程序进行优化，提高用户体验。通常应用程序启动时间较短，但是从用户打开应用程序到程序能正常使用，需要很长的时间。现有的性能评估仅仅以启动时长作为判定依据。

[0004] 这种性能评估方式忽略了用户使用过程直观的加载用时，而且评估结果单一，只统计网络的加载时长或者页面加载时长。此外，在页面加载完成后，还有页面绘制的过程，还要考虑到异步加载的情况，即当页面加载完成之后，页面再去请求资源进一步完善页面内容等。

发明内容

[0005] 本发明实施例提供一种加载时间计算方法、装置、计算机设备及存储介质，可以准确计算加载时间，从而准确评估应用程序性能。

[0006] 第一方面，本发明实施例提供了一种加载时间计算方法，包括：

[0007] 在页面加载的过程中，获取预设模块的加载起始时刻；

[0008] 自所述加载起始时刻起，在不同时刻分别对页面进行截图，并分别对两个时间相邻的页面图像进行比较；

[0009] 根据比较结果，确定满足变化稳定条件的目标页面图像，并根据各所述目标页面图像对应的截图时刻，确定加载结束时刻；

[0010] 根据所述加载起始时刻和所述加载结束时刻，计算所述页面的加载时长。

[0011] 第二方面，本发明实施例还提供了一种加载时间计算装置，包括：

[0012] 加载起始时刻获取模块，用于在页面加载的过程中，获取预设模块的加载起始时刻；

[0013] 截图比较模块，用于自所述加载起始时刻起，在不同时刻分别对页面进行截图，并分别对两个时间相邻的页面图像进行比较；

[0014] 加载结束时刻确定模块，用于根据比较结果，确定满足变化稳定条件的目标页面图像，并根据各所述目标页面图像对应的截图时刻，确定加载结束时刻；

[0015] 加载时长计算模块，用于根据所述加载起始时刻和所述加载结束时刻，计算所述页面的加载时长。

[0016] 第三方面，本发明实施例还提供了一种计算机设备，包括存储器、处理器及存储在

存储器上并可在处理器上运行的计算机程序所述处理器执行所述程序时实现如本发明实施例中任一所述的加载时间计算方法。

[0017] 第四方面,本发明实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如本发明实施例中任一所述的加载时间计算方法。

[0018] 本发明实施例通过在加载过程中按照不同时刻分别对的页面进行截图,得到在不同加载时刻下的页面图像,比较每两个时间相邻的页面图像,将两个时间相邻且变化较小的页面图像作为目标页面图像,并根据目标页面图像的截图时刻确定加载结束时刻,将应用程序的显示页面不再变化的时刻作为加载结束时刻,解决了现有技术中仅统计应用程序的启动时间作为性能评判标准,以及加载时间点不准确导致性能评估可靠性差的问题,将显示屏中显示页面的稳定不变的时刻作为加载结束时刻,可以准确确定应用程序的时长,从而实现准确评估应用程序性能。

附图说明

[0019] 图1是本发明实施例一中的一种加载时间计算方法的流程图;

[0020] 图2a是本发明实施例二中的一种加载时间计算方法的流程图;

[0021] 图2b是本发明实施例二中的一种应用程序首页加载过程的流程图;

[0022] 图3是本发明实施例三中的一种加载时间计算装置的结构示意图;

[0023] 图4是本发明实施例四中的一种计算机设备的结构示意图。

具体实施方式

[0024] 下面结合附图和实施例对本发明作进一步的详细说明。可以理解的是,此处所描述的具体实施例仅仅用于解释本发明,而非对本发明的限定。另外还需要说明的是,为了便于描述,附图中仅示出了与本发明相关的部分而非全部结构。

[0025] 实施例一

[0026] 图1为本发明实施例一中的一种加载时间计算方法的流程图的示意图,本实施例可适用于在应用程序启动并开始运行后,计算应用程序的首页加载时间的情况,该方法可以由本发明实施例提供的加载时间计算装置来执行,该装置可采用软件和/或硬件的方式实现,并一般可集成计算机设备中,例如,终端设备等,更具体的,移动终端的客户端中。如图1所示,本实施例的方法具体包括:

[0027] S110,在页面加载的过程中,获取预设模块的加载起始时刻。

[0028] 页面可以是指应用程序的页面。预设模块用于确定页面加载的起始时刻,可以是指在页面加载过程中,第一个加载的功能程序,示例性的,可以是指so文件或jar文件等。

[0029] 可选的,所述预设模块包括采用Hook技术注入的代码包。采用Hook技术注入的代码包是指远程线程插入(注入)技术,指的是通过在另一个进程中创建远程线程的方法进入目标进程的内存地址空间。通过注入代码包,应用程序能够设置相应的子例程来监视应用程序的消息传递以及在这些消息到达目标窗口程序之前进行处理。通过Hook技术注入的代码包,降低监控预设模块加载的难度和实现成本,以及准确确定加载起始时刻。

[0030] 加载起始时刻,用于作为加载时长的起点,以计算加载时长。

[0031] 通常,应用程序在运行过程中需要加载至少一个页面。通过统计页面的加载时间,

用于评估应用程序的性能,例如,加载时长长,表明应用程序的性能差;加载时长短,表明应用程序的性能好。

[0032] 可选的,页面包括应用程序首页。其中,首页是指应用程序启动之后显示的第一个页面。

[0033] 实际上,应用程序首页加载完成之后,用户才可以通过交互操作,对应用程序进行控制。因此,首页加载时长具体是指从用户发出启动指令开始,到首页所有资源加载完成的时长,也即用户真正开始使用应用程序的时长,可以用于评估应用程序的运行速度,以评估应用程序的性能。

