(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111198730 B
(45)授权公告日 2023.05.23

(21)申请号 201811279708.X
(22)申请日 2018.10.30
(65)同一申请的已公布的文献号
申请公布号 CN 111198730 A
(43)申请公布日 2020.05.26
(73)专利权人 广州腾讯科技有限公司
地址 510320 广东省广州市海珠区新港中路397号品牌区1号
(72)发明人 马良 卢强
(74)专利代理机构 北京派特恩知识产权代理有限公司 11270
专利代理师 王花丽 张颖玲
(51)Int.Cl.
G06F 9/451(2018.01)
G06F 9/448(2018.01)
G06F 9/445(2018.01)

(56)对比文件
CN 108108205 A,2018.06.01
CN 108664304 A,2018.10.16
CN 106528230 A,2017.03.22
US 6272674 B1,2001.08.07
US 6785891 B1,2004.08.31
CN 108536481 A,2018.09.14
CN 105975319 A,2016.09.28

审查员 张静

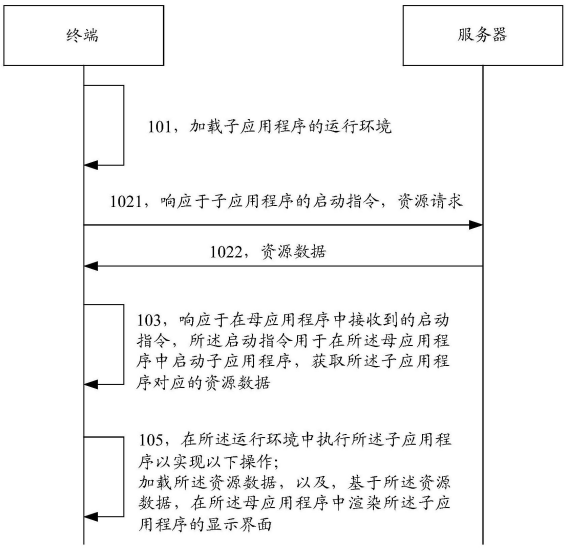
权利要求书2页 说明书14页 附图12页

(54)发明名称

子应用程序启动的方法、装置、终端及计算机存储介质

(57)摘要

本发明实施例提供一种子应用程序启动的方法、装置、终端及计算机存储介质，子应用程序启动的方法包括：预加载子应用程序的运行环境；响应于在母应用程序中接收到的启动指令，所述启动指令用于在所述母应用程序中启动子应用程序，获取所述子应用程序对应的资源数据；在所述运行环境中执行所述子应用程序以实现以下操作；加载所述资源数据，以及，基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。



CN 111198730 B

1.一种子应用程序启动的方法,其特征在于,包括:

预加载子应用程序的运行环境;

响应于在母应用程序中收到的启动指令,所述启动指令用于在所述母应用程序中启动子应用程序,获取所述子应用程序对应的资源数据;

在所述运行环境中执行所述子应用程序以实现以下操作;

加载所述资源数据,以及,

基于所述资源数据,在所述母应用程序中渲染所述子应用程序的显示界面。

2.如权利要求1所述子应用程序启动的方法,其特征在于,所述预加载子应用程序的运行环境,包括:

在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境,所预加载的运行环境的生命周期等同于所述母应用程序。

3.如权利要求2所述的子应用程序启动的方法,其特征在于,所述在所述母应用程序中,预加载持续运行于所述母应用程序中的运行环境,包括:

当所述母应用程序中包含有子应用程序历史记录时,在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境。

4.如权利要求1所述子应用程序启动的方法,其特征在于,所述预加载子应用程序的运行环境,包括:

在所述母应用程序的宿主操作系统中,预加载持续运行于所述宿主操作系统的所述运行环境,所预加载的运行环境的生命周期等同于所述宿主操作系统。

5.如权利要求1所述子应用程序启动的方法,其特征在于,所述获取所述子应用程序对应的资源数据,包括:

通过第一线程从服务器获取与所述子应用程序对应的首页界面数据;

通过第二线程从服务器获取所述子应用程序对应的后续页面数据,所述后续页面数据是用于显示除首页之外的页面;

其中,所述第一线程与所述第二线程是相互独立的线程。

6.如权利要求5所述的子应用程序启动的方法,其特征在于,所述加载所述资源数据,并基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面,包括:

响应于所述首页界面数据加载完成,基于所述首页界面数据在所述母应用程序中渲染首页界面;

响应于所述后续页面数据加载完成,基于所述后续页面数据在所述母应用程序中渲染后续页面以替换所述首页界面。

7.如权利要求6所述的子应用程序启动的方法,其特征在于,还包括:

响应于所述首页界面数据加载完成,记录所述首页界面数据下载完成时的时间并保存至指定数据库;和/或,

响应于所述首页界面数据加载完成,记录所述首页界面数据下载完成时的时间,并发送给所述子应用程序对应的开发者设备。

8.如权利要求1至7任一项所述的子应用程序启动的方法,其特征在于,所述方法还包括:

基于如下方式之一接收所述启动指令:

在所述母应用程序中加载所述子应用程序的设置列表页面，并接收所述启动指令，所述启动指令用于启动所述设置列表页面中处于选中状态的子应用程序；

在所述母应用程序中调用扫码接口，并接收所述启动指令，所述启动指令用于启动与所述扫码接口获取的路径参数适配的子应用程序；

在所述母应用程序的子应用程序检索页面中接收检索参数，并接收所述启动指令，所述启动指令用于启动与接收的检索参数适配的子应用程序；

在所述母应用程序中接收针对图形码的超出设定时长的操作，并接收所述启动指令，所述启动指令用于启动与所述图形码携带的路径参数适配的子应用程序。

9.一种子应用程序启动的装置，其特征在于，包括：

母应用程序模块，用于预加载子应用程序的运行环境；

所述母应用程序模块，用于响应于在母应用程序中接收到的启动指令，所述启动指令用于在所述母应用程序中子应用程序，获取所述子应用程序对应的资源数据；

引擎模块，用于在所述运行环境中执行所述子应用程序以实现以下操作；加载所述资源数据，以及，基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。

10.如权利要求9所述的装置，其特征在于，

所述母应用程序模块，具体用于在所述母应用程序中，预加载持续运行于所述母应用程序中的所述运行环境，所预加载的运行环境的生命周期等同于所述母应用程序。

11.如权利要求9所述的装置，其特征在于，

所述母应用程序模块，具体用于在所述母应用程序的宿主操作系统中，预加载持续运行于所述宿主操作系统的所述运行环境，所预加载的运行环境的生命周期等同于所述宿主操作系统。

12.如权利要求9所述的装置，其特征在于，

所述母应用程序模块，具体用于：

通过第一线程从服务器获取与所述子应用程序对应的首页界面数据；

通过第二线程从服务器获取所述子应用程序对应的后续页面数据，所述后续页面数据是用于显示除首页之外的页面；

其中，所述第一线程与所述第二线程是相互独立的线程。

13.如权利要求12所述的装置，其特征在于，

所述引擎模块，具体用于在所述运行环境中执行所述子应用程序以实现以下操作：

响应于所述首页界面数据加载完成，基于所述首页界面数据在所述母应用程序中渲染首页界面；

响应于所述后续页面数据加载完成，基于所述后续页面数据在所述母应用程序中进行渲染后续页面以替换所述首页界面。

14.一种终端，其特征在于，包括处理器和用于存储能够在处理器上运行的计算机程序的存储器；其中，

所述处理器用于运行所述计算机程序时，执行如权利要求1至8中任一项所述的子应用程序启动的方法。

15.一种计算机存储介质，其中存储有计算机程序，其特征在于，所述计算机程序被处理器执行时，实现如权利要求1至8中任一项所述子应用程序启动的方法。

子应用程序启动的方法、装置、终端及计算机存储介质

技术领域

[0001] 本发明涉及计算机通信技术，特别涉及一种子应用程序启动的方法、装置、终端及计算机存储介质。

背景技术

[0002] 随着终端技术及移动互联网技术的快速发展，开发者开发的应用的种类越来越多，用户使用的应用也越来越多。例如：手机、平板电脑等终端上安装的移动应用(MA，Mobile Application)越来越多，包括即时通讯应用、金融类应用、学习类应用、支付类应用、生活便利类的应用。用户想要享用某种服务如向商家付款，那么用户首先得安装具有支付功能的应用，然后打开应用享用服务，这种需要安装才能享用服务的方式越来越不适合快速、便捷和共享的生活理念。

[0003] 目前，知识、信息类的轻量化产品的共享可最大范围的共享和受益，出行共享、服务共享、实体经济的共享则会受到一定的地域或者时间的限制，此类服务性共享则需要寻找一个虚拟的节点，在这种情况下，小程序应运而生，小程序正是为快速、便捷和共享提供一种虚拟节点，这样便通过互联网的连接完成了广泛地共享。小程序是一种不需要下载安装即可使用的子应用程序，它实现了应用“触手可及”的梦想，用户无需下载安装即可打开应用。由于小程序本身并不需要安装和卸载，所以小程序也体现了“用完即走”的理念，用户不用关心是否安装太多应用的问题。

[0004] 然而，随着越来越多应用通过小程序的普及使用，在同一母应用程序中运行的小程序越来越多，随着小程序数量和功能的增多也存在小程序在母应用程序中的启动速度慢、启动耗时的问题。

发明内容

[0005] 为解决现有存在的技术问题，本发明实施例提供一种子应用程序启动的方法、装置、终端及计算机存储介质，能够加速子应用程序的启动，减少子应用程序启动前需要等待的时间。

[0006] 为达到上述目的，本发明实施例的技术方案是这样实现的：

[0007] 第一方面，本发明实施例提供一种子应用程序启动的方法，包括：预加载子应用程序的运行环境；响应于在母应用程序中接收到的启动指令，所述启动指令用于在所述母应用程序中启动子应用程序，获取所述子应用程序对应的资源数据；在所述运行环境中执行所述子应用程序以实现以下操作；加载所述资源数据，以及，基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。

[0008] 第二方面，本发明实施例提供一种子应用程序启动的装置，包括：母应用程序模块，用于预加载子应用程序的运行环境，响应于在母应用程序中接收到的启动指令，所述启动指令用于在所述母应用程序中启动子应用程序，获取所述子应用程序对应的资源数据；引擎模块，用于在所述运行环境中执行所述子应用程序以实现以下操作；加载所述资源数

据，以及，基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。

[0009] 第三方面，本发明实施例提供一种终端，包括存储器，用于存储可执行程序；处理器，用于通过执行所述存储器中存储的可执行程序时，实现本发明实施例提供的子应用程序启动的方法。

[0010] 第四方面，本发明实施例提供一种可读存储介质，存储有计算机程序，所述计算机程序被处理器执行时实现本发明实施例提供的子应用程序启动的方法。

[0011] 本发明实施例具有如下有益效果：

[0012] 通过响应于在母应用程序中接收对应子应用程序的启动指令之前，预加载子应用程序的运行环境，从而响应于启动指令，获取子应用程序对应的资源数据，并在运行环境中执行子应用程序，以加载资源数据以及基于资源数据在母应用程序中渲染子应用程序的显示界面，如此，获取到子应用程序对应的资源数据后，可以通过预加载的运行环境直接解释执行，以减少子应用程序启动时初始化运行环境的耗时，提升子应用程序的启动速度。

