



(12) 发明专利申请

(10) 申请公布号 CN 114077458 A

(43) 申请公布日 2022. 02. 22

(21) 申请号 202111238225.7

(22) 申请日 2021.10.25

(71) 申请人 上海浦东发展银行股份有限公司
地址 200001 上海市黄浦区中山东一路12号

(72) 发明人 王超 张俊 沈含雪 冯超
祝宏杰

(74) 专利代理机构 广州华进联合专利商标代理有限公司 44224

代理人 唐敏

(51) Int. Cl.

G06F 9/445 (2018.01)

G06F 8/65 (2018.01)

G06F 8/71 (2018.01)

权利要求书2页 说明书11页 附图5页

(54) 发明名称

应用程序内业务模块启动方法、装置、设备和存储介质

(57) 摘要

本申请涉及一种应用程序内业务模块启动方法、装置、计算机设备和存储介质。所述方法包括：在应用程序启动时，向服务端请求响应信息；接收服务端下发的响应信息，基于响应信息获取服务端与应用程序对应的拆分包的版本号，以及服务端与应用程序对应的基础库数据包的最新版本号；基于最新版本号，将本地与应用程序相对应的基础库数据包更新至最新版本，并加载到内存；基于服务端的拆分包的版本号，确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号；确定最新拆分包版本号与基础库数据包的最新版本号匹配，并且基础库数据包已加载完成，加载最新拆分包版本号对应的最新拆分包至内存，启动业务模块。采用本方法能够提高应用程序中业务模块页面打开的速度。



1. 一种应用程序内业务模块启动方法,其特征在于,所述方法包括:
在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;
接收服务端下发的响应信息,基于所述响应信息获取服务端与所述应用程序对应的拆分包的版本号,以及服务端与所述应用程序对应的基础库数据包的最新版本号;
基于所述最新版本号,将本地与所述应用程序相对应的基础库数据包更新至最新版本,并加载到内存;
基于所述服务端的所述拆分包的版本号,确定本地与所述业务模块对应的拆分包的版本已更新至最新拆分包版本号;
确定所述最新拆分包版本号与所述基础库数据包的最新版本号匹配,并且所述基础库数据包已加载完成,加载所述最新拆分包版本号对应的最新拆分包至内存;
基于加载的所述最新拆分包及所述基础库数据包,启动所述业务模块。
2. 根据权利要求1所述的方法,其特征在于,所述向服务端请求与拆分包版本相关的响应信息,包括:
获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求;
向服务端发送所述请求,以请求与拆分包版本相关的响应信息。
3. 根据权利要求1所述的方法,其特征在于,所述基于所述服务端的所述拆分包的版本号,确定本地与所述业务模块对应的拆分包的版本已更新至最新拆分包版本号,包括:
基于所述响应信息,获取所述服务端与所述业务模块对应的拆分包的版本号;
当所述服务端与所述业务模块对应拆分包的版本号与本地与所述业务模块对应拆分包的版本号一致时,确定本地与所述业务模块对应的拆分包的版本已更新至最新拆分包版本号。
4. 根据权利要求3所述的方法,其特征在于,所述基于所述服务端的所述拆分包的版本号,确定本地与所述业务模块对应的拆分包的版本已更新至最新拆分包版本号,还包括:
当多个所述服务端与所述业务模块对应拆分包的版本号、所述本地与所述业务模块对应拆分包的版本号不一致时,获取所述响应信息中的拆分包压缩文件下载地址;
基于所述拆分包压缩文件下载地址,下载并解压所述与所述业务模块对应的拆分包至本地,使本地与所述业务模块对应的拆分包的版本更新至最新拆分包版本号。
5. 根据权利要求1所述的方法,其特征在于,所述获取与所述业务模块对应的拆分包的最新拆分包版本号后,还包括:
当所述最新拆分包版本号与所述服务端基础库数据包的最新版本号不匹配时,获取与所述最新拆分包版本号对应的业务包,由所述业务包启动所述业务模块。
6. 根据权利要求5所述的方法,其特征在于,所述获取与所述最新拆分包版本号对应的业务包,由所述业务包启动所述业务模块,包括:
当本地存在所述业务包时,加载所述业务包至内存,启动所述业务模块;
当本地不存在所述业务包时,向服务端请求下载所述业务包至本地,再加载所述业务包至内存,启动所述业务模块。
7. 根据权利要求1所述的方法,其特征在于,所述确定所述最新拆分包版本号与所述基础库数据包的最新版本号匹配后,还包括:
若所述最新版本号对应的基础库数据包未加载完成,停止所述最新版本号对应的基础

库数据包的加载,合并加载所述最新版本号对应的基础库数据包、所述本地与所述业务模块对应拆分包到内存。

8.一种应用程序内业务模块启动装置,其特征在于,所述装置包括:

响应信息请求模块,用于在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;

响应信息接收模块,用于接收服务端下发的响应信息,基于所述响应信息获取服务端与所述应用程序对应的拆分包的版本号,以及服务端与所述应用程序对应的基础库数据包的最新版本号;

预加载模块,用于基于所述最新版本号,将本地与所述应用程序相对应的基础库数据包更新至最新版本,并加载到内存;

拆分包更新模块,用于基于所述服务端的所述拆分包的版本号,确定本地与所述业务模块对应的拆分包的版本已更新至最新拆分包版本号;

拆分包加载模块,用于确定所述最新拆分包版本号与所述基础库数据包的最新版本号匹配,并且所述基础库数据包已加载完成,加载所述最新拆分包版本号对应的最新拆分包至内存;

业务模块启动模块,用于基于最新加载的所述最新拆分包及所述基础库数据包,启动所述业务模块。

9.一种计算机设备,包括存储器和处理器,所述存储器存储有计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至7中任一项所述的方法的步骤。

10.一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,所述计算机程序被处理器执行时实现权利要求1至7中任一项所述的方法的步骤。