[0034] 具体的,应用程序的加载过程包括广告启动和加载阶段,本地资源加载阶段,页面跳转阶段,网络资源加载阶段和页面绘制阶段等。

[0035] 通过对应用程序的加载时长的计算,可以确定应用程序的性能,从而,技术人员可以通过减少广告时长,采用异步方式加载本地资源,缓存首页数据,和减少首页数据复杂度等方式中的至少一项,实现降低首页完全加载用时,以用户的角度出发优化应用程序。

[0036] S120,自所述加载起始时刻起,在不同时刻分别对页面进行截图,并分别对两个时间相邻的页面图像进行比较。

[0037] 截图用于获取当前页面的展示内容的图像。分别在不同时刻对页面进行截图,得到不同时刻下的页面图像。其中,页面截图可以是指对页面中的区域进行截图,也可以是对整个页面进行截图,具体根据需要进行设定,对此,本发明实施例不作具体限制。其中,页面中的区域可以预先配置,示例性的,可以是静态区域,其中,静态区域是指页面中内容几乎固定不变的区域,例如,原生页面中的页面元素,如控件或背景区域等。而从服务器获取并显示的资源会随着服务器提供的内容不同而实时改变,从而,该资源所属区域不是静态区域,该资源可以是指多媒体资源,如文本、图像、音频和视频等。页面中的区域还可以是除动态变化图像之外的区域,动态变化图像是指自动循环切换到图像,例如gif格式的图像。

[0038] 可选的,所述对页面进行截图,包括:对页面中稳定区域进行截图,其中,所述稳定区域中展示内容的变化频率满足低频条件。

[0039] 稳定区域可以是指页面中自动变化频率低的区域,其中,自动变化频率是指,在没有用户输入的操作指令的情况下,区域变化的频率。稳定区域用于判断页面是否加载完成。变化频率用于。低频条件用于判断稳定区域中的展示内容是否稳定不变。示例性的,低频条件可以包括图像的格式,相应的,如果展示内容对应的图像格式不为gif,则确定展示内容满足低频条件。低频条件还可以是频率阈值,相应的,如果展示内容的变化频率低于设定阈值,则确定展示内容满足低频条件。

[0040] 通过对稳定区域进行截图,作为页面图像,可以准确确定在页面中将网络资源加载完成的时刻,从而将网络资源的加载时间作为加载时长的部分,实现准确计算以应用程序正常使用为结束时刻的加载时长。

[0041] 此外,对静态区域进行截图,确定的加载结束时刻,从而确定的加载时长实际是,应用程序启动到页面将本地资源加载完成的时长。由此,用户可以根据需要,通过对页面不同区域的截图,以实现确定不同类型的加载时长。对此,本发明实施例不作具体限制。

[0042] 不同时刻对页面进行截图,可以是按照预设周期,分别对页面进行截图。

[0043] 两个时间相邻的页面图像,是指两个页面图像在时间上是相邻的,也即两个页面

图像的截图时刻之间不存在其他截取的页面图像。

[0044] 两个时间相邻的页面图像进行比较，实际是比较两个页面图像是否相似。两个时间相邻的页面图像进行比较用于判断两个时间相邻的页面图像是否相似，也即用于判断在对应的两个截图时刻中，该页面是否发生变化，从而判断页面是否加载完成。如果两个时间相邻的页面图像相同或相似，即在对应的两个截图时刻中，该页面没有发生变化，则确定该页面加载完成；如果两个时间相邻的页面图像不同，即在对应的两个截图时刻中，该页面发生变化，则确定该页面没有加载完成。

[0045] S130，根据比较结果，确定满足变化稳定条件的目标页面图像，并根据各所述目标页面图像对应的截图时刻，确定加载结束时刻。

[0046] 比较结果用于确定两个时间相邻的页面图像是否相似。变化稳定条件用于判断两个目标页面图像是否相同或相似。当两个页面图像相同或相似时，这两个页面图像均为目标页面图像。示例性的，如果确定比较结果为相同结果，则确定两个时间相邻的页面图像为满足变化稳定条件的目标页面图像。

[0047] 目标页面图像对应的截图时刻，可以是指截取得到目标页面图像的时刻，该时刻可以是指绝对系统时间，还可以是相对加载起始时刻。

[0048] 可以将时间轴中最早的截图时刻作为加载结束时刻。

[0049] 可选的，所述根据比较结果，确定满足变化稳定条件的目标页面图像，并根据各所述目标页面图像对应的截图时刻，确定加载结束时刻，包括：将比较结果为相同结果的两个相邻时间的页面图像作为满足变化稳定条件的目标页面图像；获取各所述目标页面图像对应的截图时刻；将与所述加载起始时刻最近的截图时刻作为加载结束时刻。

[0050] 其中，与加载起始时刻最近的截图时刻，用于确定页面稳定不变的最早时刻。与加载起始时刻最近的截图时刻，实际是指该截图时刻与加载起始时刻之间的时长最短，也即在时间轴中该截图时刻相较于其他截图时刻最早。