附图说明

[0013] 图1为本发明实施例提供的客户端运行子应用程序的一个可选的系统架构示意图；

[0014] 图2为本发明实施例提供的母应用程序中显示子应用程序页面的示意图；

[0015] 图3为本发明实施例提供的运行母应用程序的终端的结构示意图；

[0016] 图4为本发明实施例的子应用程序的组成结构示意图；

[0017] 图5为本发明另一实施例提供的运行母应用程序的终端的结构示意图；

[0018] 图6为本发明实施例中子应用程序启动的方法的流程示意图；

[0019] 图7为本发明另一实施例中子应用程序启动的方法的流程示意图；

[0020] 图8为本发明再一实施例中子应用程序启动的方法的流程示意图；

[0021] 图9为本发明又一实施例中子应用程序启动的方法的流程示意图

[0022] 图10A至图10D为本发明实施例中微信小程序启动的方法的界面示意图。

具体实施方式

[0023] 以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0024] 除非另有定义，本文所使用的所有的技术和科学术语与属于本发明的技术领域的技术人员通常理解的含义相同。本文中在本发明的说明书中所使用的术语只是为了描述具体的实施例的目的，不是旨在于限制本发明。本文所使用的术语“及/或”包括一个或多个相关的所列项目的任意的和所有的组合。

[0025] 在以下的描述中，涉及到“一些实施例”，其描述了所有可能实施例的子集，但是应当理解，“一些实施例”可以是所有可能实施例的相同子集或不同子集，并且可以在不冲突的情况下相互结合。

[0026] 对本发明进行进一步详细说明之前，对本发明实施例中涉及的名词和术语进行说明，本发明实施例中涉及的名词和术语适用于如下的解释。

[0027] 1)母应用程序，包括终端中运行的用于提供各种服务的应用程序，典型地如即时

通信类应用、支付类应用等。

[0028] 2)子应用程序,在母应用程序中运行并显示页面的程序,也称为小程序,是基于面向前端的语言(例如JavaScript)开发的、在超文本标记语言(HTML,Hyper Text Markup Language)页面中实现服务的程序,通过母应用程序下载后可以直接在母应用程序中解释执行,不需要下载安装即可使用,节省在母应用程序中安装步骤。

[0029] 3)小游戏,是小程序的一个游戏类目,可以实现即点即玩(运行)的技术效果,无需下载安装。

[0030] 4)网络脚本语言,一种不需要预编译而可以直接解释成机器语言来执行的语言,例如JavaScript,用来向前端中的HTML页面添加交互行为,例如游戏交互行为,可以直接整合到HTML页面中,也可实现为单独的JavaScript代码,有利于结构和行为的分离;具有跨平台特性从而支持在不同平台中运行,本发明实施例中简称js。

[0031] 5)网络脚本语言运行环境,是一个专门处理网络脚本语言的虚拟机,例如js引擎,用来将网络脚本语言解释成机器语言给终端中的硬件(例如CPU)执行。

[0032] 6)网页视图(Web-View)组件,也称为视图组件,用于实现网页中如点击、下拉框、表格等网页中各种内容的容器,负责子应用程序的显示,能够自动将内容在母应用程序中填充,以形成子应用程序的页面。

[0033] 7)事件,从子应用程序的视图层到逻辑层的通讯方式,网页视图组件中发生用户操作时所产生的数据以事件的形式传递到逻辑层,在逻辑层执行与事件绑定的事件处理函数,例如调用系统的各种能力接口。

[0034] 8)子应用程序公共库,子应用程序的能力需要对应的母应用程序来支撑,每一个基础库都只能在对应的母应用程序版本上运行。

[0035] 9)首页界面,用于表征相应的小程序的启动界面的图片,包含相应小程序的图标及名称。其中,首页界面数据可以包括有图片内容的数据、图片尺寸的数据、图片大小数据、图片位置的数据等等;显示除首页界面的任一页面的数据在本文中称为后续页面数据。

[0036] 10)响应于,用于表示所执行的操作所依赖的条件或者状态,当所依赖的条件或状态被满足时,所执行的一个或多个操作以实时的,也可以具有设定的延迟的方式执行;在没有特别说明的情况下,所执行的多个操作不存在执行先后顺序的限制。

[0037] 本发明实施例提供子应用程序启动的方法、装置、终端以及存储用于实现子应用程序启动的方法的计算机可读存储介质。就子应用程序启动的方法而言,本发明实施例提供终端侧实施的方案,请参阅图1,为本发明实施例提供的母应用程序运行子应用程序的一个可选的系统架构示意图。设置于终端200中的母应用程序242(如微信客户端WeChat)在首次运行子应用程序之前,发送初始化请求给母应用程序的服务器100来下载子应用程序。

[0038] 在一些实施例中,为了提高下载速度,请求可以被重定向至内容分发网络(CDN,Content Delivery Network)300,内容分发网络300将对请求分配合适的节点服务器(例如基于链路最短条件、负载均衡条件等选择的对应的节点服务器)进行响应,即,由内容分发网络300代替母应用程序的服务器100向母应用程序242发送子应用程序的数据包(Package),即子应用程序的源代码,从而实现子应用程序的下载。

[0039] 母应用程序242对子应用程序进行初始化完成后,加载子应用程序的运行环境且持续运行,当母应用程序242接收到运行子应用程序的指令时,在已经加载的运行环境解

释、并执行子应用程序，以实现子应用程序的页面的加载，包括首页界面和后续页面。

[0040] 子应用程序通过发送数据请求给服务器100，以获取对应的首页界面数据、后续页面数据及页面更新所需的数据。

[0041] 举例来说，数据请求可以采用AJAX请求的形式，AJAX即异步JavaScript和可扩展标记语言(eXtensible Markup Language)(Asynchronous JavaScript And XML)，服务器100基于子应用程序发送的AJAX请求返回相应的JavaScript对象标记(JSON，JavaScript Object Notation)格式的数据，进而子应用程序根据获取的JSON数据进行渲染页面，实现子应用程序在母应用程序242中的页面显示/更新。

[0042] 举例来说，母应用程序242在下载子应用程序时获取的子应用程序的源代码中包括以下不同类型的文件：1)，.json后缀的配置文件，用于子应用程序(全局)公共设置，决定页面文件的路径、窗口表现、设置网络超时时间、设置多标签(tab)等；2)，.js后缀的配置文件，用于实现子应用程序的(全局)逻辑；3)，.wxss后缀的配置文件，用于子应用程序的公共样式配置。

[0043] 作为终端200中设置母应用程序和子应用程序的示例，参见图2，在终端200中能够运行母应用程序242，子应用程序241用于在母应用程序242中实现模块化的功能，可以是用于扩展母应用程序242的功能，或者是用于替换母应用程序242中部分的功能；子应用程序241在母应用程序242中根据触发指令运行。图2中虽然只示出了母应用程序242中的一个子应用程序，但是可以理解，根据母应用程序实现功能的需求，母应用程序中可以运行多个子应用程序。

[0044] 接下来，再对运行母应用程序242的终端200的结构进行说明。

[0045] 参见图3，图3为本发明实施例提供的运行母应用程序242的终端200的一个结构示意图，终端200包括：硬件层21、驱动层22、操作系统层23和应用程序层24。然而应当理解，图3示出的终端200的结构仅为示例，并不构成对终端结构的限定。例如，终端可以根据实施需要设置较图3更多的组件，或者根据实施需要省略设置部分组件。

[0046] 硬件层21包括处理器211、存储器212、输入/输出接口213以及通信接口214，组件可以经系统总线连接通信。其中，处理器211可以采用中央处理器(CPU，Central Processing Unit)、微处理器(MCU，Microcontroller Unit)、专用集成电路(ASIC，Application Specific Integrated Circuit)或逻辑可编程门阵列(FPGA，Field－Programmable Gate Array)实现。

[0047] 输入/输出接口213可以采用如显示屏、触摸屏、扬声器等输入/输出器件实现。

[0048] 存储器212中存储有存储介质，存储介质可以采用闪存、硬盘、光盘等非易失性存储介质实现，也可以采用双倍率(DDR，Double Data Rate)动态缓存等易失性存储介质实现，其中存储有用以执行上述子应用程序中的网页处理方法的可执行程序。示例性地，存储介质可以与终端的其他组件在同一位置设置，也可以相对于终端中的其他组件分布设置，存储介质中至少包括操作系统、网络通信程序。

[0049] 通信接口214，向处理器211提供外部数据如异地设置的存储介质的访问功能。

[0050] 驱动层22，包括用于供操作系统识别硬件层21并与硬件层21各组件通信的中间件221，例如可以为针对硬件层21的各组件的驱动程序的集合。

[0051] 操作系统层23，用于提供面向用户的图形界面，示例性地，包括驱动程序231、图形

库232、系统框架233、用户界面套件234、插件图标、桌面背景和应用图标，操作系统层23支持用户经由图形界面对设备的控制本发明实施例对上述设备的软件，如操作系统类型、版本不做限定，例如可以是Linux操作系统、UNIX操作系统或其他操作系统。

[0052] 应用程序层24，包括子应用程序241及母应用程序242，当然，还可以包括用于实现其他常规功能的应用程序，如附带有js引擎243的网页浏览器等。

[0053] 接下来对本发明实施例中子应用程序241的架构进行说明，图4为本发明实施例提供的子应用程序的架构示意图，如图4所示，子应用程序242框架的核心是一个响应的数据绑定系统，整个系统分为两块，分别为视图(View)层及逻辑层，通过子应用程序框架可以绑定视图层和逻辑层相应的参数，如参数名称，让数据和视图非常简单的保持同步，当进行数据修改的时候，只需要在逻辑层修改数据，视图层就会做相应的更新。

[0054] 视图层和逻辑层在两个相互独立的线程中运行，二者之间的信息交互通过系统层的桥接器(JSBridge)进行。当用户在子应用程序的页面进行操作(如点击操作)，视图层的组件捕获用户操作后触发与该组件绑定的事件(如点击操作对应的点击事件)，通过JSBridge将事件发送至逻辑层，逻辑层对事件进行解析，执行对应逻辑(如调用子应用程序接口)并把进行页面更新所需数据通过JSBridge传递给视图层，视图层将接收的数据反应为视图。

[0055] 接下来分别对子应用程序242的视图层和逻辑层分别进行说明。

[0056] 视图层可以通过XML与层叠样式表(CSS，Cascading Style Sheets)编写，主要功能为将逻辑层的数据反应成视图，同时将视图层发生的事件发送给逻辑层；其中，XML用于描述页面的结构，CSS用于描述页面的样式。

[0057] XML是框架设计的一套标签语言，结合基础组件、事件系统，可以构建出页面的结构，XML用于子应用程序的数据绑定、列表渲染、条件渲染、提供模板、事件、引用。CSS是一套样式语言，用于描述XML的组件样式，决定XML的组件的显示效果。

[0058] 组件(Component)是视图层的基本组成单元，自带一些功能与样式，一个组件通常包括“开始标签”和“结束标签”，“属性”用来修饰这个组件，“内容”在两个标签之内。其中，组件共有的属性包括：用于唯一的标识组件(ID)、组件的样式类(Slass)、组件的内联样式(Style)、组件是否显示(Hidden)、自定义属性(data-*)，可以对该组件的功能或样式进行修饰、组件的事件(bind*/catch*)。

