



(12)发明专利申请

(10)申请公布号 CN 109656611 A

(43)申请公布日 2019. 04. 19

(21)申请号 201811585218.2

(22)申请日 2018.12.24

(71)申请人 成都四方伟业软件股份有限公司
地址 610000 四川省成都市高新区科园三路4号1栋2层

(72)发明人 何晋平 汤琦

(74)专利代理机构 北京超凡志成知识产权代理
事务所(普通合伙) 11371
代理人 邓超

(51) Int. Cl.
G06F 8/656(2018.01)

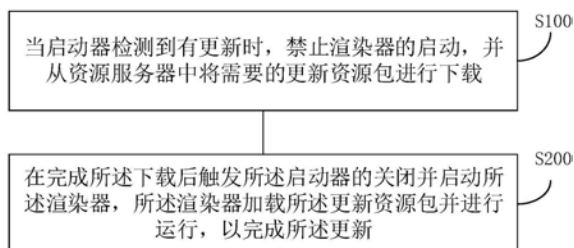
权利要求书1页 说明书6页 附图3页

(54)发明名称

三维可视化应用热更新方法、装置和终端

(57)摘要

本发明公开了一种三维可视化应用热更新方法、装置和终端,该三维可视化应用包括启动器和渲染器,所述方法包括:当所述启动器检测到有更新时,禁止所述渲染器启动,并从资源服务器中将需要的更新资源包进行下载;在完成所述下载后触发所述启动器进行关闭并启动所述渲染器,所述渲染器加载所述更新资源包并进行运行,以完成所述更新。根据本发明的技术方案,可实现对三维可视化应用的快速热更新,并且对于框架代码修改等更新问题,可不用花费较长的时间来重新开发一个完整的应用并让客户重新下载等,更新效率高,还可提高客户体验等。



1. 一种三维可视化应用热更新方法,其特征在于,所述三维可视化应用包括启动器和渲染器,所述方法包括:

当所述启动器检测到有更新时,禁止所述渲染器的启动,并从资源服务器中将需要的更新资源包进行下载;

在完成所述下载后触发所述启动器的关闭并启动所述渲染器,所述渲染器加载所述更新资源包并进行运行,以完成所述更新。

2. 根据权利要求1所述的三维可视化应用热更新方法,其特征在于,还包括:

所述启动器在用户成功登陆后检测是否有更新。

3. 根据权利要求1所述的三维可视化应用热更新方法,其特征在于,还包括:

当所述启动器未检测到有更新时,则启动所述渲染器。

4. 根据权利要求1所述的三维可视化应用热更新方法,其特征在于,所述启动器检测是否有更新的过程包括:

从资源服务器中获取云端存储的版本号;

读取所述三维可视化应用的当前版本号;

比较所述云端存储的版本号和所述当前版本号的大小;

若所述云端存储的版本号高于所述当前版本号,则检测到有更新并发送更新请求到资源服务器。

5. 根据权利要求4所述的三维可视化应用热更新方法,其特征在于,还包括:

所述启动器在接收到来自所述资源服务器发送的含所述更新资源包对应的MD5文件的MD5文件列表后,读取本地的MD5文件列表并进行比对,以确定需要下载的更新资源包。

6. 根据权利要求1所述的三维可视化应用热更新方法,其特征在于,所述资源服务器中的更新资源包预先通过AssetBundle打包处理并进行存储,其中,所述更新资源包包括更新的资源和/或编译代码库。

7. 根据权利要求6所述的三维可视化应用热更新方法,其特征在于,所述更新的资源包包括美术资源、配置文件、新增应用逻辑和修改应用逻辑中的一种或多种。

8. 一种三维可视化应用热更新装置,其特征在于,所述热更新装置包括启动模块和渲染模块,

所述启动模块用于当检测到有更新时,禁止所述渲染器的启动,并从资源服务器中将需要的更新资源包进行下载;