[0051] 通过将与加载起始时刻最近的截图时刻作为加载结束时刻，使加载结束时刻与实际加载完成时刻最接近，从而，准确确定加载结束时刻。

[0052] 需要说明的是，在对两个时间相邻的页面图像进行比较时，可以在比较结果中第一次出现相同结果时，停止后续的两个相邻时间的页面图像比较。也即，分别对两个时间相邻的页面图像进行比较，包括：按照时间顺序，分别对两个时间相邻的页面图像进行比较，直至首次出现相同结果的两个页面图像。从而可以节省比较过程的工作量，提高加载结束时刻的计算效率，从而提高加载时长的计算效率。

[0053] S140，根据所述加载起始时刻和所述加载结束时刻，计算所述页面的加载时长。

[0054] 加载起始时刻与加载结束时刻之间的时间差即为页面的加载时长。

[0055] 本发明实施例通过在加载过程中按照不同时刻分别对的页面进行截图，得到在不同加载时刻下的页面图像，比较每两个时间相邻的页面图像，将两个时间相邻且变化较小的页面图像作为目标页面图像，并根据目标页面图像的截图时刻确定加载结束时刻，将应用程序的显示页面不再变化的时刻作为加载结束时刻，解决了现有技术中仅统计应用程序的启动时间作为性能评判标准，以及加载时间点不准确导致性能评估可靠性差的问题，将显示屏中显示页面的稳定不变的时刻作为加载结束时刻，可以准确确定应用程序的时长，从而实现准确评估应用程序性能。

[0056] 实施例二

[0057] 图2a为本发明实施例二中的一种加载时间计算方法的流程图,本实施例以上述实施例为基础进行具体化,将所述分别对两个时间相邻的页面图像进行比较,具体化为:在所述页面加载完成之后,从所述至少两个截取的页面图像中,分别计算两个时间相邻的页面图像之间的相似度;如果两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值,确定所述两个目标页面图像的比较结果为相同结果。

[0058] 本实施例的方法具体包括:

[0059] S210,在页面加载的过程中,获取预设模块的加载起始时刻。

[0060] 本发明实施例中的页面,预设模块、加载起始时刻、截图、页面图像、变化稳定条件、目标页面图像、截图时刻、加载结束时刻和加载时长均可以参考上述实施例的描述。

[0061] 可选的,所述预设模块包括采用Hook技术注入的代码包。

[0062] 示例性的,代码包为jar包,相应的,加载时刻为应用程序ClassLoader加载jar包的时刻。

[0063] 可选的,所述页面包括应用程序首页。

[0064] S220,自所述加载起始时刻起,在不同时刻分别对页面进行截图。

[0065] 可选的,所述对页面进行截图,包括:对页面中稳定区域进行截图,其中,所述稳定区域中展示内容的变化频率满足低频条件。

[0066] S230,在所述页面加载完成之后,从所述至少两个截取的页面图像中,分别计算两个时间相邻的页面图像之间的相似度。

[0067] 相似度用于评价两个页面图像的相似程度。

[0068] S240,如果两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值,确定所述两个目标页面图像的比较结果为相同结果。

[0069] 其中,相似度阈值用于判断两个页面图像是否相同。通常,如果相似度超过相似度阈值,确定两个页面图像相同。

[0070] 可选的,所述相似度阈值可以通过服务器或用户指令更新。

[0071] 相似度阈值可以实时配置,例如通过服务器下发数据进行更新,或者直接通过用户指令接收到的输入数据进行更新。通过更新相似度阈值,可以提高相似度阈值的配置灵活性,兼容多个应用场景。

[0072] S250,根据比较结果,确定满足变化稳定条件的目标页面图像,并根据各所述目标页面图像对应的截图时刻,确定加载结束时刻。

[0073] 可选的,所述根据比较结果,确定满足变化稳定条件的目标页面图像,并根据各所述目标页面图像对应的截图时刻,确定加载结束时刻,包括:将比较结果为相同结果的两个相邻时间的页面图像作为满足变化稳定条件的目标页面图像;获取各所述目标页面图像对应的截图时刻;将与所述加载起始时刻最近的截图时刻作为加载结束时刻。

[0074] S260,根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。

[0075] 本发明实施例的页面的加载时长实际是应用程序开始加载预设模块到页面信息绘制完成之间的时长。具体的,如图2b所示,应用程序的首页加载过程具体包括:

[0076] S271,接收到应用程序启动指令。

[0077] 启动指令可以是指用户输入的触发应用程序启动的指令。例如,启动指令可以是

用户通过移动终端的触摸屏对应用程序的图标的点击操作。

[0078] S272,启动Zygote进程,加载so文件,以及Hook注入的jar包。

[0079] 具体的,Zygote进程又称受精卵进程,它由app_process启动,Zygote进程最大意义是作为一个Socket的Server端,接收着四面八方的进程创建请求,Android中所有的应用进程的创建都是一个应用进程通过Binder请求System Server进程,System Server进程发送socket消息给Zygote进程,统一由Zygote进程创建出来的。

[0080] S273,创建Application和Splash Activity。

[0081] 应用首次启动时,首先Application进程,进入Splash Activity(启动页),经过短时间(如1-2秒)跳转至应用程序的主界面(如Main Activity)。通常,Splash Activity中会展示广告或图片。

[0082] S274,加载资源,初始化库。

[0083] 获取Splash Activity的本地资源以及Main Activity的本地资源,并进行加载和初始化。

[0084] S275,请求和播放启动广告。

[0085] 获取在Splash Activity中待展示的启动广告,并播放。

[0086] S276,启动Main Activity。

[0087] Main Activity即为应用程序的主界面。

[0088] S277,通过Main Activity绘制应用程序首页信息。

[0089] 在应用程序的主界面中绘制首页信息。

[0090] S278,应用程序首页信息绘制完成。

[0091] 首页信息绘制完成之后,用户可以通过交互操作控制应用程序执行相应功能。

[0092] 其中,S272可以作为加载起始时刻,S278作为加载结束时刻。为了避免产生对应用程序进行首页加载的干扰,本发明实施例提供的加载时间计算方法,通常在上述应用程序的首页加载过程中仅进行截图操作,在S278之后,即在首页信息绘制完成之后进行执行,页面图像比较操作以及加载时长计算操作等。