[0059] 基础组件分为以下八类：视图容器(View Container)、基础内容(如图标、文字、进度条)、表单(如按钮、输入框、标签等)、导航(Navigation)、多媒体(如音频，图片，视频)、地图(Map)、画布(Canvas)和客服会话。

[0060] 逻辑层对数据进行处理后发送给视图层，同时接收视图层的事件反馈，由JavaScript编写，在JavaScript基础上，增加了APP和Page方法，进行程序和页面的注册；提供了丰富的应用程序接口(API)，如播放音乐、预览图片等媒体API，上传文件、下载文件等网络API、保存文件、打开文件等文件API，获取本地数据缓存、清除本地数据缓存等数据API，获取当前位置、打开内置地图等位置API、获取系统信息、拨打电话等设备API，显示提示框、显示菜单列表等界面API，再如扫一扫、支付等；每个页面拥有独立的作用域(在JavaScript文件中声明的变量和函数只在该文件中有效；不同的文件中可以声明相同名字的变量和函数，不会互相影响)，并提供模块化(将一些公共的代码抽离成为一个单独的js

文件,作为一个模块)能力。

[0061] 事件是子应用程序的视图层到逻辑层的通信方式,可以将用户的页面操作反馈到逻辑层进行处理,可以绑定在视图层的组件上,当事件被触发,通过将事件传递到逻辑层,执行逻辑层中对应事件的事件处理函数,而事件对象(组件触发事件时,逻辑层绑定该事件的处理函数会收到一个事件对象)可以携带额外信息,如序列号(ID),数据集(dataset)和事件(touches)等。

[0062] 在一些实施例中,如图5所示为本发明实施例提供的运行母应用程序242的终端200的一个结构示意图,其与图3所示终端的区别在于,应用程序层24中,母应用程序242不包括附带js引擎的网页浏览器,而操作系统层23包括运行于操作系统层级的js引擎237。

[0063] 上述对实现本发明实施例的终端、母应用程序和子应用程序的架构进行了说明,以下继续说明实现本发明实施例的子应用程序启动的装置,用于实现本发明实施例子应用程序启动的装置可以采用硬件结合软件实施的方式,如图3示出终端200中处理器211可以处理器中的硬件及软件模块组合执行完成。

[0064] 作为子应用程序启动的装置的软件实施示例,例如,可以实施为母应用程序模块和引擎模块,母应用程序模块,用于预加载子应用程序的运行环境;响应于在母应用程序中接收到的对应所述子应用程序的启动指令,获取所述子应用程序对应的资源数据。引擎模块,用于在js引擎的运行环境中执行子应用程序以实现以下操作;加载资源数据,以及基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。

[0065] 请再参阅图3和图5,母应用程序模块可以是母应用程序242,引擎模块可以是图3中示出的js引擎243(运行于母应用程序242中,与母应用程序242具有相同的生命周期),也可以是图5中示出的js引擎237(运行于操作系统中,与操作系统具有相同的生命周期)。

[0066] 下面以图3和图5为例,说明子应用程序启动的装置中各模块的功能。

[0067] 在一些实施例中,请参阅图3,母应用程序242具体用于在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境,即js引擎243,所预加载的运行环境的生命周期等同于所述母应用程序。

[0068] 在一些实施例中,母应用程序242具体用于当所述母应用程序中包含有子应用程序历史记录时,在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境。

[0069] 在一些实施例中,请参阅图5,母应用程序242具体用于在所述母应用程序的宿主操作系统中,预加载持续运行于所述宿主操作系统的所述运行环境,所预加载的运行环境的生命周期等同于所述宿主操作系统。

[0070] 在一些实施例中,母应用程序242具体用于通过第一线程从服务器获取与所述子应用程序对应的首页界面数据;通过第二线程从服务器获取所述子应用程序对应的后续页面数据,所述后续页面数据是用于显示除首页之外的页面;其中,所述第一线程与所述第二线程是相互独立的线程。

[0071] 在一些实施例中,js引擎具体用于通过执行子应用程序实现以下操作:响应于所述首页界面数据加载完成,基于所述首页界面数据在所述母应用程序中渲染首页界面;响应于所述后续页面数据加载完成,基于所后续页面数据在所述母应用程序中渲染后续页面以替换首页界面。

[0072] 在一些实施例中，母应用程序242具体用于响应于所述首页界面数据加载完成时，记录所述首页界面数据下载完成时的时间并保存至指定数据库；和/或，响应于所述首页界面数据加载完成时，记录所述首页界面数据下载完成时的时间，并发送给所述子应用程序对应的开发者设备。

[0073] 在一些实施例中，母应用程序242具体用于基于如下方式之一接收所述启动指令：

[0074] 方式1)在所述母应用程序中加载所述子应用程序的设置列表页面，并接收所述启动指令，所述启动指令用于启动所述设置列表页面中处于选中状态的子应用程序；

[0075] 方式2)在所述母应用程序中调用扫码接口，并接收所述启动指令，所述启动指令用于启动通过所述扫码接口获取的路径参数适配的子应用程序；

[0076] 方式3)在所述母应用程序中接收检索参数，并接收所述启动指令，所述启动指令用于启动与在子应用程序检索页面中接收的检索参数相应的子应用程序；

[0077] 方式4)在所述母应用程序中接收针对图形码的超出设定时长的操作，并接收所述启动指令，所述启动指令用于启动与所述图形码携带的路径参数适配的子应用程序。

[0078] 本发明实施例还提供了一种计算机可读存储介质，如图3所示的包括可执行计算机程序的存储器212，上述计算机程序可由处理器211执行，以完成前述子应用程序启动的方法的步骤。可读存储介质可以是磁性随机存取存储器(FRAM)、只读内存(ROM)、可编程只读存储器(PROM)、非易失性只读存储器(EPROM)、带电可插可编程只读存储器(EEPROM)、闪存(Flash Memory)、磁表面存储器、光盘、或光盘只读存储器(CD-ROM)等存储器；也可以是包括上述存储器之一或任意组合的各种设备，如计算机设备。

[0079] 参阅图6，为本发明实施例提供的子应用程序启动的方法的可选的流程示意图，可应用于图3或图5所示的终端200，将结合下面的步骤进行说明。

[0080] 步骤101，母应用程序预加载子应用程序的运行环境。

[0081] 终端在接收子应用程序的启动指令之前，预先加载好的子应用程序的运行环境，从而在接收到子应用程序的启动指令后，可以根据启动指令和预先加载好的子应用程序的运行环境获取子应用程序对应的资源数据，在子应用程序的运行环境对资源数据解释执行，降低子应用程序的启动阶段初始化运行环境的耗时。这里，子应用程序的运行环境可以是指js代码解释执行所需的运行环境，也即js引擎。其中，获取子应用程序对应的资源数据是指从母应用程序的服务器获取子应用程序对应的资源数据。可选的，所述获取子应用程序对应的资源数据也可以是指从母应用程序的指定客户端获取所述子应用程序对应的资源数据。

[0082] 步骤103，母应用程序响应于接收到的启动指令，所述启动指令用于在所述母应用程序中启动子应用程序，获取所述子应用程序对应的资源数据。

[0083] 这里，步骤103之前，还可以包括：步骤1021，母应用程序响应于接收到的对应所述子应用程序的启动指令，向服务器发送资源请求；步骤1022，服务器向母应用程序发送资源数据。

[0084] 子应用程序对应的资源数据包括：子应用程序在启动过程中所显示的首页界面数据，子应用程序在启动后所需显示的后续页面数据，所述后续页面数据是用于显示除首页之外的页面。所述资源数据如可以包括子应用程序对应的首页界面数据、后续页面及页面更新所需的后续页面数据。响应于在母应用程序中接收对应所述子应用程序的启动指令，

服务器向母应用程序发送对应的子应用程序的数据包(Package),即所述子应用程序的源代码,从而实现子应用程序的下载。

[0085] 步骤105,在所述运行环境中执行所述子应用程序以实现操作。

[0086] 所实现的操作包括:加载所述资源数据,以及,基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面。

[0087] 在运行环境中执行子应用程序,母应用程序通过已经加载好的运行环境解释子应用程序的源代码来进行子应用程序的资源数据的加载,并子应用程序根据获取的资源数据进行页面渲染,实现子应用程序在母应用程序中的页面显示/更新。

[0088] 本发明上述实施例中,终端在接收子应用程序的启动指令之前,预先加载好的子应用程序的运行环境,从而在接收到子应用程序的启动指令后,可以根据启动指令和预先加载好的子应用程序的运行环境对获取的子应用程序对应的资源数据解释执行,降低子应用程序的启动阶段初始化运行环境的耗时。与已知的技术中,常用的在应用启动的过程中,通过加载一个显示正在加载中的加载图的方式而言,本发明实施例通过预先加载好子应用程序的运动环境,做到了真正意义上的预加载,从本质上提升了子应用程序的启动速度。

[0089] 在一些实施例中,步骤101,预加载子应用程序的运行环境,包括:

[0090] 在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境,所预加载的运行环境的生命周期等同于所述母应用程序。

[0091] 请参阅图7,所述预加载子应用程序的运行环境可以是:步骤1011,母应用程序预加载持续运行于母应用程序中的运行环境。其中,在所述母应用程序中预加载持续运行于所述母应用程序中的运行环境可以是指,在母应用程序启动时,即预加载母应用程序中的子应用程序的运行环境。如图3所示终端示例中,母应用程序中包括附带有子应用程序的运行环境的网页浏览器,该子应用程序的运行环境可以是指js引擎。通过在母应用程序启动时,通过启动预加载进程对子应用环境的运行环境进行加载,实现在子应用程序启动之前对子应用程序的运行环境的预加载。预加载的运行环境的生命周期等同于所述母应用程序,是指运行于母应用程序中的运行环境与母应用程序的开启和关闭的触发条件相同。

[0092] 本实施例中,通过在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境,母应用程序响应于子应用程序的启动指令,并获取到服务器发送的资源数据后,所述步骤105可以包括:步骤1051,母应用程序在运行环境中加载所述资源数据;步骤1052,母应用程序将加载后的所述资源数据发送给对应的子应用程序,步骤1053,子应用程序基于加载后的资源数据,在母应用程序中渲染对应的显示界面。

[0093] 在一些实施例中,所述在所述母应用程序中,预加载持续运行于所述母应用程序中的运行环境,包括:

[0094] 当所述母应用程序中包含有子应用程序历史记录时,在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境。

[0095] 母应用程序启动时,可以通过判断所述母应用程序中包含有运行任一子应用程序的历史记录、或者包含有运行特定类型的子应用程序的历史记录而对母应用程序是否需要预加载相应的运行环境进行预判,当确定所述母应用程序中包含有任一子应用程序历史记录或者有运行特定类型的子应用程序的历史记录时,则在所述母应用程序中,预加载持续运行于所述母应用程序中的所述运行环境,如此,可以对是否需要预加载子应用程序的运