应用程序内业务模块启动方法、装置、设备和存储介质

技术领域

[0001] 本申请涉及计算机技术领域,特别是涉及一种应用程序内业务模块启动方法、装置、设备和存储介质。

背景技术

[0002] 随着应用程序(Application,APP)热更新的广泛应用,应用程序摆脱了发版的限制,开发者可以在不发布新版本的情况下,修复BUG和发布功能。

[0003] 在现有技术中,在热更新过程中,服务器将基础库数据包与拆分包作为两个独立的文件单独下发给客户端来进行业务模块的热更新。

[0004] 但在实际应用中,基础类库有任何修改都需要重新集成基础库数据包至原生APP中,且所有对应拆分包都须同步更新变动,由于服务器无法保证将基础库数据包与拆分包同时下发给客户端,因此客户端接收到的基础库数据包与拆分包的版本可能出现不一致,则会导致热更新失败,使得客户端需要重新获取新的基础库数据包与拆分包,导致APP页面打开速度缓慢。因此,如何实现业务模块更新支持拆分包、基础库数据包,提升应用程序中业务模块页面打开速度是现阶段亟待解决的问题。

发明内容

[0005] 基于此,有必要针对上述技术问题,提供一种能够实现业务模块更新支持拆分包、基础库数据包,提升应用程序中业务模块页面打开速度的应用程序内业务模块启动方法、装置、设备和存储介质。

[0006] 一种应用程序内业务模块启动方法,包括:

[0007] 在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;

[0008] 接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号;

[0009] 基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存;

[0010] 基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号;

[0011] 确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存;

[0012] 基于加载的最新拆分包及基础库数据包,启动业务模块。

[0013] 在其中一个实施例中,向服务端请求与拆分包版本相关的响应信息,包括:

[0014] 获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求;

[0015] 向服务端发送请求,以请求与拆分包版本相关的响应信息。

[0016] 在其中一个实施例中,基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,包括:

- [0017] 基于响应信息,获取服务端与业务模块对应的拆分包的版本号;
- [0018] 当服务端与业务模块对应拆分包的版本号与本地与业务模块对应拆分包的版本号一致时,获取本地与业务模块对应的拆分包中版本已更新至最新拆分包版本号。
- [0019] 在其中一个实施例中,基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,还包括:
- [0020] 当多个服务端与业务模块对应拆分包的版本号、本地与业务模块对应拆分包的版本号不一致时,获取响应信息中的拆分包压缩文件下载地址;
- [0021] 基于拆分包压缩文件下载地址,下载并解压与业务模块对应的拆分包至本地,使本地与业务模块对应的拆分包的版本更新至最新拆分包版本号。
- [0022] 在其中一个实施例中,获取与业务模块对应的拆分包的最新拆分包版本号后,还包括:
- [0023] 当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块。
- [0024] 在其中一个实施例中,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块,包括:
- [0025] 当本地存在业务包时,加载业务包至内存,启动业务模块;
- [0026] 当本地不存在业务包时,向服务端请求下载业务包至本地,再加载业务包至内存,启动业务模块。
- [0027] 在其中一个实施例中,确定最新拆分包版本号与基础库数据包的最新版本号匹配后,还包括:
- [0028] 若最新版本号对应的基础库数据包未加载完成,停止最新版本号对应的基础库数据包的加载,合并加载最新版本号对应的基础库数据包、本地与业务模块对应拆分包到内存。
- [0029] 一种应用程序内业务模块启动装置,包括:
- [0030] 响应信息请求模块,用于在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;
- [0031] 响应信息接收模块,用于接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号;
- [0032] 预加载模块,用于基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存;
- [0033] 拆分包更新模块,用于基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号;
- [0034] 拆分包加载模块,用于确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存;
- [0035] 业务模块启动模块,用于基于加载的最新拆分包及基础库数据包,启动业务模块。
- [0036] 一种计算机设备,包括存储器和处理器,存储器存储有计算机程序,处理器执行计算机程序时实现上述任一实施例所述的方法的步骤。
- [0037] 一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时

实现上述任一实施例所述的方法的步骤。

[0038] 上述应用程序内业务模块更新方法、装置、设备和存储介质,首先在应用程序启动时,向服务端请求与拆分包版本相关的响应信息,从而获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号,用于后续的版本匹配。进一步,基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,从而实现将本地基础库数据包更新至最新版本,并加载最新版本号对应的基础库数据包到内存。进一步,基于服务端的拆分包的版本号,将本地与业务模块对应的拆分包版本已更新至最新拆分包版本版本号。进一步,确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,单独加载最新拆分包至内存,基于加载的最新拆分包及基础库数据包,启动业务模块,在服务端与业务模块对应的拆分包、服务端与业务模块对应的基础库数据包版本一致的情况下可以提高打开应用程序业务模块的速度。

附图说明

- [0039] 图1为一个实施例中应用程序内业务模块启动方法的应用场景图;
- [0040] 图2为一个实施例中应用程序内业务模块启动方法的流程示意图;
- [0041] 图3为一个实施例中基础库数据包更新预加载的步骤流程示意图;
- [0042] 图4为另一个实施例中拆分包更新加载的步骤流程示意图;
- [0043] 图5为一个实施例中应用程序内业务模块启动装置的结构框图;
- [0044] 图6为一个实施例中计算机设备的内部结构图。

具体实施方式

[0045] 为了使本申请的目的、技术方案及优点更加清楚明白,以下结合附图及实施例,对本申请进行进一步详细说明。应当理解,此处描述的具体实施例仅仅用以解释本申请,并不用于限定本申请。