在完成所述下载后触发所述启动模块的关闭并启动所述渲染模块,所述渲染模块用于加载所述更新资源包并进行运行,以完成所述更新。

9. 一种终端,其特征在于,所述终端包括处理器和存储器,所述存储器存储有三维可视化应用,所述三维可视化应用在进行热更新时,所述处理器执行如权利要求1-7任一项所述的三维可视化应用热更新方法。

10. 一种计算机存储介质,其特征在于,其存储有计算机程序,所述计算机程序在被执行时实施如权利要求1-7任一项所述的三维可视化应用热更新方法。

三维可视化应用热更新方法、装置和终端

技术领域

[0001] 本发明涉及三维可视化技术领域,尤其涉及一种三维可视化应用热更新方法、装置和终端。

背景技术

[0002] 当前,人类世界已经迈入大数据(Big Data)时代,随着互联网、物联网、云计算等信息技术的迅猛发展,信息技术与人类世界的经济、军事、科研、生活等方面不断交叉融合,数据变得无处不在、触手可及。数据本身具备的不可阅读性和抽象性,给人们的数据应用造成了巨大困难,于是三维可视化技术应运而生。

[0003] 对于三维可视化应用而言,为保障客户对应用的较好体验,以及为缩短用户取得新版客户端的相关流程,各种热更新的方案相继被提出。其中,热更新可实现动态下发代码,它可以使开发者在不发布新版本的情况下进行修复BUG或发布新功能,而对于用户而言,也可不用长时间等待下一个新版本等。

[0004] 然而,目前的三维可视化应用的热更新方案中,存在开发难度大、效率低下,更新速度慢,且当框架代码有所改动或需要修改时,则必须由开发方重新花费较长的时间开发出一个完整应用软件,而不能在原有应用的基础上直接进行热更新等,故往往导致客户的体验不好等。

发明内容

[0005] 鉴于上述问题,本发明提出一种三维可视化应用热更新方法、装置和终端,通过启动器与渲染器的依次启动切换等以实现热更新,可以解决现有技术中的更新效率慢、无法直接进行框架代码更新等问题。

[0006] 本发明实施例提出一种三维可视化应用热更新方法,所述三维可视化应用包括启动器和渲染器,所述热更新方法包括:

[0007] 当所述启动器检测到有更新时,禁止所述渲染器的启动,并从资源服务器中将需要的更新资源包进行下载;

[0008] 在完成所述下载后触发所述启动器的关闭并启动所述渲染器,所述渲染器加载所述更新资源包并进行运行,以完成所述更新。

[0009] 在上述的三维可视化应用热更新方法中,可选地,所述启动器在用户成功登陆后检测是否有更新。

[0010] 在上述的三维可视化应用热更新方法中,可选地,当所述启动器未检测到有更新时,则启动所述渲染器。

[0011] 在上述的三维可视化应用热更新方法中,可选地,所述启动器检测是否有更新的过程包括:

[0012] 所述启动器检测是否有更新的过程包括:

[0013] 从资源服务器中获取云端存储的版本号;

- [0014] 读取所述三维可视化应用的当前版本号；
- [0015] 比较所述云端存储的版本号和所述当前版本号的大小；
- [0016] 若所述云端存储的版本号高于所述当前版本号，则检测到有更新并发送更新请求到资源服务器。
- [0017] 在上述的三维可视化应用热更新方法中，可选地，所述启动器在接收到来自所述资源服务器发送的含所述更新资源包对应的MD5文件的MD5文件列表后，读取本地的MD5文件列表并进行比对，以确定需要下载的更新资源包。
- [0018] 在上述的三维可视化应用热更新方法中，可选地，所述资源服务器中的更新资源包预先通过AssetBundle打包处理并进行存储，其中，所述更新资源包包括更新的资源和/或编译代码库。
- [0019] 在上述的三维可视化应用热更新方法中，可选地，所述更新的资源包括美术资源、配置文件、新增应用逻辑和修改应用逻辑中的一种或多种。
- [0020] 本发明的另一实施例还提出一种三维可视化应用热更新装置，所述热更新装置包括启动模块和渲染模块，
- [0021] 所述启动模块用于当检测到有更新时，禁止所述渲染器的启动，并从资源服务器中将需要的更新资源包进行下载；
- [0022] 在完成所述下载后触发所述启动模块的关闭并启动所述渲染模块，所述渲染模块用于加载所述更新资源包并进行运行，以完成所述更新。
- [0023] 本发明的又一实施例还提出一种终端，所述终端包括处理器和存储器，所述存储器存储有三维可视化应用，所述三维可视化应用在进行热更新时，所述处理器执行上述的三维可视化应用热更新方法。
- [0024] 本发明的再一实施例还提出一种计算机存储介质，其存储有计算机程序，所述计算机程序在被执行时实施上述的三维可视化应用热更新方法。
- [0025] 通过本发明的技术方案可以实现对三维可视化应用的快速热更新，并且对于框架代码修改等更新问题，可不用花费较长的时间来重新开发一个完整的应用并让客户重新下载等，更新效率高，还可提高客户体验等。