行环境的场景进行识别和区分。其中，该特定类型的子应用程序可以是用户配置的部分类型的子应用程序，如生活便利类应用、也可以是根据设置规则而自动匹配的部分类型的子应用程序，如运行记录的次数高于一定值、或者与运行记录的次数高于一定值的类型相同的其它子应用程序。通常，母应用程序中包含有子应用程序历史记录可以表征存在用户在母应用程序中启动子应用程序的需求，通过对是否需要预加载子应用程序的运行环境的场景进行识别和区分，可以节省系统开销。

[0096] 在一些实施例中，步骤101，预加载子应用程序的运行环境，包括：

[0097] 在所述母应用程序的宿主操作系统中，预加载持续运行于所述宿主操作系统的所述运行环境，所预加载的运行环境的生命周期等同于所述宿主操作系统。

[0098] 请参阅图8，所述预加载子应用程序的运行环境可以是：步骤1013，母应用程序的宿主操作系统预加载持续运行于操作系统层的运行环境。其中，在所述母应用程序的宿主操作系统中，预加载持续运行于所述宿主操作系统的运行环境可以是指，在操作系统启动时，即预加载持续运行于母应用程序的宿主操作系统层的运行环境。如图5所示终端示例中，操作系统层包括附带有子应用程序的运行环境的网页浏览器，该子应用程序的运行环境可以是指js引擎。通过在操作系统启动时，即对子应用环境的运行环境进行加载，实现在子应用程序启动之前对子应用程序的运行环境的预加载。预加载的运行环境的生命周期等同于所述宿主操作系统，是指运行于宿主操作系统中的运行环境与宿主操作系统的开启和关闭的触发条件相同，如终端开机时，操作系统开启，同步预加载运行于母应用程序的宿主操作系统的子应用程序的运行环境，终端关机时，操作系统关闭，同步关闭运行于母应用程序的宿主操作系统的运行环境。通过预加载持续运行于所述宿主操作系统的运行环境，可以适用于母应用程序中无子应用程序的运行环境的使用场景。

[0099] 本实施例中，通过在所述母应用程序的宿主操作系统中，预加载持续运行于所述宿主操作系统的所述运行环境后，该方法还包括：步骤1014，操作系统检测打开母应用程序的点击操作，步骤1020，当检测到点操作时，母应用程序启动。所述母应用程序响应于资源数据下载完成，子应用程序基于资源数据渲染显示界面包括：步骤1050，母应用程序向子应用程序发送首页界面数据下载完成的通知消息，步骤1053，子应用程序基于所述首页界面数据在所述母应用程序中渲染首页界面；步骤1055，母应用程序向子应用程序发送首页界面数据下载完成的通知消息，步骤1056，子应用程序基于所述后续页面数据在所述母应用程序中渲染后续页面以替换所述首页界面。

[0100] 在一些实施例中，请继续参阅图8，所述获取所述子应用程序对应的资源数据，包括：

[0101] 步骤1032，通过第一线程从服务器获取与所述子应用程序对应的首页界面数据；

[0102] 步骤1033，通过第二线程从服务器获取所述子应用程序对应的后续页面数据，所述后续页面数据是用于显示除首页之外的页面；

[0103] 其中，所述第一线程与所述第二线程是相互独立的线程。

[0104] 也即，通过独立的第一线程和第二线程从服务器分别获取与所述子应用程序对应的首页界面数据和后续页面数据。本实施例中，母应用程序响应于子应用程序的启动指令，向服务器发送资源请求之后，服务器响应于所述资源请求向母应用程序发送资源数据包括：步骤1024，服务器向母应用程序发送首页界面数据；以及步骤1025，服务器向母应用程

序发送后续页面数据。相应的，母应用程序从服务器获取所述子应用程序对应的资源数据可以包括通过独立的第一线程从服务器获取与所述子应用程序对应的首页界面数据、以及通过独立的第二线程并行从服务器获取与所述子应用程序对应的后续页面数据。

[0105] 子应用程序对应的资源数据可以分离为首页界面数据和后续页面数据，通过独立的第一线程和第二线程分别获取首页界面数据和后续页面数据，从而接收到用户针对子应用程序的启动指令后，在子应用程序完全启动之前，可以首先快速拉取该子应用程序对应的首页界面数据，在预加载的运行环境中解释执行，渲染后快速并显示在显示界面中，并同时通过独立的线程下载子应用程序对应的后续页面数据，在所述后续页面数据下载完成时，在预加载的运行环境中解释执行，渲染后绘制该子应用程序对应的后续页面，并将所述后续页面替换所述首页图片进行更新显示，完成子应用程序的启动过程。其中，通过独立的线程从服务器分别获取与所述子应用程序对应的首页界面数据和后续页面数据，可以是指通过独立的第一线程和第二线程并行下载该首页界面数据和后续页面数据，如此，可以减少获取子应用程序对应的资源数据所需时间。

[0106] 由于子应用程序对应的首页界面数据的大小会明显小于后续页面数据的大小，通过将子应用程序的首页界面数据从资源数据进行分离，并通过独立的第一线程和第二线程分别并行获取首页界面数据和后续页面数据，从而可以快速拉取明显小于后续页面数据的首页界面数据后显示，首页界面数据的快速拉取后显示，从而可以减少用户等待的时间，使得子应用程序能够具备更快响应速度的体验。其中，该首页界面数据可以是指设置的首页图片数据，首页图片是指用于表征相应子应用程序的启动界面的符合统一标准的图片，首页图片数据可以包括有图片内容的数据，如对应的子应用程序的图标及名称、图片尺寸的数据、图片大小数据、图片位置的数据等等。

[0107] 其中，将子应用程序的首页界面数据从资源数据进行分离可以是从子应用程序的平台侧设置。如，在开发者通过平台侧发布子应用程序时，获取到子应用程序的发布指令时，确定是否接收到包含有所述子应用程序对应的设置首页界面数据的资源数据；当确定接收到包含所述首页界面数据的资源数据时，形成携带有所述首页图片的链接信息的所述子应用程序对应的启动入口。

[0108] 形成携带有首页图片的链接信息的所述子应用程序对应的启动入口，是指基于启动入口启动该子应用程序时，则相应可以同步获取到首页图片的链接信息。这里，还可以是指将子应用程序对应的资源数据分离为首页界面数据和后续页面数据，从而形成子应用程序对应的启动入口可以包括独立的首页图片的链接信息和后续页面数据的链接信息，基于启动入口启动该子应用程序时，则相应同步分别获取到首页图片的链接信息和后续页面数据的链接信息，并通过独立的线程分别下载首页界面数据和后续页面数据。

[0109] 子应用程序的发布平台在获取到子应用程序的发布指令时，通过确定是否接收到包含有子应用程序对应的设置首页界面数据的资源数据，确定接收到包含所述首页界面数据的资源数据时，形成携带有所述首页图片的链接信息的所述子应用程序对应的启动入口。如此，一方面，通过将子应用程序对应的首页界面数据从资源数据中分离出来，在通过子应用程序的启动入口，子应用程序时被启动后，可以通过独立的线程快速拉取该分离的首页界面数据，从而可以加速该子应用程序对应的首页界面数据的获取速度；另一方面，通过于子应用程序发布时，将包含有所述首页界面数据的资源数据作为发布前的必要满足的

条件，相当于在小程序平台侧制定统一标准，使得开发者在小程序发布时，必须统一提供小程序首页图片的图片链接的开发方式，可以降低游戏开发者编码出错的概率。

[0110] 在一些实施例中，在步骤105中，所述加载所述资源数据，并基于所述资源数据在所述母应用程序中渲染所述子应用程序的显示界面，包括：

[0111] 响应于所述首页界面数据加载完成，基于所述首页界面数据在所述母应用程序中渲染首页界面；

[0112] 响应于所述后续页面数据加载完成，基于所述后续页面数据在所述母应用程序中渲染后续页面以替换所述首页界面。

[0113] 由于首页界面数据和后续页面数据是通过独立的线程分别获取，响应于首页界面数据下载完成时，首先绘制首页界面并显示，该首页界面可以是首页图片，通过将首页界面数据从子应用程序对应的资源数据中整体分离出来，并通过独立的线程获取，可以快速获得该子应用程序对应的首页图片并显示，从而可以在子应用程序完全启动之前，通过首先快速拉取相应的子应用程序对应的首页图片并显示在显示界面中，减少用户等待的时间。

[0114] 通过将子应用程序对应的资源数据可以分离为首页界面数据和后续页面数据，并通过独立的线程分别获取首页界面数据和后续页面数据，需要说明的是，由于子应用程序对应的首页界面数据的大小会明显小于后续页面数据的大小，通过独立的线程并行分别获取首页界面数据和后续页面数据时，能够更加迅速地获取到首页界面并呈现给用户，当子应用程序的后续页面数据，也即子应用程序启动后的页面显示数据全部加载完成后，再相应绘制子应用程序启动后的后续页面数据并替换该首页图片，从而可以将子应用程序对应的首页图片更加快速的呈现给用户，减少用户等待的时间，使得子应用程序能够具备更快响应速度的体验。

[0115] 在一些实施例中，该方法还包括：

[0116] 响应于所述首页界面数据加载完成，记录所述首页界面数据下载完成时的时间并保存至指定数据库；和/或，

[0117] 响应于所述首页界面数据加载完成，记录所述首页界面数据下载完成时的时间，并发送给所述子应用程序对应的开发者设备。

[0118] 这里，指定数据库可以是指位于服务器的远程数据库。终端可以通过记录所述首页界面数据下载完成时的时间并保存至指定数据库，子应用程序的开发者可以通过该指定数据库获取到相应的子应用程序的首页界面下载完成时的时间以及对应的接收子应用程序的启动指令的时间，从而得到相应的子应用程序的首页界面数据的下载所需时间，通过对首页界面数据的下载所需时间进行分析统计，便于开发者根据分析统计的结果进行监控和优化。可选的，终端也可以记录接收子应用程序的启动指令的时间和所述首页界面数据下载完成时的时间，并发送给子应用程序对应的开发者设备，便于该子应用程序的开发者可以获取到相应的子应用程序的首页界面数据的下载所需时间，通过对首页界面数据的下载所需时间进行分析统计，根据分析统计的结果了解相应子应用程序的启动性能，进行监控和优化。

[0119] 在一些实施例中，所述方法还包括：

[0120] 基于如下方式之一接收所述启动指令：

[0121] 在所述母应用程序中加载所述子应用程序的设置列表页面，并接收启动指令，所

述启动指令用于启动所述设置列表页面中处于选中状态的子应用程序；

[0122] 在所述母应用程序中调用扫码接口，并接收启动指令，所述启动指令用于启动与所述扫码接口获取的路径参数适配的子应用程序；

[0123] 在所述母应用程序的子应用程序检索页面中接收检索参数，并接收启动指令，所述启动指令用于启动与接收的检索参数适配的子应用程序；

[0124] 在所述母应用程序中接收针对图形码的超出设定时长的操作，并接收启动指令，所述启动指令用于启动与所述图形码携带的路径参数适配的子应用程序。

[0125] 其中，子应用程序的启动入口可以包括多种不同的形式，比如可以是携带有子应用程序对应的路径参数的二维码信息、携带有与子应用程序对应的图标的列表页面、携带有与子应用程序对应的图标的检索页面、与子应用程序对应图形码等。终端可以通过子应用程序的启动入口，接收子应用程序的启动指令可以包括：终端检测到用户对子应用程序的启动入口的操作时，则接收到子应用程序的启动指令。比如，终端通过子应用程序的设置启动入口，接收子应用程序的启动指令，主要包括如下几种方式。