[0046] 本申请提供的应用程序内业务模块启动方法,可以应用于如图1所示的应用环境中。其中,终端102通过网络与服务器104进行通信。在应用程序启动时,终端102向服务器104请求与拆分包版本相关的响应信息,服务器104接收请求后下发相应的响应信息。进一步,终端102接收服务器104下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包(Common包)的最新版本号,用于后续的版本匹配。进一步,当本地与应用程序对应的基础库数据包的版本号、服务端与应用程序对应的基础库数据包的最新版本号不匹配时,终端102下载最新版本号对应的基础库数据包,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载最新版本号对应的基础库数据包到内存。进一步,基于响应信息中的服务端的拆分包的版本号,终端102确定本地与业务模块对应的拆分包版本已更新至最新拆分包版本号。进一步,当最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,终端102加载最新拆分包至内存,基于加载的最新拆分包及基础库数据包,启动业务模块。

[0047] 其中,终端102可以但不限于各种个人计算机、笔记本电脑、智能手机、平板电脑和便携式可穿戴设备,服务器104可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

[0048] 在一个实施例中,如图2所示,提供了一种应用程序内业务模块启动方法,以该方法应用于图1中的终端102为例进行说明,包括以下步骤:

[0049] 步骤202,在应用程序启动时,向服务端请求与拆分包版本相关的响应信息。

[0050] 在本实施例中,基于JavaScript的开源框架React Native(以下简称为RN)的广泛应用,使得应用程序(Application,APP)摆脱了发版的限制,服务端发布RN代码,终端根据RN代码,在不关闭应用程序的情况下进行业务模块的更新。由于终端无法直接得知应用程序中业务模块的更新情况,因此,当用户启动应用程序时,终端需要向服务端请求响应信息。

[0051] 在本实施例中,终端获取应用程序对应的本地拆分包的版本号,将其作为参数,向服务端请求响应信息时携带。

[0052] 其中,响应信息可以包括但不限于:服务端与应用程序对应的基础库数据包的最新版本号、服务端与应用程序对应的拆分包的版本号、多个服务端与业务模块对应的拆分包的版本号、多个服务端与业务模块对应的拆分包的拆分包压缩文件下载地址。

[0053] 其中,基础库数据包(Common包)是指同一个应用程序中JavaScript基础库集合的产物,通常包含RN及App基础库。

[0054] 步骤204,接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号。

[0055] 在本实施例中,基于响应信息,终端获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号,可以用于匹配确定本地与应用程序对应的拆分包、本地与应用程序对应的基础库数据包是否为最新版本。

[0056] 步骤206,基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存。

[0057] 在本实施例中,当本地与应用程序对应的基础库数据包的版本号与服务端的基础库数据包的最新版本号不一致时,本地与应用程序对应的基础库数据包的版本号与服务端的基础库数据包的最新版本号不匹配,即基础库数据包存在更新。当基础库数据包存在更新时,终端基于响应信息下载最新版本号对应的基础库数据包到本地,即将本地基础库数据包更新至最新版本,再预加载该最新版本号对应的基础库数据包到内存。

[0058] 进一步,在本实施例中,当本地与应用程序对应的基础库数据包的版本号与服务端的基础库数据包的最新版本号一致时,本地与应用程序对应的基础库数据包的版本号与服务端的基础库数据包的最新版本号匹配,即本地与应用程序对应的基础库数据包的版本号已经是最新版本,此时终端无需下载新的基础库数据包,只要预加载此本地基础库数据包至内存。当本地基础库数据包产生变动时,由于本地基础库数据包的预加载,所有对应本地基础库数据包的拆分包不必同时更新变动,给对应本地基础库数据包的拆分包的更新预留了时间。

[0059] 步骤208,基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号。

[0060] 在本实施例中,终端将本地与业务模块对应的拆分包的版本号与服务端的拆分包的最新版本号进行匹配,确定拆分包是否存在更新。

[0061] 在本实施例中,当拆分包存在更新时,服务端与业务模块对应的拆分包版本为最

新拆分包版本,终端将其下载至本地,并获取其版本号。当拆分包不存在更新时,本地与业务模块对应的拆分包的版本号已经是最新拆分包版本,此时终端无需下载新的拆分包,只需获取本地拆分包的版本号并将其作为最新拆分包版本号,避免重复的下载。

[0062] 步骤210,确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存。

[0063] 在本实施例中,终端确认最新拆分包版本与基础库数据包的最新版本相匹配后再加载最新拆分包至内存,可以避免拆分包、基础库数据包版本不匹配导致的更新失败。

[0064] 步骤212,基于加载的最新拆分包及基础库数据包,启动业务模块。

[0065] 在本实施中,最新拆分包版本与基础库数据包的最新版本相匹配,从而拆分包可以调用基础库数据包,启动业务模块,完成应用程序内业务模块的更新。

[0066] 上述应用程序内业务模块启动方法中,首先在应用程序启动时,向服务端请求与拆分包版本相关的响应信息,从而获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号,用于后续的版本匹配。进一步,基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,从而实现将本地基础库数据包更新至最新版本,并加载最新版本号对应的基础库数据包到内存。进一步,基于服务端的拆分包的版本号,将本地与业务模块对应的拆分包版本已更新至最新拆分包版本号。进一步,确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,单独加载最新拆分包至内存,基于加载的最新拆分包及基础库数据包,启动业务模块,在服务端与业务模块对应的拆分包、服务端与业务模块对应的基础库数据包版本一致的情况下可以提高打开应用程序业务模块的速度。

[0067] 在其中一个实施例中,参考图3,向服务端请求与拆分包版本相关的响应信息,可以包括:获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求;向服务端发送请求,以请求与拆分包版本相关的响应信息。

[0068] 在本实施例中,应用程序对应的本地拆分包可以包括多个本地与业务模块对应的拆分包。其中,每个业务模块至少存在一个对应的拆分包。

[0069] 在本实施例中,当应用程序中存在业务模块A,那么,终端可以先获取至少一个本地与业务模块A对应的拆分包版本号,将该拆分包版本号作为参数,向服务端请求相应信息时携带,以请求获取与业务模块A对应的拆分包版本相关的响应信息。