附图说明

- [0026] 为了更清楚地说明本发明的技术方案，下面将对实施例中所需要使用的附图作简单地介绍，应当理解，以下附图仅示出了本发明的某些实施例，因此不应被看作是对本发明保护范围的限定。
- [0027] 图1为本发明实施例的三维可视化应用热更新的应用示意图；
- [0028] 图2为本发明实施例的三维可视化应用热更新方法的第一流程示意图；
- [0029] 图3为本发明实施例的三维可视化应用热更新方法的检测是否有更新的流程示意图；
- [0030] 图4为本发明实施例的三维可视化应用热更新方法的第二流程示意图；
- [0031] 图5为本发明实施例的三维可视化应用热更新装置的结构示意图。
- [0032] 主要元件符号说明：
- [0033] 100-三维可视化应用热更新装置；10-启动模块；20-渲染模块。

具体实施方式

[0034] 下面将结合本发明实施例中附图,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实施例,而不是全部的实施例。通常在此处附图中描述和示出的本发明实施例的组件可以以各种不同的配置来布置和设计。因此,以下对在附图中提供的本发明的实施例的详细描述并非旨在限制要求保护的本发明的范围,而是仅仅表示本发明的选定实施例。基于本发明的实施例,本领域技术人员在没有做出创造性劳动的前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0035] 此外,术语“第一”、“第二”仅用于描述目的,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量。由此,限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。在本发明的描述中,“多个”的含义是两个或两个以上,除非另有明确具体的限定。

[0036] 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。

[0037] 下面结合具体的实施例对本发明进行详细说明。

[0038] 实施例1

[0039] 请参照图1和图2,本实施例提出一种三维可视化应用热更新方法,可应用于三维可视化软件的热更新,如3D办公用软件、3D游戏等。如图1所示,若资源服务器存储有版本更新资源,安装有该三维可视化应用的各用户终端可通过从该资源服务器中下载需要使用的更新资源包并进行热更新。

[0040] 本实施例中,该三维可视化应用包括启动器和渲染器,利用这关联的两个exe程序可以实现该三维可视化应用的快速热更新。如图2所示,下面对该热更新方法的步骤进行详细说明。

[0041] 步骤S100,当启动器检测到有更新时,禁止渲染器的启动,并从资源服务器中将需要的更新资源包进行下载。

[0042] 本实施例中,所述启动器启动后将先检测是否有版本更新,若检测到有更新时,将禁止所述渲染器的启动并从资源服务器中下载相应的更新资源包。示范性地,所述启动器将下载的更新资源包进行解压等操作后,将其存储到对应的本地配置文件中,以新增或更新相关资源文件。

[0043] 其中,如图3所示,所述启动器检测是否有更新的过程主要包括以下几个步骤。

[0044] 步骤S110,从资源服务器中获取云端存储的版本号。

[0045] 步骤S120,读取所述三维可视化应用的当前版本号。

[0046] 本实施例中,所述启动器包括更新模块,该更新模块可用于检测版本的更新情况以及从资源服务器进行资源的下载等。在上述步骤S110和步骤S120中,所述启动器将通过该更新模块发送获取版本号信息请求到资源服务器,以获取云端存储的版本号。