[0126] 第一，通过母应用程序提供的子应用程序的设置列表页面，检测到针对所述列表页面中某一子应用程序的触控操作时，接收到所述子应用程序的启动指令，其中，母应用程序提供的子应用程序的设置列表页面，可以是携带有子应用程序的图标的列表信息，以母应用程序为微信为例，通过微信->发现->小程序，可以进入到小程序的列表页面，即为携带有子应用程序的设置列表页面。

[0127] 第二，通过在母应用程序的扫码模式下获取到子应用程序对应的路径参数时，接收到所述子应用程序的启动指令，其中，在所述母应用程序中调用扫码接口，母应用程序在扫码模式下启动照相机，当扫描到子应用程序对应的二维码信息时，则获取到子应用程序对应的路径参数，即根据该路径参数启动子应用程序的启动程序。

[0128] 第三，在母应用程序的搜索模式下获取到子应用程序对应的检索参数，基于所述检索参数返回携带有相应子应用程序列表信息的检索页面，检测到针对所述检索页面中子应用程序的确定打开操作时，接收到所述子应用程序的启动指令，其中，母应用程序在搜索模式下，接收用户输入的检索参数，并根据检索参数返回显示有符合检索条件的子应用程序的列表信息的检索页面，用户可以通过选取检索页面中的目标子应用程序确认启动、或者在检索页面中符合检索条件的子应用程序即为目标子应用程序时，通过在搜索模式下点击确认按键以表征对目标子应用程序确认启动，以母应用程序为微信为例，通过微信->小程序->点击放大镜->输入检索参数，可以进入到微信的小程序搜索模式并输入检索参数，在搜索栏下方即可以显示根据检索参数命中的小程序。

[0129] 第四，检测到针对携带有子应用程序对应的路径参数的图形码的设置时长的触控操作时，接收到所述子应用程序的启动指令，其中，设置时长的触控操作可以理解为长按操作、也可以是指一般的点击操作。通过对子应用程序对应的二维码信息的长按，启动对应的子应用程序。该子应用程序对应的图形码可以是包含有子应用程序对应的路径参数的图片，用户可以基于母应用程序的会话窗口接收子应用程序的图形码，或者，用户可以将子应用程序的图形码发送至指定的位置，终端通过检测用户对子应用程序对应的图形码的长按操作，则接收到对应的子应用程序的启动指令。

[0130] 为了能够更便于对本发明实施例所提供的子应用程序启动的方法的实现方式的

理解，以下以母应用程序为微信，子应用程序为小程序为例，结合参阅图9、图10A至图10D，对子应用程序启动的方法进行进一步说明。

[0131] 预加载子应用程序的运行环境。其中，预加载子应用程序的运行环境可以包括：方案一，操作系统启动操作系统层面的浏览器模块；或者方案二，包含有浏览器模块的母应用程序启动母应用程序中的浏览器模块。该浏览器模块包含子应用程序的运行环境，如js引擎。在方案一中，操作系统检测到终端的开机指令时，通过操作系统激活子应用程序初始化进程，加载子应用程序的运行环境且持续运行，也即，以子应用程序的运行环境为js引擎为例，可以通过终端从操作系统层面提供一直处于活跃状态的js引擎，如此，以母应用程序为微信、子应用程序为小程序为例，在后续启动微信中的小程序时，可以根据小程序的启动指令和预先加载好的子应用程序的运行环境，从网页后台获取相应的资源数据后即解释执行。在方案二中，操作系统检测针对母应用程序的点击操作，当检测到点击操作时，母应用程序启动，母应用程序加载子应用程序的运行环境且持续运行，以母应用程序为微信、子应用程序为小程序为例，当打开微信时，通过不同的进程预先拉起小程序初始化进程和微信启动进程，从而实现打开微信时即启动小程序初始化进程，通过初始化进程加载子应用程序的运行环境且持续运行，如此，在后续启动微信中的小程序时，可以根据小程序的启动指令和预先加载好的子应用程序的运行环境，从网页后台获取相应的资源数据后即解释执行，如图10A所示，示出了在打开微信时，同步启动微信中的浏览器模块，进行初始化操作的步骤。

[0132] 操作系统检测打开母应用程序的点击操作。操作系统检测是否有点击包含浏览器模块的母应用程序的操作，当检测到点击操作后，母应用程序启动。如上所述，在方案一中，是操作系统启动操作系统层面的浏览器模块之后，再检测打开母应用程序的点击操作；在方案二中，操作系统检测打开母应用程序的点击操作后，母应用程序启动和启动母应用程序中的浏览器模块通过并列的进程同步执行。

[0133] 母应用程序启动后，获取用户输入或点击网址。以微信为例，获取用户输入网址可以包括：当微信被用户触发运行后，微信将跳转到微信发现页面，在微信发现页面可以看到如图框中标注的“XX小程序”功能项目；如没有显示用户欲运行的小程序，则可以点击右上方的“放大镜”按钮，进入子应用程序检索页面，在搜索栏中输入小程序名称作为检索信息，微信将根据关键词搜索对应的小程序。在小程序搜索模式下，获取在搜索模式中输入的检索信息，如图10B所示。

[0134] 母应用程序通过第一线程从网页后台下载首页界面资源、以及第二线程从网页后台下载其它资源；当首页界面资源下载完成时，子应用程序绘制显示首页；当其它资源下载完成时，子应用程序绘制显示后续页面。这里，网页后台是指母应用程序对应的服务器，其它资源是指子应用程序对应的除首页界面之外并在子应用程序启动后显示的其它页面资源。用户通过在母应用程序中选取目标小程序进行点击之后，即快速拉取相应目标小程序的首页界面数据，当首页界面数据下载完成时，子应用程序通过已加载好的js引擎解释执行并绘制对应的首页图片显示，如图10C所示，用户选取小程序列表页面中的“XX小程序”启动之后，可以快速拉取到“XX小程序”的首页图片并显示。如图10D所示，其它资源是指小程序启动后对应显示的页面数据，当小程序对应的后续页面数据全部下载完成时，绘制后续页面替换首页图片显示。

[0135] 母应用程序还可以记录首页界面资源下载完成时的时间给到开发者，可以是记录首页图片下载完成时的时间并发送到开发者设备，以供小程序的开发者获取用于分析统计，根据分析统计的结果对对应子应用进程进行监控/优化。

[0136] 综上所述，本发明实施例中，用户在开机或打开母应用程序的同时，进行了平台资源的准备和js引擎的初始化操作，用户在母应用程序中输入子应用程序的网址后，从母应用程序对应的服务器同步下载子应用程序的首页界面资源和其它资源。如，在打开微信时，即完成小程序的运行环境的预加载操作，确保降低后续小程序启动时初始化运行环境的耗时，小程序对应的资源数据分离为首页界面数据和后续页面数据，通过独立的线程分别获取首页界面数据和后续页面数据，从而接收到用户针对小程序的启动指令后，可以首先快速拉取该小程序对应的首页界面并显示在显示界面中，并同时下载小程序对应的后续页面数据，在所述后续页面数据下载完成时，在预加载的运行环境中解释执行并绘制对应的显示页面并将所述显示页面替换首页图片进行更新显示，完成小程序的启动过程，通过独立的线程并行获取首页界面数据和后续页面数据，能够更加迅速地获取到小程序的首页图片呈现给用户，从而可以减少用户等待的时间，使得小程序能够具备更快响应速度的体验。

[0137] 以上所述，仅为本发明的具体实施方式，但本发明的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本发明揭露的技术范围内，可轻易想到变化或替换，都应涵盖在本发明的保护范围之内。因此，本发明的保护范围应以所述权利要求的保护范围为准。

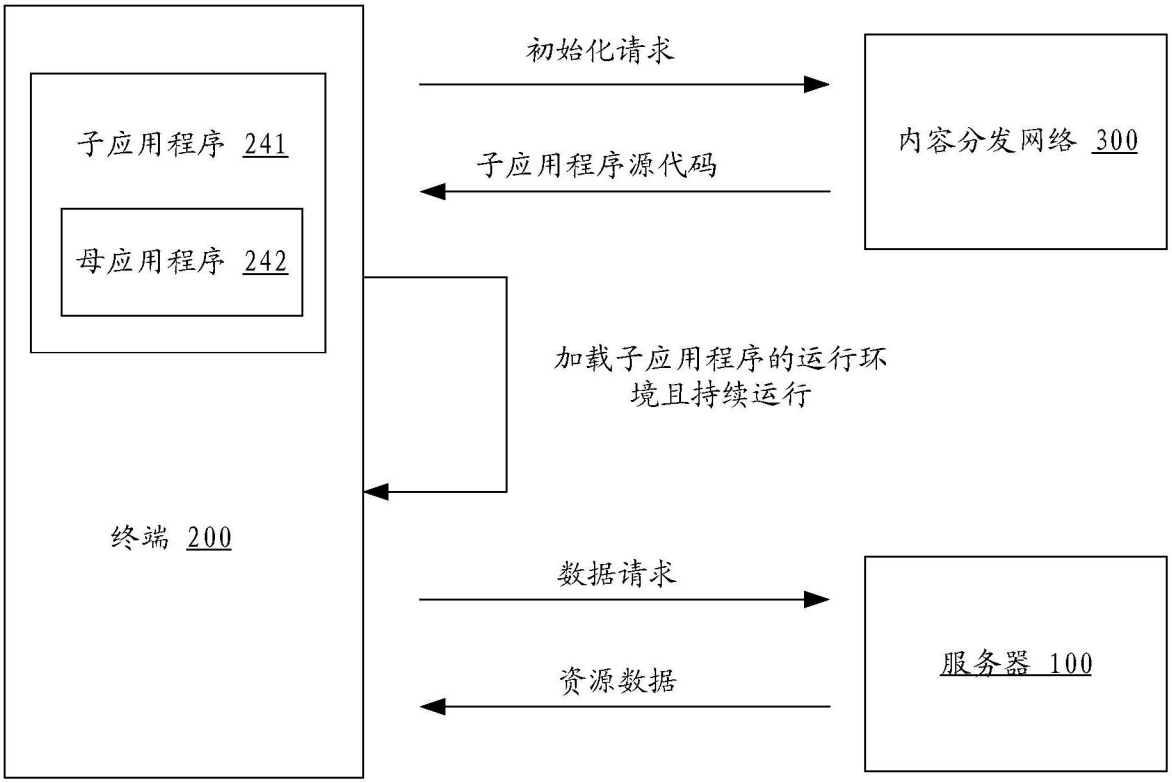

初始化请求
子应用程序源代码
内容分发网络 300
子应用程序 241
母应用程序 242
加载子应用程序的运行环境且持续运行
终端 200
数据请求
资源数据
服务器 100