[0070] 在本实施例中,终端将获取本地各业务模块对应的至少一个拆分包版本号,将各版本号作为参数,向服务端请求相应信息时携带,以请求获取与应用程序中各业务模块对应的拆分包版本相关的响应信息。

[0071] 在其中一个实施例中,参考图4,基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,可以包括:基于响应信息,获取服务端与业务模块对应的拆分包的版本号;当服务端与业务模块对应拆分包的版本号与本地与业务模块对应拆分包的版本号一致时,获取本地与业务模块对应的拆分包中版本已更新至最新拆分包版本号。

[0072] 在本实施例中,终端获取响应信息中服务端与业务模块对应的拆分包版本号,即服务端的拆分包的最新版本号,并将其与本地与业务模块对应的拆分包的版本号进行拆分包版本匹配。

[0073] 进一步,在本实施例中,当本地与业务模块对应的拆分包的版本号与服务端的拆分包的最新版本号一致时,本地与业务模块对应的拆分包的版本号已经是最新拆分包版本,此时终端无需下载新的拆分包,只需获取本地拆分包的版本号并将其作为最新拆分包版本号,避免重复的下载,以减少更新中需要下载的文件体积。

[0074] 在其中一个实施例中,基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,还可以包括:当多个服务端与业务模块对应拆分包的版本号、本地与业务模块对应拆分包的版本号不一致时,获取响应信息中的拆分包压缩文件下载地址;基于拆分包压缩文件下载地址,下载并解压与业务模块对应的拆分包至本地,使本地与业务模块对应的拆分包的版本更新至最新拆分包版本号。

[0075] 在本实施例中,本地拆分包可以包括:本地与应用程序对应拆分包、多个本地与业务模块对应的拆分包等。

[0076] 当本地与业务模块对应的拆分包的版本号与服务端的拆分包的最新版本号不一致时,即拆分包存在更新。当拆分包存在更新时,终端基于响应信息中拆分包压缩文件的下载地址,下载最新版本号对应的拆分包到本地,即将本地拆分包更新至最新版本,并获取最新拆分包版本号。

[0077] 在本实施例中,响应信息中还可以包括更新指令,终端基于响应信息中拆分包压缩文件的下载地址,下载最新版本号对应的拆分包到本地,可以包括:获取响应信息中的拆分包压缩文件下载地址及更新指令。

[0078] 进一步,在本实施例中,当更新指令为不强制更新,基于响应信息中的拆分包压缩文件下载地址,与业务模块对应的拆分包压缩文件将进行后台静默下载至本地;当更新指令为强制更新,基于响应信息中的拆分包压缩文件下载地址,终端将与业务模块对应的拆分包压缩文件下载解压至本地。

[0079] 在其中一个实施例中,参考图4,获取与业务模块对应的拆分包的最新拆分包版本号后,还可以包括:当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块。

[0080] 其中,业务包是指实现某个业务功能的完整RN产物,可单独使用,也称作全包。

[0081] 在本实施例中,在获取到最新拆分包版本号后,终端将最新拆分包版本号与服务端基础库数据包的最新版本号进行匹配,当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,拆分包无法调用基础库数据包启动业务模块。此时,终端获取与最新拆分包版本号对应的业务包,将其加载至内存,由该业务包启动业务模块。

[0082] 在其中一个实施例中,继续参考图4,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块,可以包括:当本地存在业务包时,加载业务包至内存,启动业务模块;当本地不存在业务包时,向服务端请求下载业务包至本地,再加载业务包至内存,启动业务模块。

[0083] 在本实施例中,参考图4,终端获取与最新拆分包版本号对应的业务包之前,要先识别本地是否存在最新拆分包版本号对应的业务包,以避免重复下载,提高业务模块启动速度。

[0084] 在本实施例中,终端先获取本地与业务模块对应的业务包的版本号、最新拆分包版本号对应的业务包版本号,并将二者进行比较。进一步,当本地与业务模块对应的业务包

的版本号、最新拆分包版本号对应的业务包版本号相一致,即本地存在最新拆分包对应的业务包,则终端将该业务包加载至内存,由业务包启动业务模块。当本地与业务模块对应的业务包的版本号、最新拆分包版本号对应的业务包版本号不一致,即本地不存在最新拆分包对应的业务包,则终端将向服务端请求下载最新拆分包版本号对应的业务包,再将该业务包加载至内存,由业务包启动业务模块。

[0085] 在其中一个实施例中,继续参考图4,确定最新拆分包版本号与基础库数据包的最新版本号匹配后,还可以包括:若最新版本号对应的基础库数据包未加载完成,停止最新版本号对应的基础库数据包的加载,合并加载最新版本号对应的基础库数据包、本地与业务模块对应拆分包到内存。

[0086] 在本实施例中,当终端确定最新拆分包版本号与基础库数据包的最新版本号匹配后,还可以确认基础库数据包的最新版本的预加载情况。

[0087] 其中,基础库数据包的最新版本的预加载情况可以包括:已加载完成、未加载完成、加载失败等。

[0088] 在本实施例中,当基础库数据包的最新版本已加载完成,则拆分包可以直接调用基础库数据包,启动业务模块。当基础库数据包的最新版本未加载完成或加载失败时,终端可以停止最新版本号对应的基础库数据包的预加载,合并加载最新版本号对应的基础库数据包及拆分包到内存,再启动业务模块。

[0089] 应该理解的是,虽然图2-4的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,图2-4中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些子步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0090] 在一个实施例中,如图5所示,提供了一种应用程序内业务模块启动装置,包括:响应信息请求模块100、响应信息接收模块200、预加载模块300、最新拆分包版本号获取模块400、拆分包加载模块500和业务模块启动模块600,其中:

[0091] 响应信息请求模块100,用于在应用程序启动时,向服务端请求与拆分包版本相关的响应信息。

[0092] 响应信息接收模块200,用于接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号。

[0093] 预加载模块300,用于基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存。

[0094] 拆分包更新模块400,用于基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号。

[0095] 拆分包加载模块500,用于确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存。

[0096] 业务模块启动模块600,用于基于加载的最新拆分包及基础库数据包,启动业务模

块。

[0097] 在其中一个实施例中,响应信息请求模块100,可以包括:

[0098] 相应信息请求生成子模块,用于获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求。

[0099] 响应信息请求发送子模块,用于向服务端发送请求,以请求与拆分包版本相关的响应信息。

[0100] 在其中一个实施例中,拆分包更新模块400,可以包括:

[0101] 拆分包版本号获取子模块,用于基于响应信息,获取服务端与业务模块对应的拆分包的版本号。

[0102] 拆分包版本号获取子模块,用于当服务端与业务模块对应拆分包的版本号与本地与业务模块对应拆分包的版本号一致时,获取本地与业务模块对应的拆分包中版本已更新至最新拆分包版本号。

[0103] 在其中一个实施例中,拆分包更新模块400,还可以包括:

[0104] 压缩文件下载地址获取子模块,用于当多个服务端与业务模块对应拆分包的版本号、本地与业务模块对应拆分包的版本号不一致时,获取响应信息中的拆分包压缩文件下载地址。

[0105] 压缩文件下载解压子模块,用于基于拆分包压缩文件下载地址,下载并解压与业务模块对应的拆分包至本地,使本地与业务模块对应的拆分包的版本更新至最新拆分包版本号。

[0106] 在其中一个实施例中,上述装置还可以包括:

[0107] 业务包获取模块,用于当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块。

[0108] 在其中一个实施例中,业务包获取模块,可以包括:

[0109] 本地业务包加载子模块,用于当本地存在业务包时,加载业务包至内存,启动业务模块。

[0110] 服务端业务包下载加载子模块,用于当本地不存在业务包时,向服务端请求下载业务包至本地,再加载业务包至内存,启动业务模块。

[0111] 在其中一个实施例中,上述装置还可以包括:

[0112] 合并加载模块,用于若最新版本号对应的基础库数据包未加载完成,停止最新版本号对应的基础库数据包的加载,合并加载最新版本号对应的基础库数据包、本地与业务模块对应拆分包到内存。

[0113] 关于应用程序内业务模块启动装置的具体限定可以参见上文中对于应用程序内业务模块启动方法的限定,在此不再赘述。上述应用程序内业务模块启动装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中,也可以以软件形式存储于计算机设备中的存储器中,以便于处理器调用执行以上各个模块对应的操作。

[0114] 在一个实施例中,提供了一种计算机设备,该计算机设备可以是服务器,其内部结构图可以如图6所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中,该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包

括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储基础库数据包、拆分包及业务包的版本号数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种应用程序内业务模块启动方法。

[0115] 本领域技术人员可以理解,图6中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0116] 在一个实施例中,提供了一种计算机设备,包括存储器和处理器,该存储器存储有计算机程序,该处理器执行计算机程序时实现以下步骤:在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号;基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存;基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号;确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存;基于加载的最新拆分包及基础库数据包,启动业务模块。

[0117] 在其中一个实施例中,处理器执行计算机程序时可以实现:向服务端请求与拆分包版本相关的响应信息,包括:获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求;向服务端发送请求,以请求与拆分包版本相关的响应信息。

[0118] 在其中一个实施例中,处理器执行计算机程序时可以实现:基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,包括:基于响应信息,获取服务端与业务模块对应的拆分包的版本号;当服务端与业务模块对应拆分包的版本号与本地与业务模块对应拆分包的版本号一致时,获取本地与业务模块对应的拆分包中版本已更新至最新拆分包版本号。

[0119] 在其中一个实施例中,处理器执行计算机程序时可以实现:基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,还包括:当多个服务端与业务模块对应拆分包的版本号、本地与业务模块对应拆分包的版本号不一致时,获取响应信息中的拆分包压缩文件下载地址;基于拆分包压缩文件下载地址,下载并解压与业务模块对应的拆分包至本地,使本地与业务模块对应的拆分包的版本更新至最新拆分包版本号。

[0120] 在其中一个实施例中,处理器执行计算机程序时可以实现:获取与业务模块对应的拆分包的最新拆分包版本号后,还包括:当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块。

[0121] 在其中一个实施例中,处理器执行计算机程序时可以实现:获取与最新拆分包版本号对应的业务包,由业务包启动业务模块,包括:当本地存在业务包时,加载业务包至内存,启动业务模块;当本地不存在业务包时,向服务端请求下载业务包至本地,再加载业务包至内存,启动业务模块。

[0122] 在其中一个实施例中,处理器执行计算机程序时可以实现:确定最新拆分包版本

号与基础库数据包的最新版本号匹配后,还包括:若最新版本号对应的基础库数据包未加载完成,停止最新版本号对应的基础库数据包的加载,合并加载最新版本号对应的基础库数据包、本地与业务模块对应拆分包到内存。

[0123] 在一个实施例中,提供了一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现以下步骤:在应用程序启动时,向服务端请求与拆分包版本相关的响应信息;接收服务端下发的响应信息,基于响应信息获取服务端与应用程序对应的拆分包的版本号,以及服务端与应用程序对应的基础库数据包的最新版本号;基于最新版本号,将本地与应用程序相对应的基础库数据包更新至最新版本,并加载到内存;基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号;确定最新拆分包版本号与基础库数据包的最新版本号匹配,并且基础库数据包已加载完成,加载最新拆分包版本号对应的最新拆分包至内存;基于加载的最新拆分包及基础库数据包,启动业务模块。

[0124] 在其中一个实施例中,计算机程序被处理器执行时可以实现:向服务端请求与拆分包版本相关的响应信息,包括:获取应用程序对应的本地拆分包的版本号,将其作为参数生成对应的请求;向服务端发送请求,以请求与拆分包版本相关的响应信息。