[0093] 本发明实施例通过获取页面加载过中的在不同加载时刻下的页面图像,比较每两个时间相邻的页面图像,将两个时间相邻且相似的页面图像作为目标页面图像,并根据目标页面图像的截图时刻确定加载结束时刻,将应用程序的显示页面不再变化的时刻作为加载结束时刻,可以准确确定应用程序启动到可即时使用的时长,包括本地资源加载阶段、网络资源加载阶段和页面绘制阶段,从而实现评估应用程序的本地资源加载性能、网络资源加载性能和页面绘制性能等,从而实现全面完整准确评估应用程序性能,提高应用程序性能评估的可靠性和准确性。

[0094] 实施例三

[0095] 图3为本发明实施例三中的一种加载时间计算装置的示意图。实施例三是实现本发明上述实施例提供的加载时间计算方法的相应装置,该装置可采用软件和/或硬件的方式实现,并一般可集成计算机设备中,例如,终端设备等,具体是,手机、平板电脑或车载设备等。

[0096] 相应的,本实施例的装置可以包括:

[0097] 加载起始时刻获取模块310,用于在页面加载的过程中,获取预设模块的加载起始

时刻;

[0098] 截图比较模块320,用于自所述加载起始时刻起,在不同时刻分别对页面进行截图,并分别对两个时间相邻的页面图像进行比较;

[0099] 加载结束时刻确定模块330,用于根据比较结果,确定满足变化稳定条件的目标页面图像,并根据各所述目标页面图像对应的截图时刻,确定加载结束时刻;

[0100] 加载时长计算模块340,用于根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。

[0101] 本发明实施例通过在加载过程中按照不同时刻分别对的页面进行截图,得到在不同加载时刻下的页面图像,比较每两个时间相邻的页面图像,将两个时间相邻且变化较小的页面图像作为目标页面图像,并根据目标页面图像的截图时刻确定加载结束时刻,将应用程序的显示页面不再变化的时刻作为加载结束时刻,解决了现有技术中仅统计应用程序的启动时间作为性能评判标准,以及加载时间点不准确导致性能评估可靠性差的问题,将显示屏中显示页面的稳定不变的时刻作为加载结束时刻,可以准确确定应用程序的时长,从而实现准确评估应用程序性能。

[0102] 进一步的,所述加载结束时刻确定模块330,包括:目标页面图像确定单元,用于将比较结果为相同结果的两个相邻时间的页面图像作为满足变化稳定条件的目标页面图像;获取各所述目标页面图像对应的截图时刻;将与所述加载起始时刻最近的截图时刻作为加载结束时刻。

[0103] 进一步的,所述截图比较模块320,包括:区域截图单元,用于对页面中稳定区域进行截图,其中,所述稳定区域中展示内容的变化频率满足低频条件。

[0104] 进一步的,所述截图比较模块320,包括:图像相似度计算单元,用于在所述页面加载完成之后,从所述至少两个截取的页面图像中,分别计算两个时间相邻的页面图像之间的相似度;如果两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值,确定所述两个目标页面图像的比较结果为相同结果。

[0105] 进一步的,所述相似度阈值可以通过服务器或用户指令更新。

[0106] 进一步的,所述预设模块包括采用Hook技术注入的代码包。

[0107] 进一步的,所述页面包括应用程序首页。

[0108] 上述加载时间计算装置可执行本发明实施例所提供的加载时间计算方法,具备执行的加载时间计算方法相应的功能模块和有益效果。

[0109] 实施例四

[0110] 图4为本发明实施例四提供的一种计算机设备的结构示意图。图4示出了适于用来实现本发明实施方式的示例性计算机设备12的框图。图4显示的计算机设备12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。

[0111] 如图4所示,计算机设备12以通用计算设备的形式表现。计算机设备12的组件可以包括但不限于:一个或者多个处理器或者处理单元16,系统存储器28,连接不同系统组件(包括系统存储器28和处理单元16)的总线18。计算机设备12可以是挂接在总线上的设备。

[0112] 总线18表示几类总线结构中的一种或多种,包括存储器总线或者存储器控制器,外围总线,图形加速端口,处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说,这些体系结构包括但不限于工业标准体系结构(Industry Standard

Architecture,ISA)总线,微通道体系结构(Micro Channel Architecture,MCA)总线,增强型ISA总线、视频电子标准协会(Video Electronics Standards Association,VESA)局域总线以及外围组件互连(Peripheral Component Interconnect,PCI)总线。

[0113] 计算机设备12典型地包括多种计算机系统可读介质。这些介质可以是任何能够被计算机设备12访问的可用介质,包括易失性和非易失性介质,可移动的和不可移动的介质。