图1

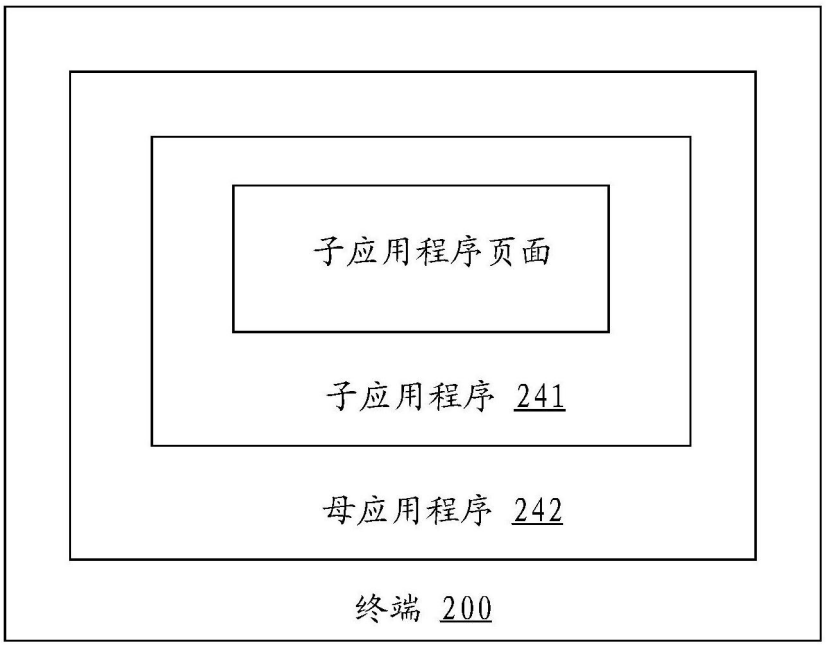

子应用程序页面
子应用程序 241
母应用程序 242
终端 200

图2

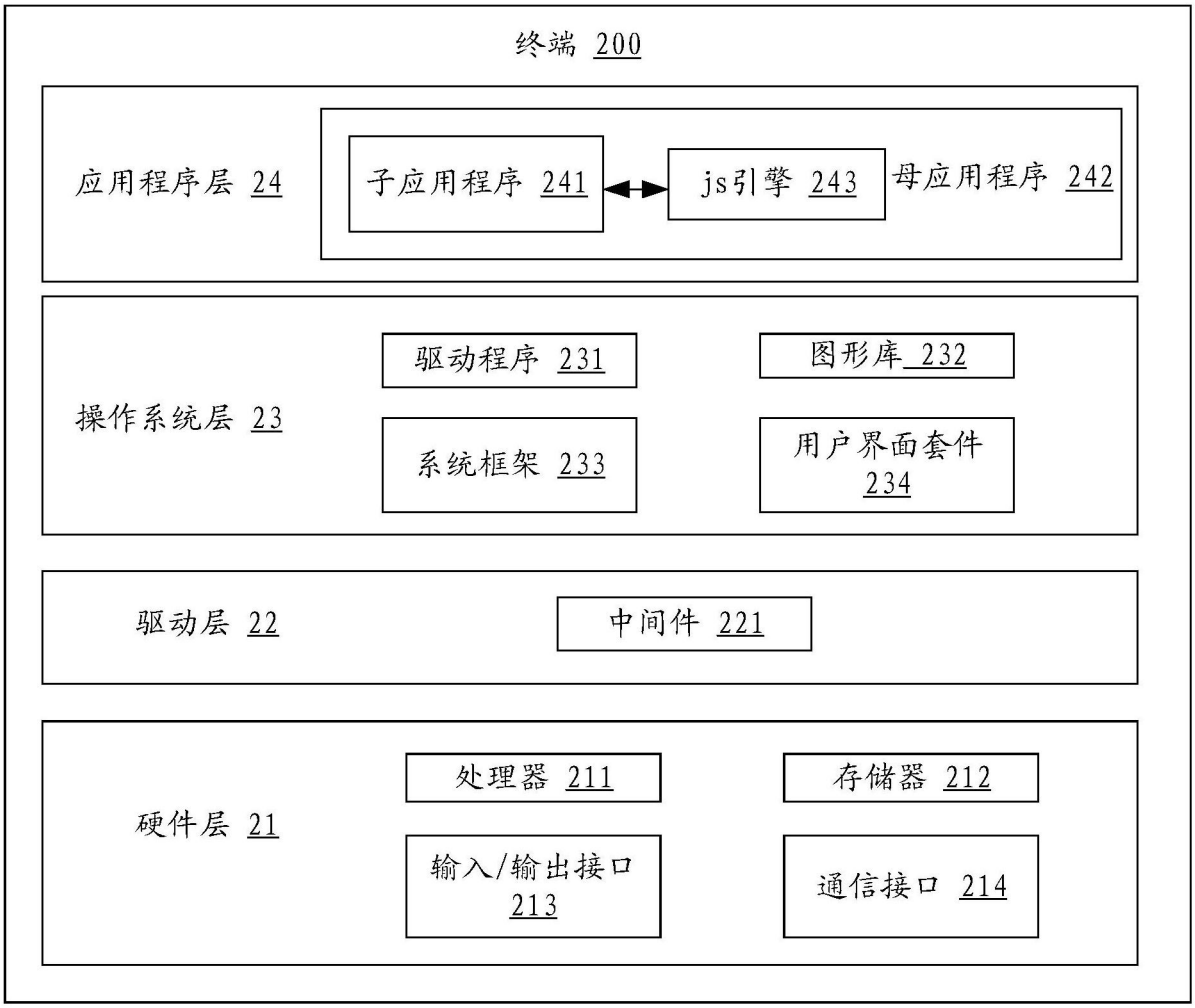

终端 200
应用程序层 24
子应用程序 241
js引擎 243
母应用程序 242
操作系统层 23
驱动程序 231
图形库 232
系统框架 233
用户界面套件 234
驱动层 22
中间件 221
硬件层 21
处理器 211
存储器 212
输入/输出接口 213
通信接口 214

图3

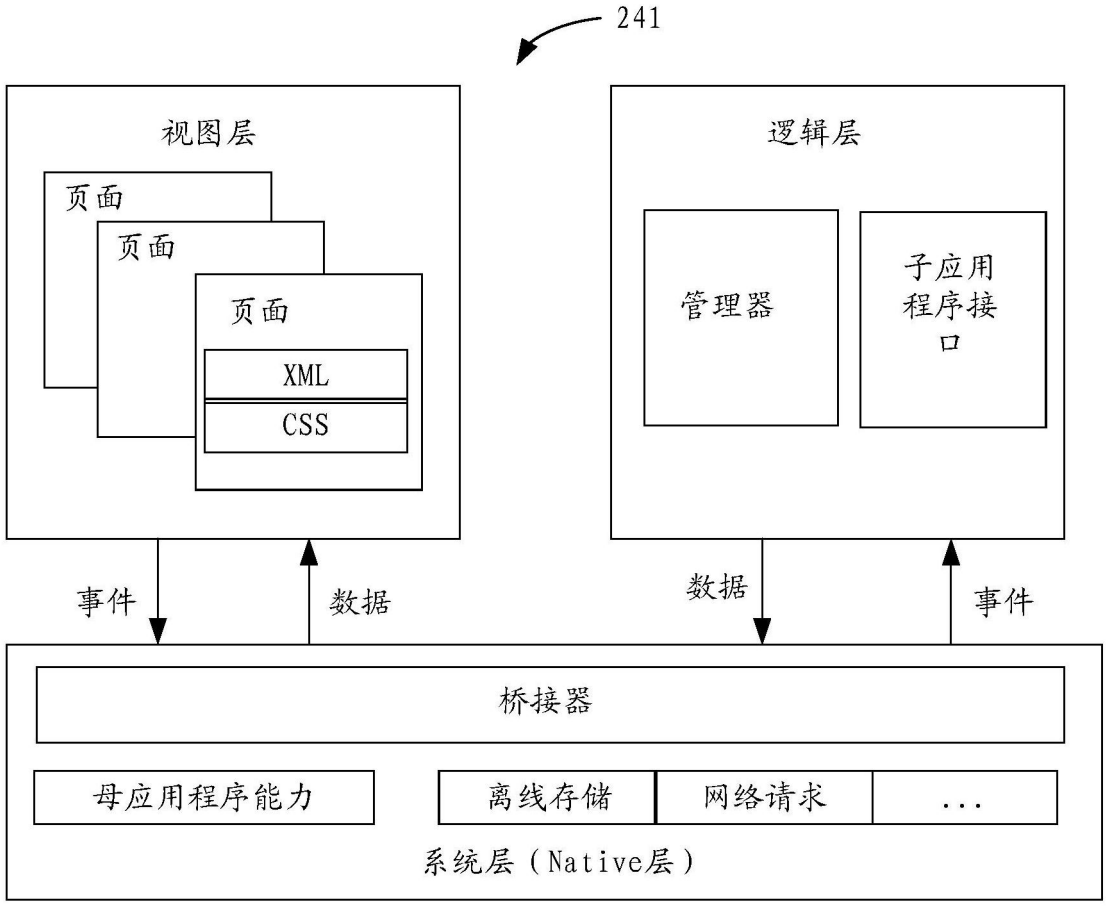

241
视图层
页面
页面
页面
XML
CSS
逻辑层
管理器
子应用程序接口
事件
数据
数据
事件
桥接器
母应用程序能力
离线存储
网络请求
...
系统层（Native层）