[0047] 于是,当所述启动器获取到云端存储的版本号时,再与从本地资源中读取的该三维可视化应用的当前版本号进行比较,即执行步骤S130,比较所述云端存储的版本号和所述当前版本号的大小。

[0048] 在上述步骤S130中,若所述云端存储的版本号小于或等于所述当前版本号,则说明该三维可视化应用的版本为最新版本,故判断为不需要更新。反之,若所述云端存储的版本号高于所述当前版本号,则判断为需要更新,即检测到有更新,于是执行步骤S140。

[0049] 步骤S140,若所述云端存储的版本号高于所述当前版本号,则检测到有更新并发送更新请求到资源服务器。

[0050] 于是,所述启动器将发送更新请求到资源服务器,并在接收到由该资源服务器返回其存储的MD5文件列表。随后,所述启动器将读取本地的MD5文件列表,并与该接收的MD5文件列表进行比对,通过找到MD5改变的文件以确定需要下载的更新资源包。

[0051] 其中,该资源服务器存储的MD5文件包括每个被上传打包的资源文件的MD5。所述MD5(Message Digest Algorithm 5,消息摘要算法)可用于提供消息的完整性保护。当更新的资源被打包提交到资源服务器时,同时会将打包的资源文件的MD5保存到一个文件以形成该MD5文件列表。故通过该MD5可以确定其对应的资源文件是否有更改,以及是否有新增的资源文件等。

[0052] 本实施例中,该资源服务器中存储的更新资源包可预先通过AssetBundle打包处理并进行存储。其中,AssetBundle打包是一种可将多个文件压缩成AssetBundle包的方法,AssetBundle包可使用压缩算法以减少包的大小,从而可更快地进行网络传输等。利用AssetBundle分组策略,可实现对需要更新或加载的资源进行灵活打包,例如,可将经常更新的资源放在一个单独的包,以跟不经常更新的包分离;或者,可将需要共享的资源放在一个单独的包里,还可将一些需要同时加载的资源放在一个包里等等。

[0053] 示范性地,所述更新资源包包括更新的资源和/或编译代码库。其中,该编译代码库中的代码通常采用如C#等编译语言编写,可用于实现该三维可视化应用的核心逻辑以及框架代码等。而所述渲染器运行前若对这些编译代码库等进行重新加载并进行编译运行,则可将一些新增或修改的核心逻辑或框架代码等进行更新。

[0054] 而现有技术中,由于三维可视化应用仅包括一个exe程序,故当该应用启动时,其渲染模块也将在下载更新资源前被同时启动,故只能对一些非核心逻辑或框架代码等资源进行热更新,而无法进行修改框架代码等更新。

[0055] 示范性地,所述更新的资源可包括但不限于为美术资源、配置文件、新增的或修改的应用逻辑等等。例如,该美术资源可包括一些修改或新增的场景资源;新增应用逻辑可包括一些新的使用、玩法、活动等等。

[0056] 可以理解,本实施例中的启动器将先启动并检测是否有版本更新,并且在检测到有更新时下载更新资源包。而且,所述启动器在下载更新资源包的这段时间内,所述渲染器仍处于关闭(即不启动)状态。当所述启动器完成该更新资源包的下载时,才触发所述渲染器的启动,同时关闭所述启动器,即执行步骤S200。

[0057] 步骤S200,在完成所述下载后触发所述启动器的关闭并启动所述渲染器,所述渲染器加载所述更新资源包并进行运行,以完成所述更新。

[0058] 于是,当所述更新资源包下载完成后,将触发所述渲染器的启动,同时也将触发所述启动器的关闭,以实现所述启动器与所述渲染器的切换运行。可以理解,所述渲染器在加载该更新资源包时,同时还将加载其他需要的资源等,以用于支持所述三维可视化应用的正常运行。随后,该渲染器将显示相关界面,以供用户进行其他操作等。到此,即完成本次的