[0114] 系统存储器28可以包括易失性存储器形式的计算机系统可读介质,例如随机存取存储器(RAM)30和/或高速缓存存储器32。计算机设备12可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例,存储系统34可以用于读写不可移动的、非易失性磁介质(图4未显示,通常称为“硬盘驱动器”)。尽管图4中未示出,可以提供用于对可移动非易失性磁盘(例如“软盘”)读写的磁盘驱动器,以及对可移动非易失性光盘(例如紧凑磁盘只读存储器(Compact Disc Read-Only Memory,CD-ROM),数字视盘(Digital Video Disc-Read Only Memory,DVD-ROM)或者其它光介质)读写的光盘驱动器。在这些情况下,每个驱动器可以通过一个或者多个数据介质接口与总线18相连。系统存储器28可以包括至少一个程序产品,该程序产品具有一组(例如至少一个)程序模块,这些程序模块被配置以执行本发明各实施例的功能。

[0115] 具有一组(至少一个)程序模块42的程序/实用工具40,可以存储在例如系统存储器28中,这样的程序模块42包括——但不限于——操作系统、一个或者多个应用程序、其它程序模块以及程序数据,这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块42通常执行本发明所描述的实施例中的功能和/或方法。

[0116] 计算机设备12也可以与一个或多个外部设备14(例如键盘、指向设备、显示器24等)通信,还可与一个或者多个使得用户能与该计算机设备12交互的设备通信,和/或与使得该计算机设备12能与一个或多个其它计算设备进行通信的任何设备(例如网卡,调制解调器等等)通信。这种通信可以通过输入/输出(Input/Output,I/O)接口22进行。并且,计算机设备12还可以通过网络适配器20与一个或者多个网络(例如局域网(Local Area Network,LAN),广域网(Wide Area Network,WAN)通信。如图所示,网络适配器20通过总线18与计算机设备12的其它模块通信。应当明白,尽管图4中未示出,可以结合计算机设备12使用其它硬件和/或软件模块,包括但不限于:微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、(Redundant Arrays of Inexpensive Disks,RAID)系统、磁带驱动器以及数据备份存储系统等。

[0117] 处理单元16通过运行存储在系统存储器28中的程序,从而执行各种功能应用以及数据处理,例如实现本发明任意实施例所提供的一种加载时间计算方法。

[0118] 实施例五

[0119] 本发明实施例五提供了一种计算机可读存储介质,其上存储有计算机程序,该程序被处理器执行时实现如本申请所有发明实施例提供的加载时间计算方法:

[0120] 也即,该程序被处理器执行时实现:在页面加载的过程中,获取预设模块的加载起始时刻;自所述加载起始时刻起,在不同时刻分别对页面进行截图,并分别对两个时间相邻的页面图像进行比较;根据比较结果,确定满足变化稳定条件的目标页面图像,并根据各所述目标页面图像对应的截图时刻,确定加载结束时刻;根据所述加载起始时刻和所述加载结束时刻,计算所述页面的加载时长。

[0121] 本发明实施例的计算机存储介质,可以采用一个或多个计算机可读的介质的任意组合。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。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(非穷举的列表)包括:具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、RAM、只读存储器(Read Only Memory,ROM)、可擦式可编程只读存储器(Erasable Programmable Read Only Memory,EPROM)、闪存、光纤、便携式CD-ROM、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中,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。

[0122] 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。

[0123] 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、无线电频率(RadioFrequency,RF)等等,或者上述的任意合适的组合。

[0124] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码,所述程序设计语言包括面向对象的程序设计语言—诸如Java、Smalltalk、C++,还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中,远程计算机可以通过任意种类的网络——包括LAN或WAN——连接到用户计算机,或者,可以连接到外部计算机(例如利用因特网服务提供商来通过因特网连接)。

[0125] 注意,上述仅为本发明的较佳实施例及所运用技术原理。本领域技术人员会理解,本发明不限于这里所述的特定实施例,对本领域技术人员来说能够进行各种明显的变化、重新调整和替代而不会脱离本发明的保护范围。因此,虽然通过以上实施例对本发明进行了较为详细的说明,但是本发明不仅仅限于以上实施例,在不脱离本发明构思的情况下,还可以包括更多其他等效实施例,而本发明的范围由所附的权利要求范围决定。