图4

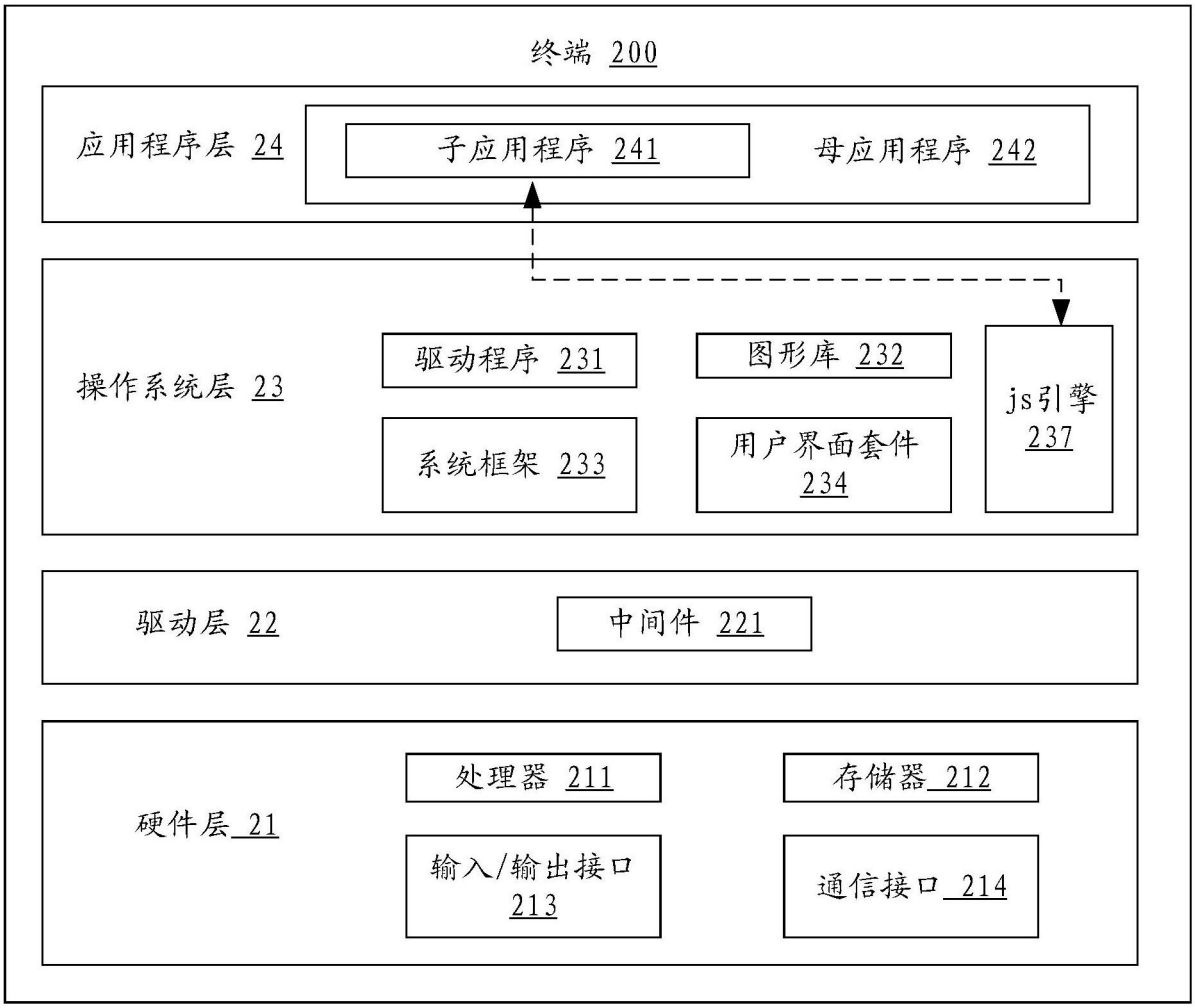

终端 200
应用程序层 24
子应用程序 241
母应用程序 242
操作系统层 23
驱动程序 231
图形库 232
js引擎 237
系统框架 233
用户界面套件 234
驱动层 22
中间件 221
硬件层 21
处理器 211
存储器 212
输入/输出接口 213
通信接口 214

图5

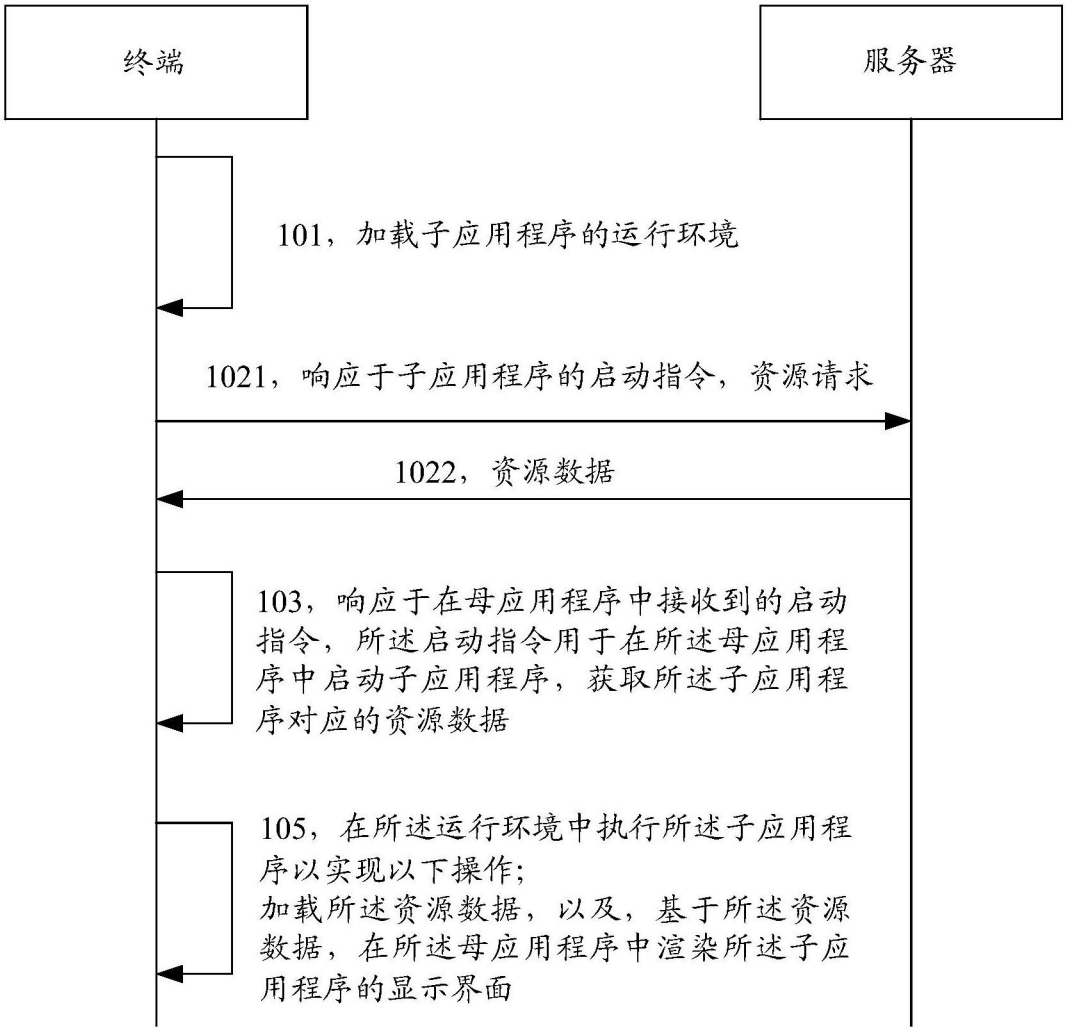

终端
服务器
101，加载子应用程序的运行环境
1021，响应于子应用程序的启动指令，资源请求
1022，资源数据
103，响应于在母应用程序中接收到的启动指令，所述启动指令用于在所述母应用程序中启动子应用程序，获取所述子应用程序对应的资源数据
105，在所述运行环境中执行所述子应用程序以实现以下操作；
加载所述资源数据，以及，基于所述资源数据，在所述母应用程序中渲染所述子应用程序的显示界面

图6

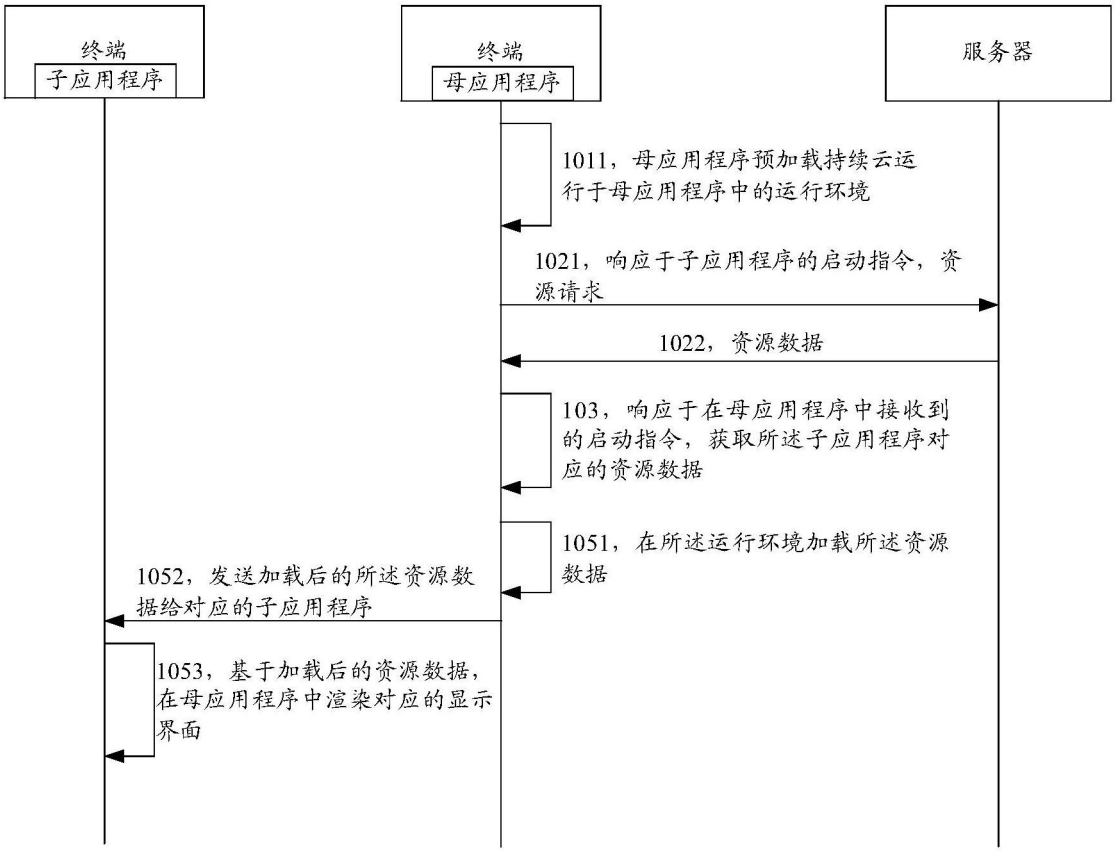

终端
子应用程序
终端
母应用程序
服务器
1011，母应用程序预加载持续云运行于母应用程序中的运行环境
1021，响应于子应用程序的启动指令，资源请求
1022，资源数据
103，响应于在母应用程序中接收到的启动指令，获取所述子应用程序对应的资源数据
1051，在所述运行环境加载所述资源数据
1052，发送加载后的所述资源数据给对应的子应用程序
1053，基于加载后的资源数据，在母应用程序中渲染对应的显示界面