热更新。

[0059] 进一步可选地,如图4所示,该三维可视化应用热更新方法包括步骤S90,启动器在用户成功登陆后检测是否有更新。

[0060] 示范性地,当用户进行应用登陆时,若检测到用户输入的账户及密码均匹配时,则允许用户登陆,并在用户登陆后开始检测该三维可视化应用是否有版本更新,即执行上述步骤S100。可以理解,通过增加用户登陆校验这一操作,可保证用户对该三维可视化应用的安全使用,进而也可防止他人恶意篡改或盗用该三维可视化应用等。

[0061] 进一步可选地,该三维可视化应用热更新方法包括步骤S300,当所述启动器未检测到有更新时,则启动所述渲染器。

[0062] 示范性地,当所述启动器未检测到更新时,则启动所述渲染器,并触发所述启动器进行关闭。所述渲染器启动后将显示相关界面,以供用户进行后续操作等。

[0063] 本实施例提出的三维可视化应用热更新方法通过将三维可视化应用的启动器先启动并检测是否有更新及下载更新资源等,再触发渲染器启动,使得渲染器可在启动时加载所有需要的资源后再运行,从而可实现热更新。利用该方法还可以实现对框架代码等的热更新,而利用启动器与渲染器的依次启动及切换运行,但客户却完全感知不到,故还可提高客户体验等。

[0064] 实施例2

[0065] 请参照图5,基于上述实施例1的三维可视化应用热更新方法,本实施例中提供一种三维可视化应用热更新装置100,该三维可视化应用热更新装置100包括启动模块10和渲染模块20。

[0066] 所述启动模块10用于当检测到有更新时,禁止所述渲染器的启动,并从资源服务器中将需要的更新资源包进行下载;

[0067] 在完成所述下载后触发所述启动模块10的关闭并启动所述渲染模块20,所述渲染模块20用于加载所述更新资源包并进行运行,以完成所述更新。

[0068] 上述的三维可视化应用热更新装置100对应于实施例1的三维可视化应用热更新方法。实施例1中的任何可选项也适用于本实施例,这里不再详述。

[0069] 本发明还提供了一种移动终端,该移动终端可以包括智能电话、平板电脑等。该移动终端包括存储器和处理器,存储器可用于存储计算机程序,处理器通过运行所述计算机程序,从而使移动终端执行上述三维可视化应用热更新方法或者上述三维可视化应用热更新装置中的各个模块的功能。

[0070] 存储器可包括存储程序区和存储数据区,其中,存储程序区可存储操作系统、至少一个功能所需的应用程序;存储数据区可存储根据移动终端的使用所创建的数据(比如音频数据、电话本等)等。此外,存储器可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件、闪存器件、或其他易失性固态存储器件。

[0071] 本发明还提供了一种计算机存储介质,用于储存上述移动终端中使用的所述计算机程序。

[0072] 在本申请所提供的几个实施例中,应该理解到,所揭露的装置和方法,也可以通过其它的方式实现。以上所描述的装置实施例仅仅是示意性的,例如,附图中的流程图和结构图显示了根据本发明的多个实施例的装置、方法和计算机程序产品的可能实现的体系架

构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分,所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。

[0073] 也应当注意,在作为替换的实现方式中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,结构图和/或流程图中的每个方框、以及结构图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0074] 另外,在本发明各个实施例中的各功能模块或单元可以集成在一起形成一个独立的部分,也可以是各个模块单独存在,也可以两个或更多个模块集成形成一个独立的部分。

[0075] 所述功能如果以软件功能模块的形式实现并作为独立的产品销售或使用,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(可以是智能手机、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、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0076] 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。