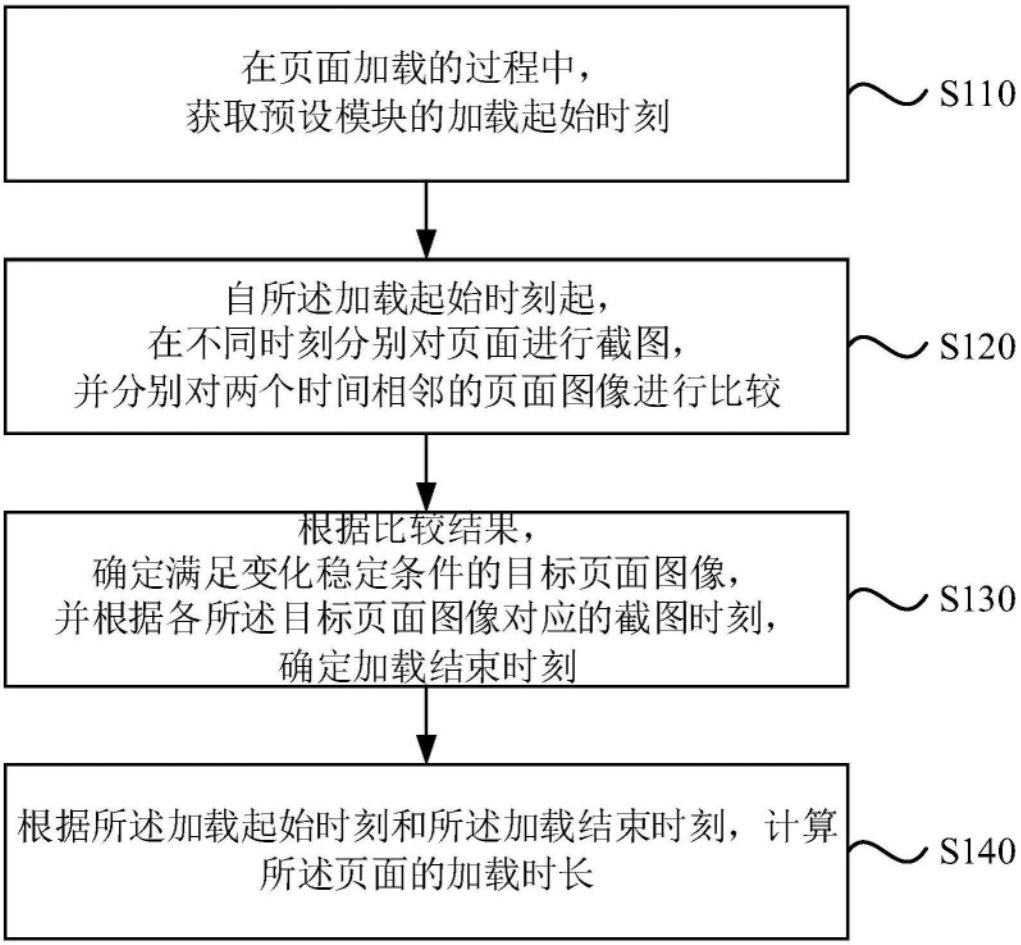

在页面加载的过程中，
获取预设模块的加载起始时刻
S110
自所述加载起始时刻起，
在不同时刻分别对页面进行截图，
并分别对两个时间相邻的页面图像进行比较
S120
根据比较结果，
确定满足变化稳定条件的目标页面图像，
并根据各所述目标页面图像对应的截图时刻，
确定加载结束时刻
S130
根据所述加载起始时刻和所述加载结束时刻，计算
所述页面的加载时长
S140