[0125] 在其中一个实施例中,计算机程序被处理器执行时可以实现:基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,包括:基于响应信息,获取服务端与业务模块对应的拆分包的版本号;当服务端与业务模块对应拆分包的版本号与本地与业务模块对应拆分包的版本号一致时,获取本地与业务模块对应的拆分包中版本已更新至最新拆分包版本号。

[0126] 在其中一个实施例中,计算机程序被处理器执行时可以实现:基于服务端的拆分包的版本号,确定本地与业务模块对应的拆分包的版本已更新至最新拆分包版本号,还包括:当多个服务端与业务模块对应拆分包的版本号、本地与业务模块对应拆分包的版本号不一致时,获取响应信息中的拆分包压缩文件下载地址;基于拆分包压缩文件下载地址,下载并解压与业务模块对应的拆分包至本地,使本地与业务模块对应的拆分包的版本更新至最新拆分包版本号。

[0127] 在其中一个实施例中,计算机程序被处理器执行时可以实现:获取与业务模块对应的拆分包的最新拆分包版本号后,还包括:当最新拆分包版本号与服务端基础库数据包的最新版本号不匹配时,获取与最新拆分包版本号对应的业务包,由业务包启动业务模块。

[0128] 在其中一个实施例中,计算机程序被处理器执行时可以实现:获取与最新拆分包版本号对应的业务包,由业务包启动业务模块,包括:当本地存在业务包时,加载业务包至内存,启动业务模块;当本地不存在业务包时,向服务端请求下载业务包至本地,再加载业务包至内存,启动业务模块。

[0129] 在其中一个实施例中,计算机程序被处理器执行时可以实现:确定最新拆分包版本号与基础库数据包的最新版本号匹配后,还包括:若最新版本号对应的基础库数据包未加载完成,停止最新版本号对应的基础库数据包的加载,合并加载最新版本号对应的基础库数据包、本地与业务模块对应拆分包到内存。

[0130] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一非易失性计算机

可读取存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限,RAM以多种形式可得,诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0131] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0132] 以上所述实施例仅表达了本申请的几种实施方式,其描述较为具体和详细,但不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。