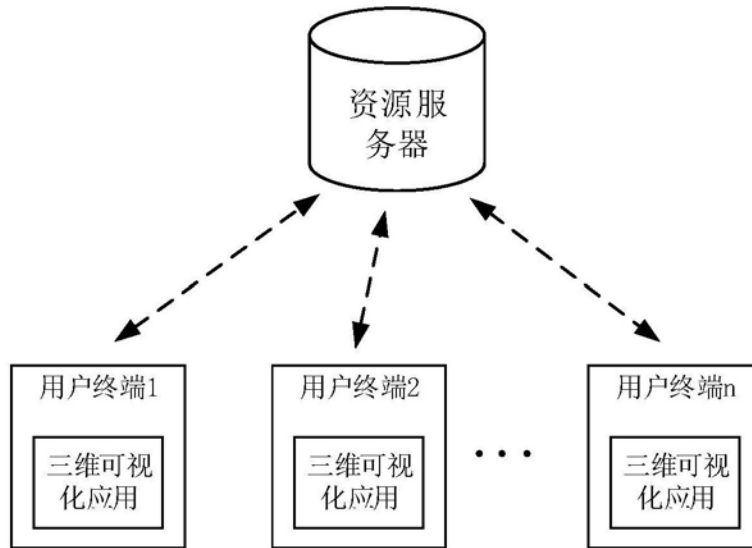


图1

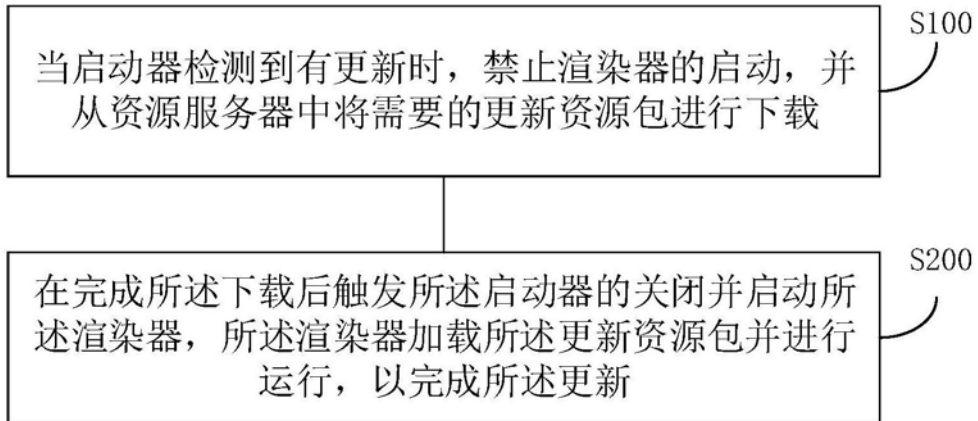


图2

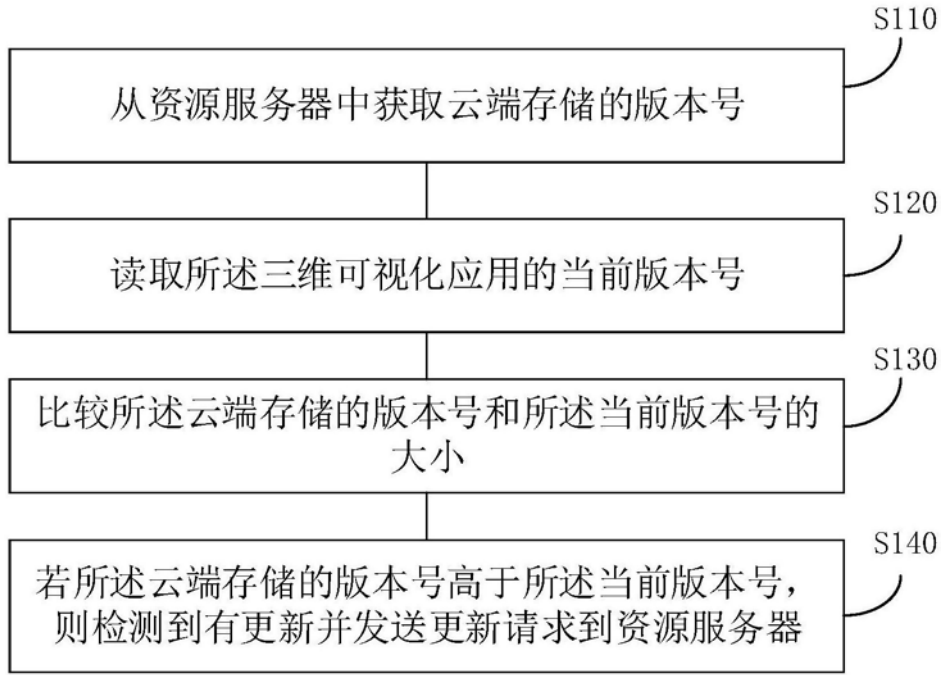