图7

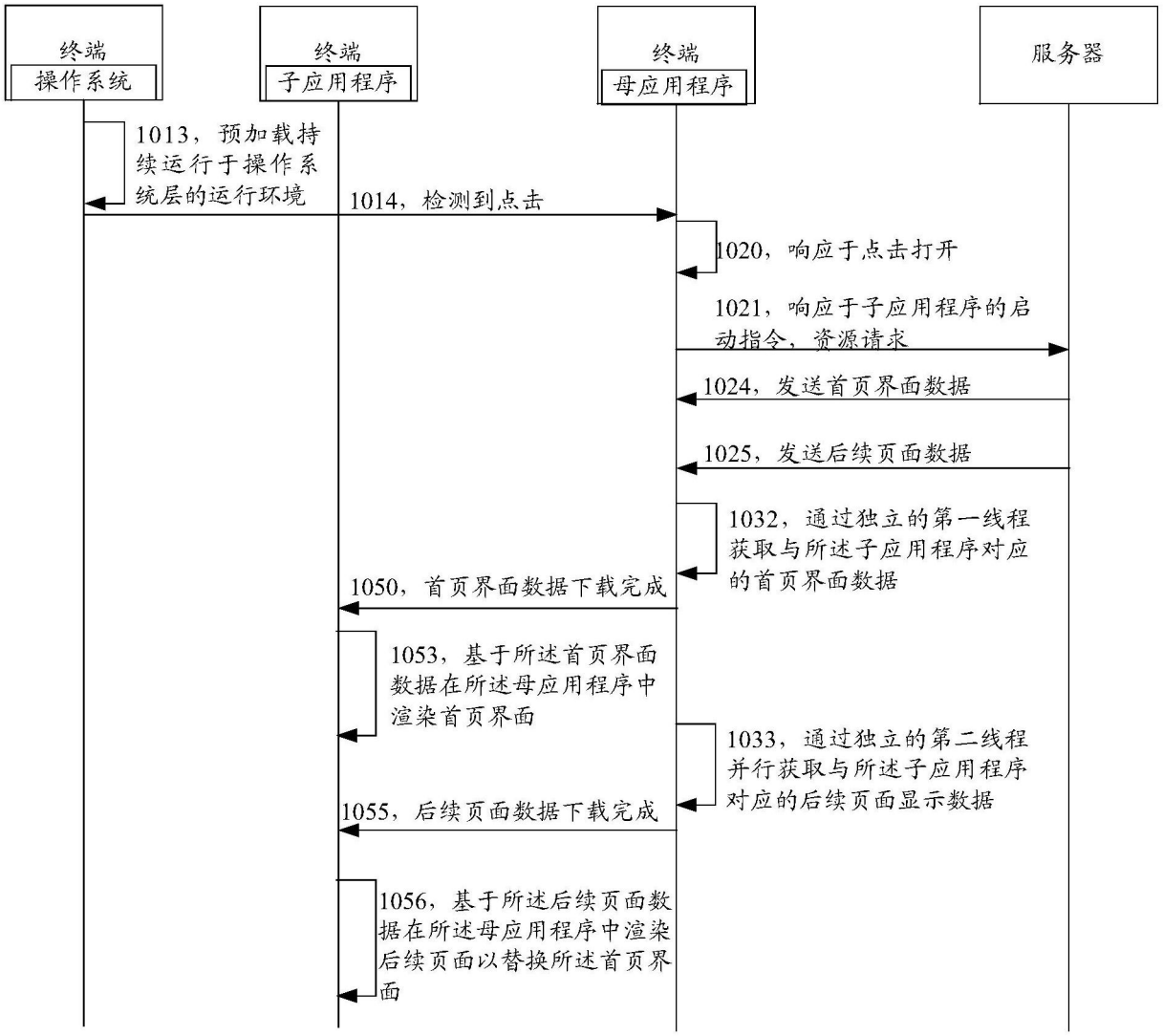

终端
操作系统
终端
子应用程序
终端
母应用程序
服务器
1013，预加载持续运行于操作系统层的运行环境
1014，检测到点击
1020，响应于点击打开
1021，响应于子应用程序的启动指令，资源请求
1024，发送首页界面数据
1025，发送后续页面数据
1032，通过独立的第一线程获取与所述子应用程序对应的首页界面数据
1050，首页界面数据下载完成
1053，基于所述首页界面数据在所述母应用程序中渲染首页界面
1033，通过独立的第二线程并行获取与所述子应用程序对应的后续页面显示数据
1055，后续页面数据下载完成
1056，基于所述后续页面数据在所述母应用程序中渲染后续页面以替换所述首页界面

图8

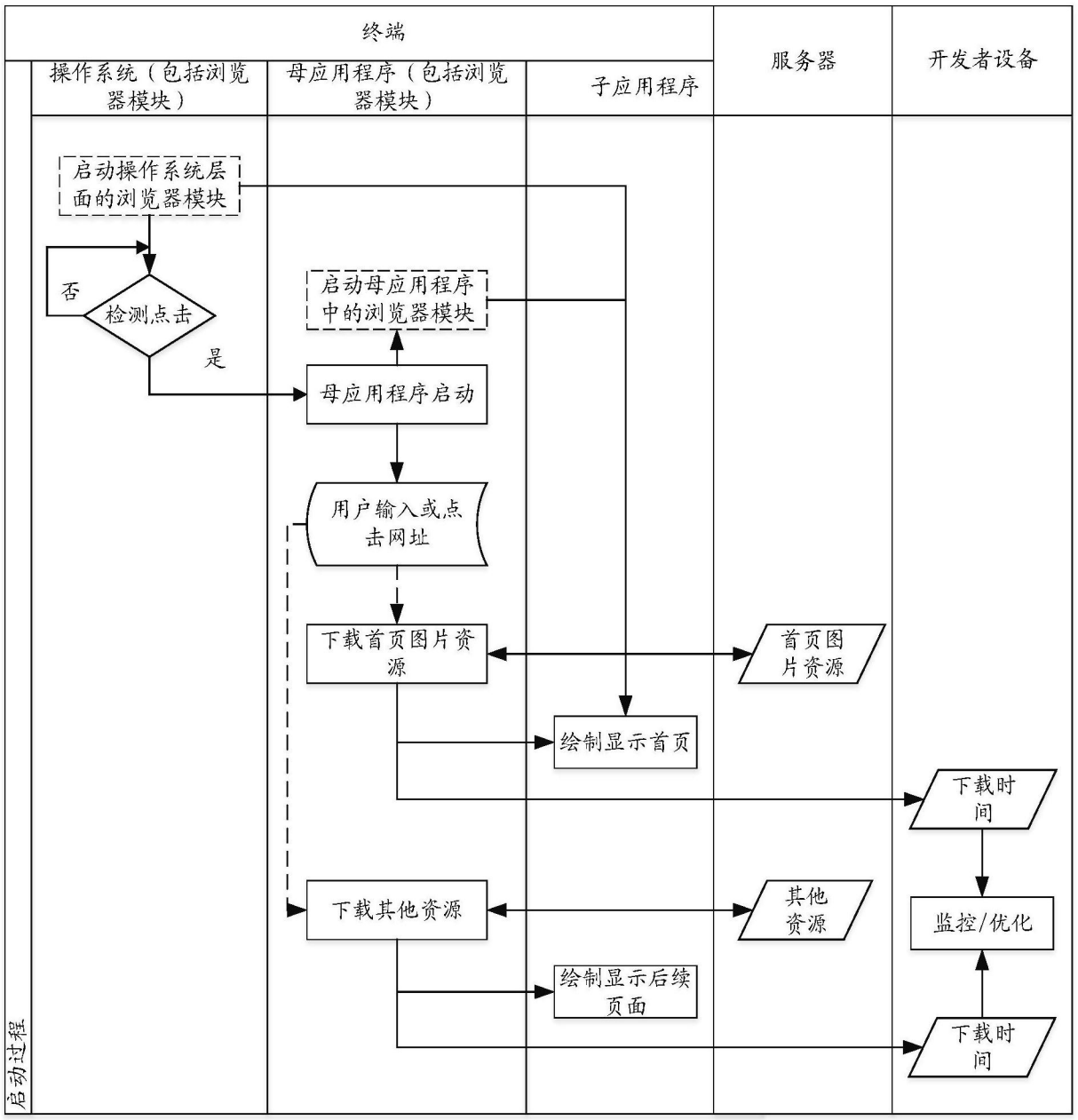

终端
操作系统（包括浏览器模块）
母应用程序（包括浏览器模块）
子应用程序
服务器
开发者设备
启动操作系统层面的浏览器模块
否
检测点击
是
启动母应用程序中的浏览器模块
母应用程序启动
用户输入或点击网址
下载首页图片资源
首页图片资源
绘制显示首页
下载时间
下载其他资源
其他资源
监控/优化
绘制显示后续页面
下载时间
启动过程

图9

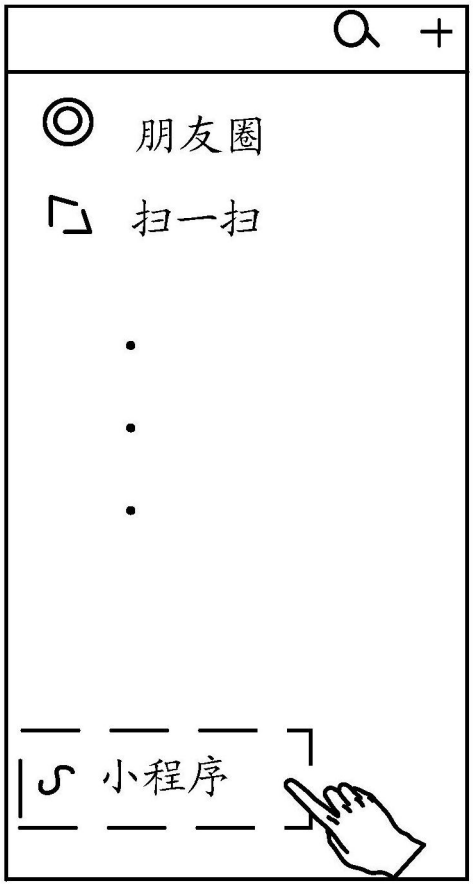


图10A

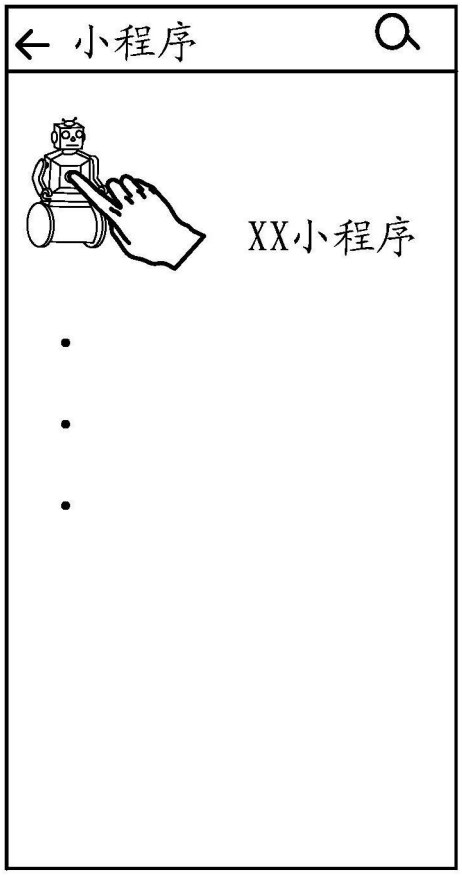


图10B



图10C

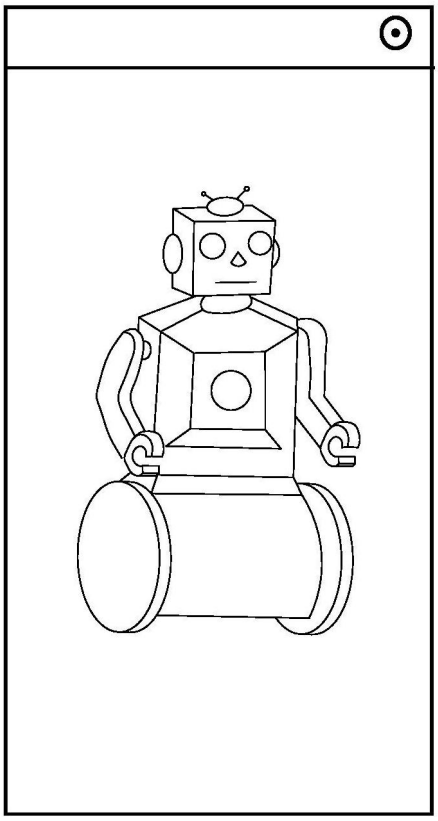


图10D