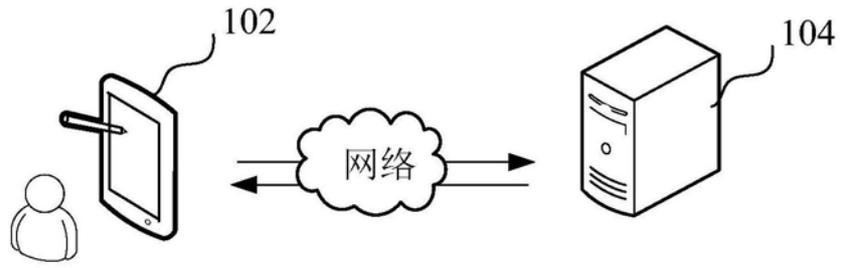


图1

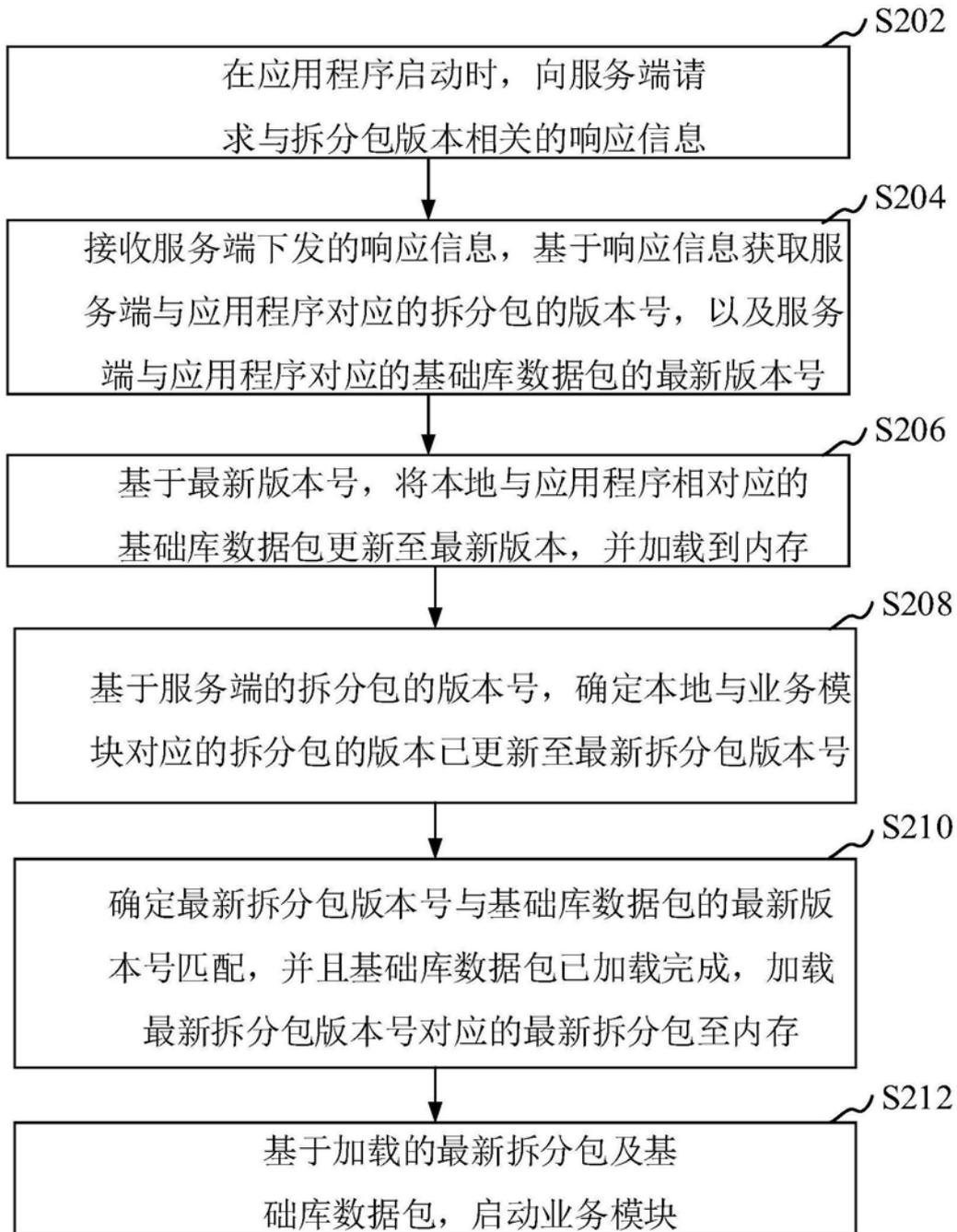


图2

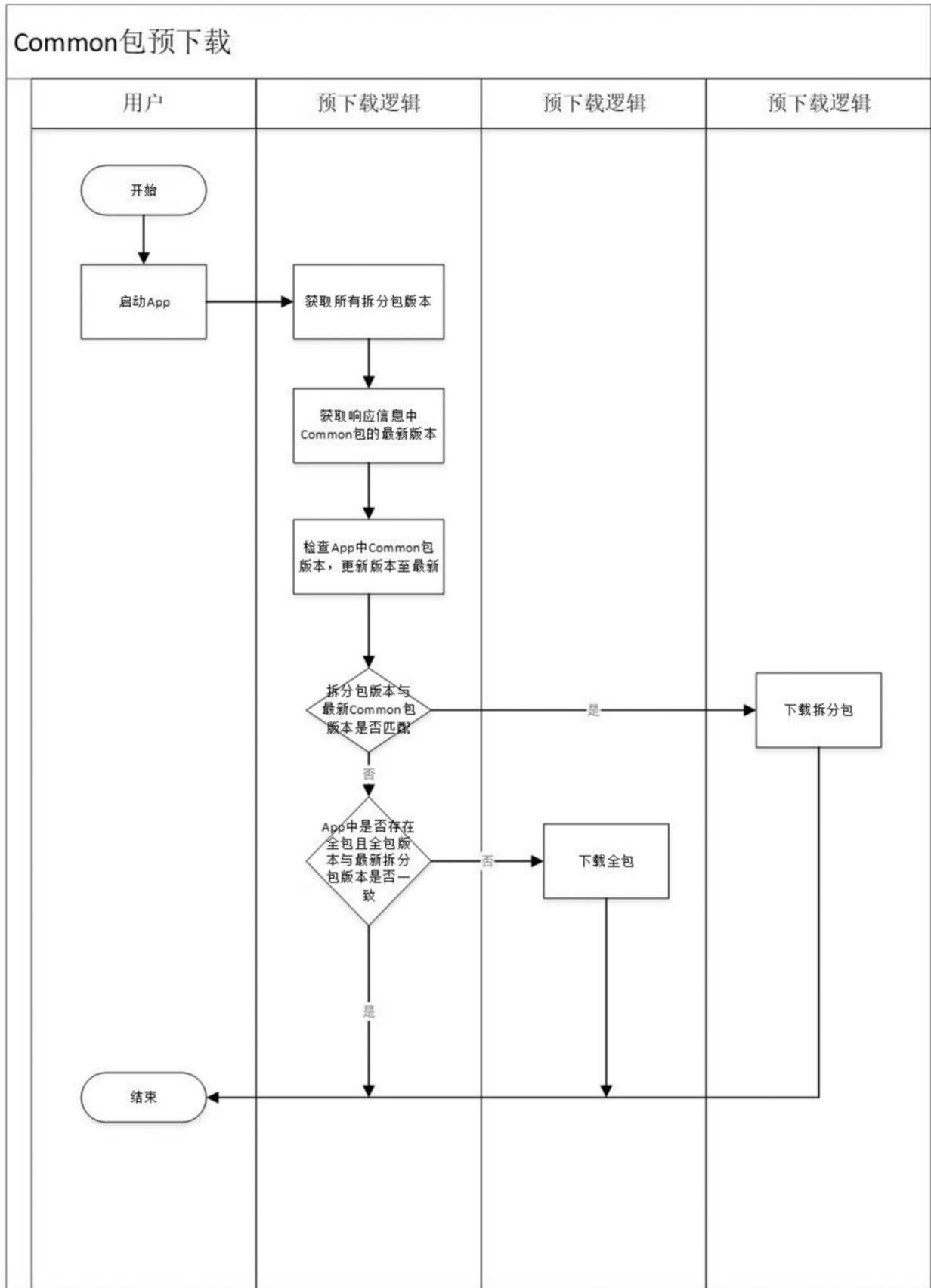


图3