图1

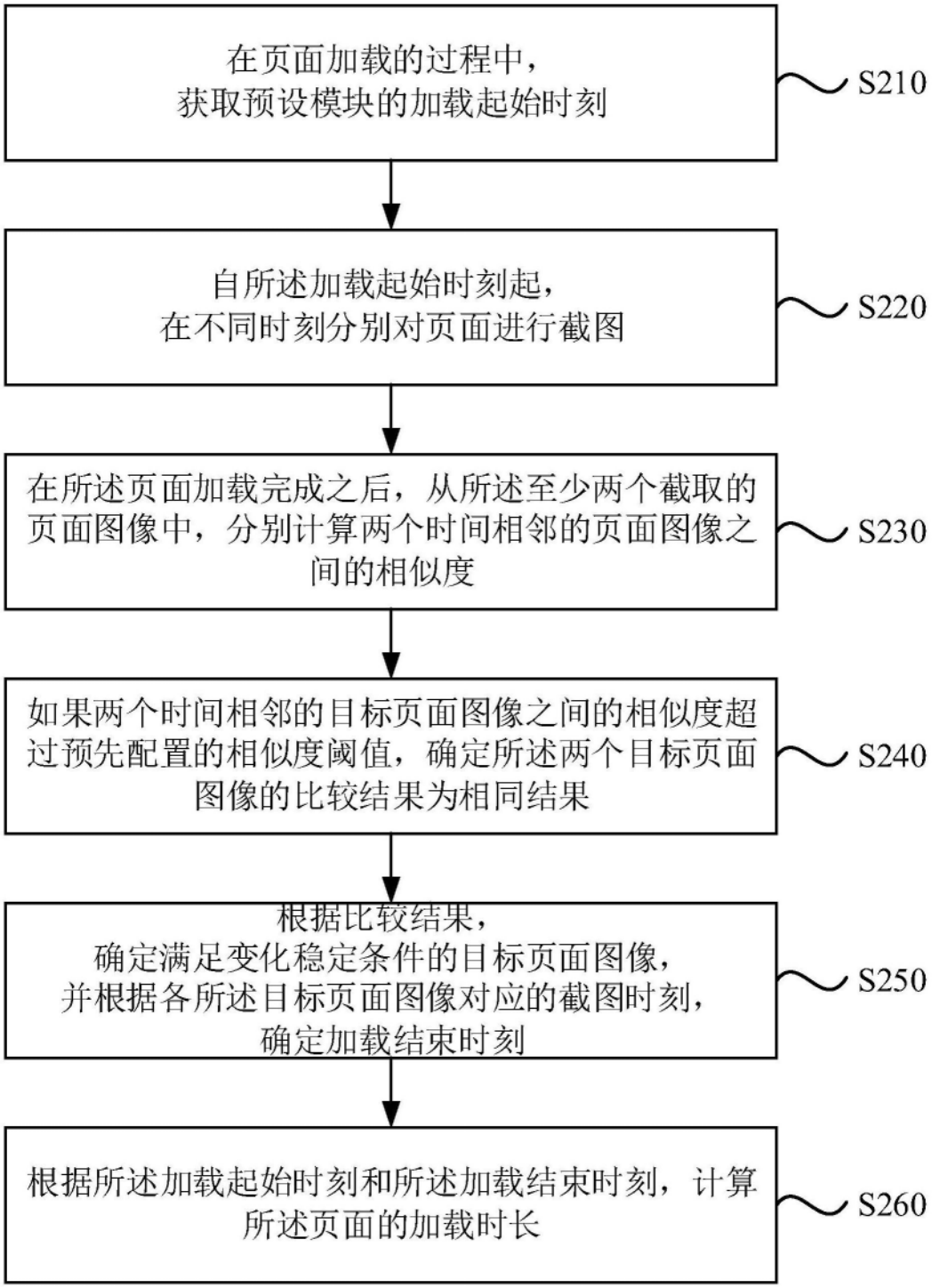

在页面加载的过程中，
获取预设模块的加载起始时刻
S210
自所述加载起始时刻起，
在不同时刻分别对页面进行截图
S220
在所述页面加载完成之后，从所述至少两个截取的页面图像中，分别计算两个时间相邻的页面图像之间的相似度
S230
如果两个时间相邻的目标页面图像之间的相似度超过预先配置的相似度阈值，确定所述两个目标页面图像的比较结果为相同结果
S240
根据比较结果，
确定满足变化稳定条件的目标页面图像，
并根据各所述目标页面图像对应的截图时刻，
确定加载结束时刻
S250
根据所述加载起始时刻和所述加载结束时刻，计算所述页面的加载时长
S260

图2a

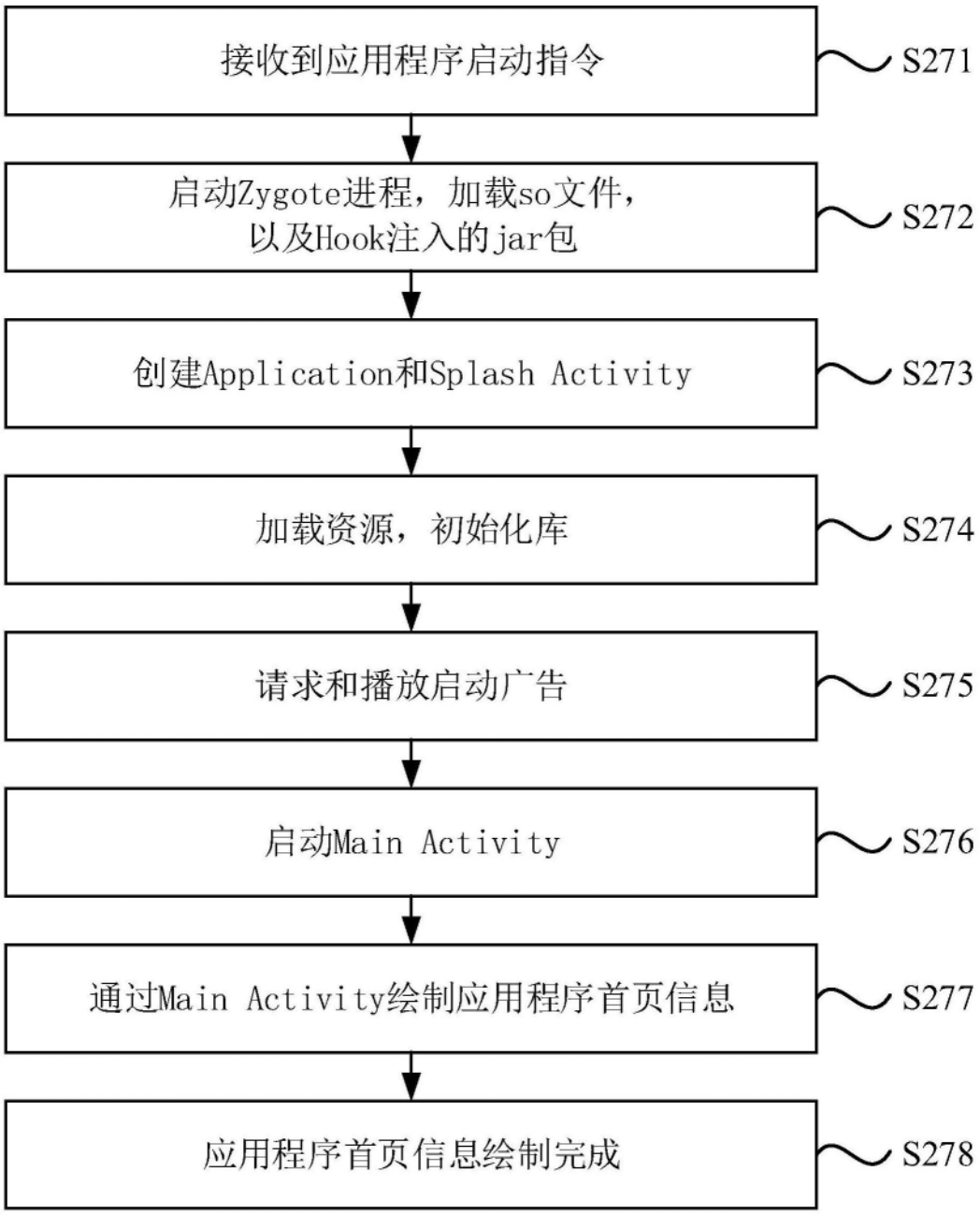

接收到应用程序启动指令
S271
启动Zygote进程，加载so文件，
以及Hook注入的jar包
S272
创建Application和Splash Activity
S273
加载资源，初始化库
S274
请求和播放启动广告
S275
启动Main Activity
S276
通过Main Activity绘制应用程序首页信息
S277
应用程序首页信息绘制完成
S278