图3

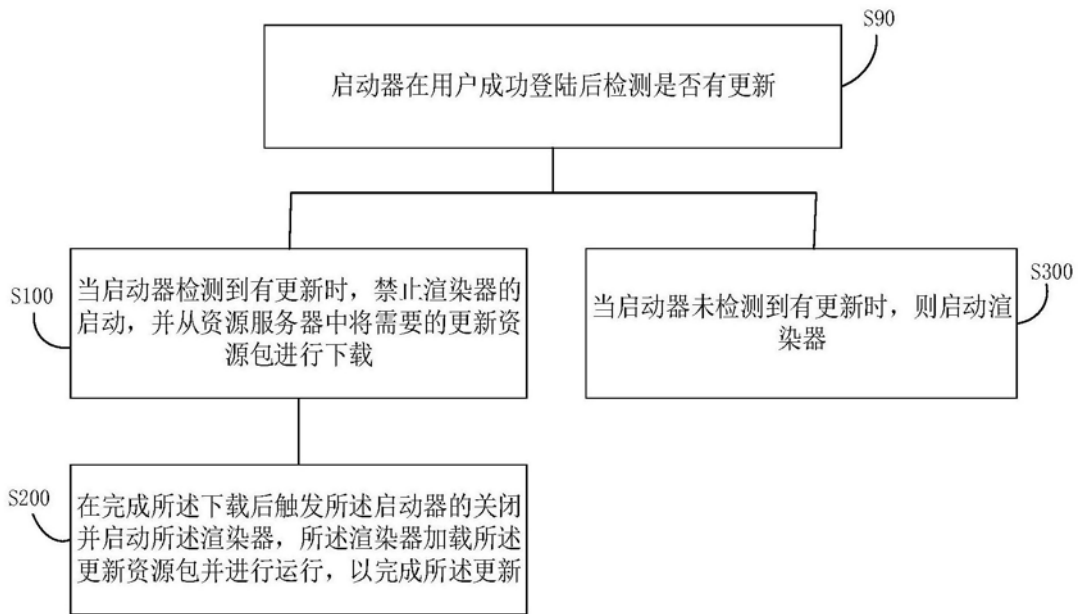


图4

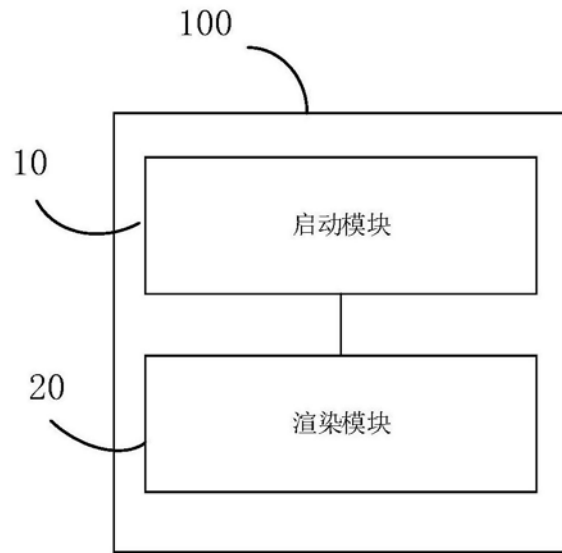


图5