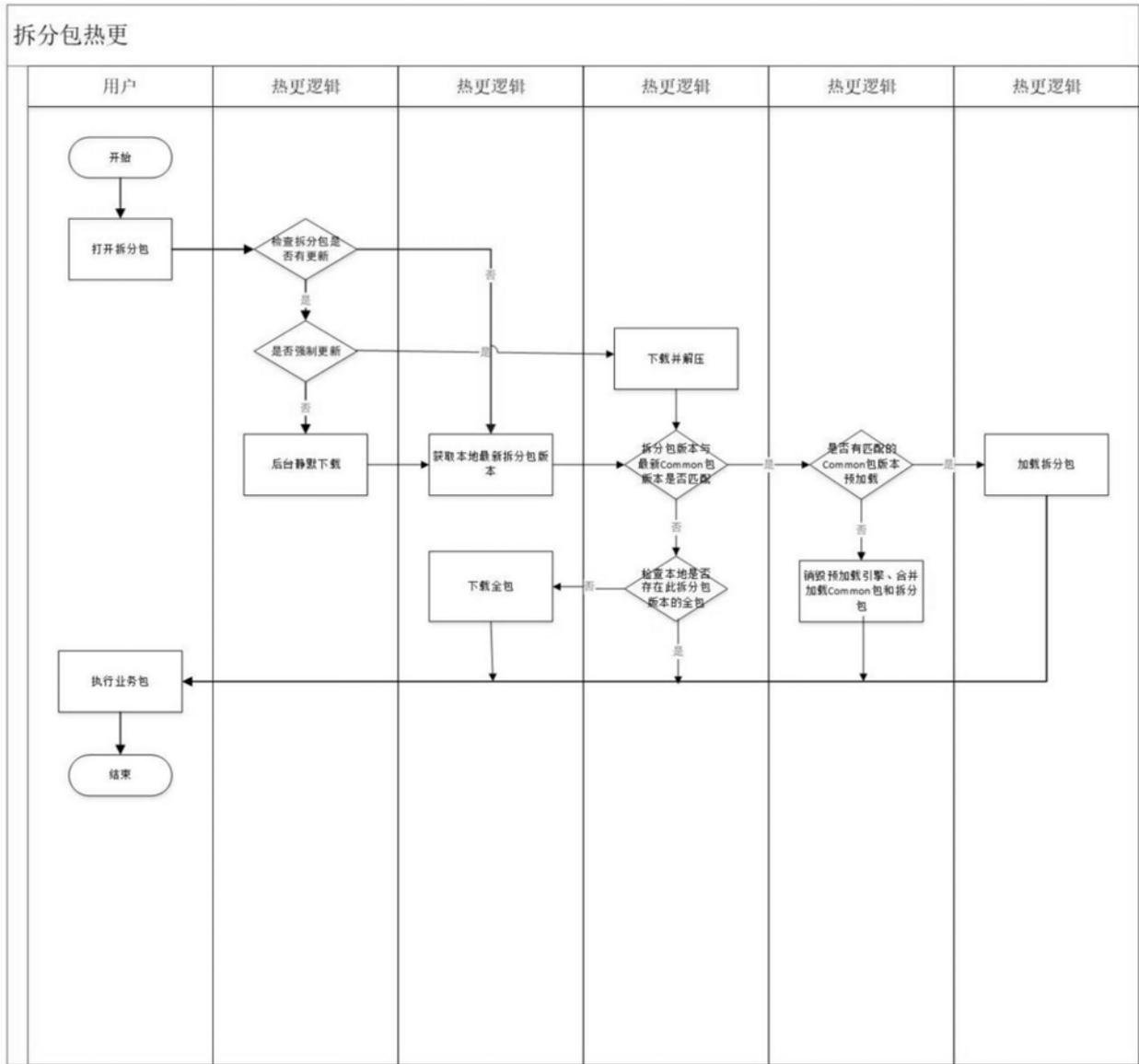


图4

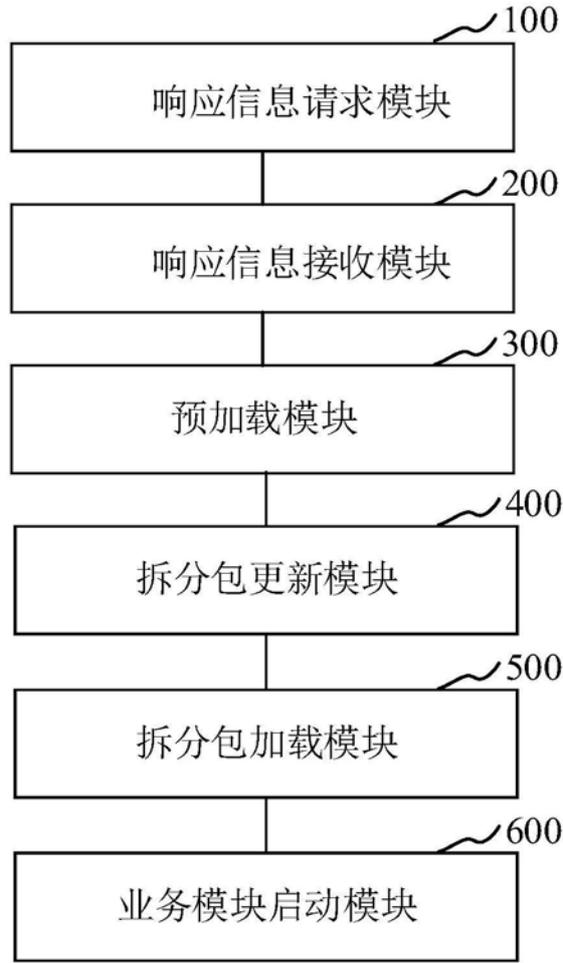


图5

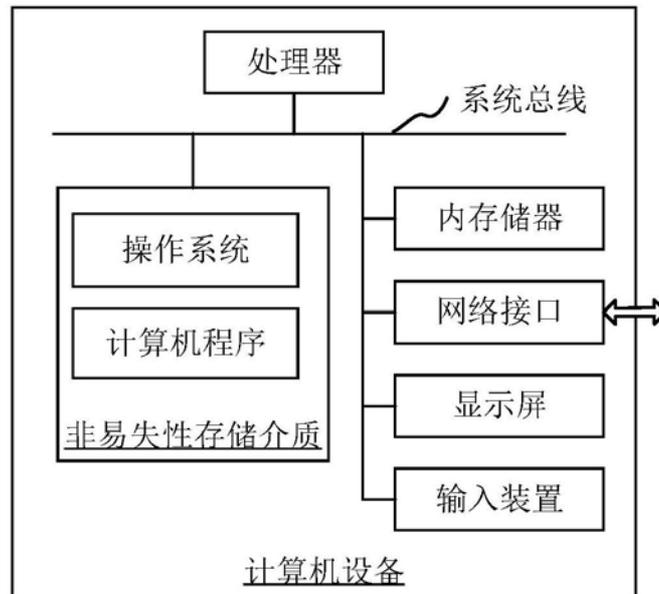


图6