图2b

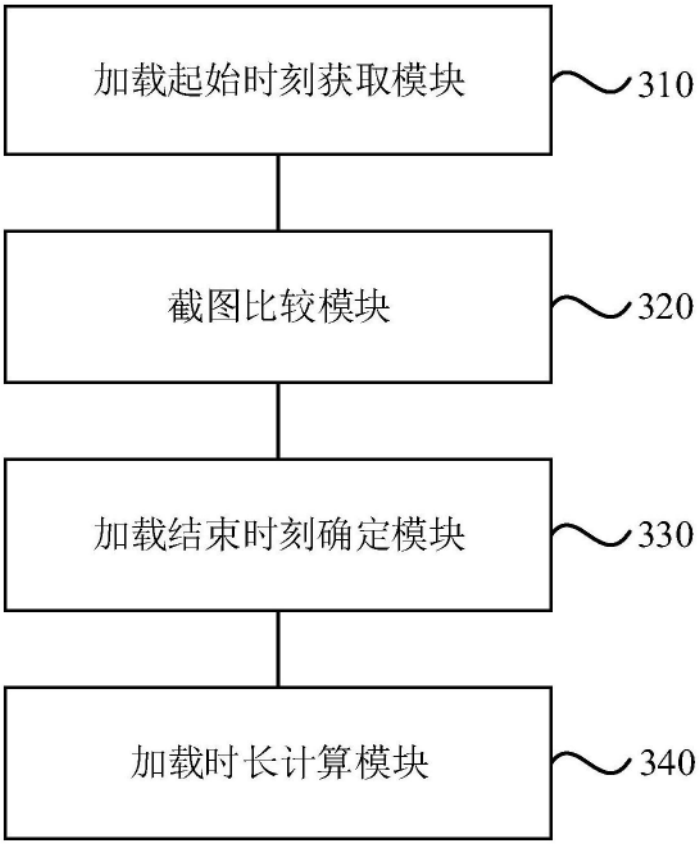


图3

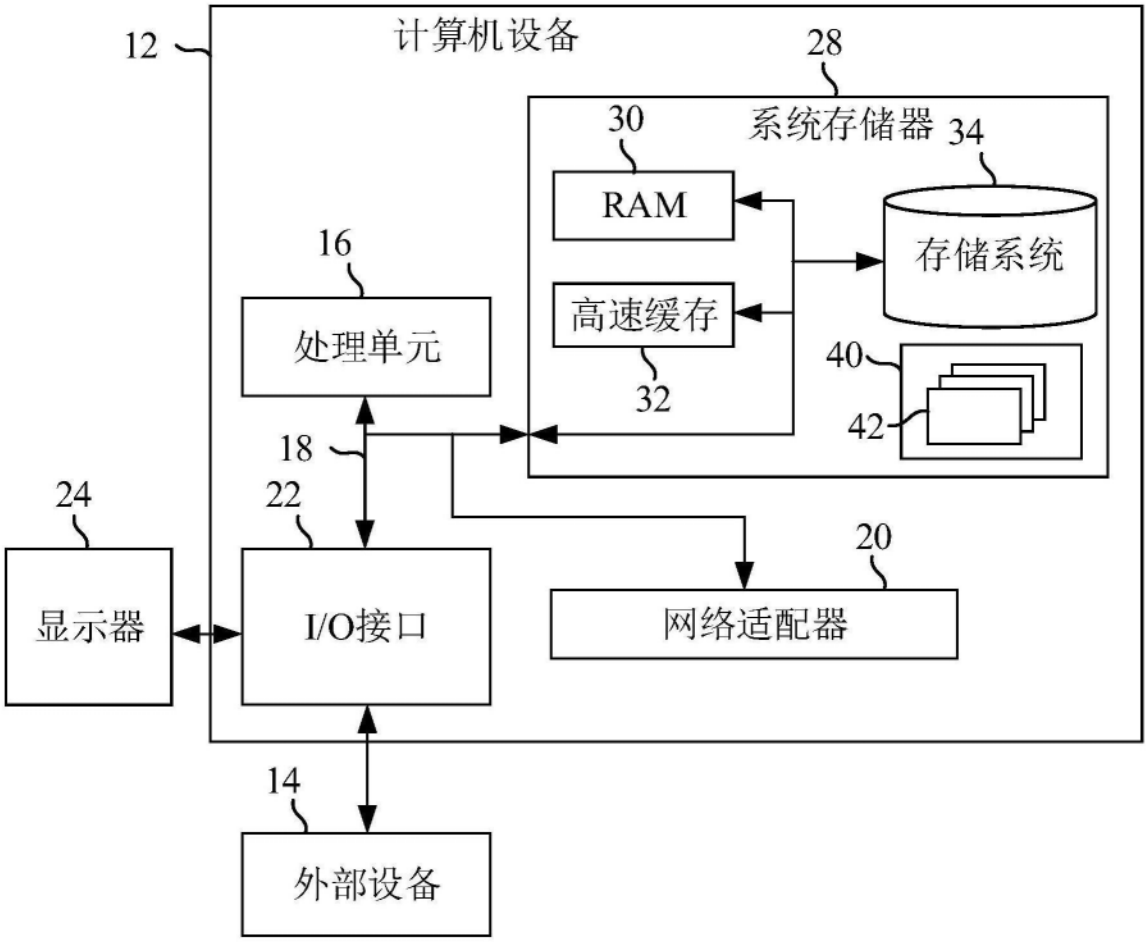

12
计算机设备
28
30
系统存储器
34
RAM
存储系统
16
处理单元
高速缓存
32
40
42
18
24
22
20
显示器
I/O接口
网络适配器
14
外部设备

图4