



(12) 发明专利申请

(10) 申请公布号 CN 103176809 A

(43) 申请公布日 2013. 06. 26

(21) 申请号 201110438889. 8

(22) 申请日 2011. 12. 23

(71) 申请人 腾讯科技(深圳)有限公司

地址 518044 广东省深圳市福田区振兴路赛格科技园 2 栋东 403 室

(72) 发明人 戴维

(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 何平 曾旻辉

(51) Int. Cl.

G06F 9/445(2006. 01)

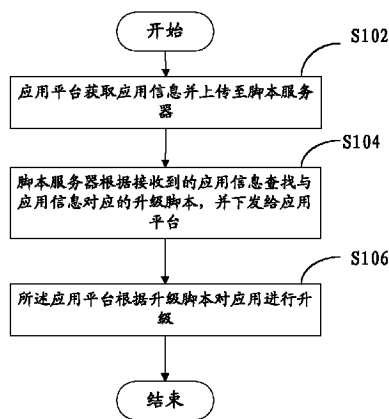
权利要求书2页 说明书6页 附图2页

(54) 发明名称

应用升级方法及系统

(57) 摘要

一种应用升级方法,所述方法包括以下步骤:应用平台获取应用信息并上传至脚本服务器;所述脚本服务器根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并下发给应用平台;所述应用平台根据所述升级脚本对应用进行升级。此外,还对应提供了一种应用升级系统。上述应用升级方法和系统可以有效地提高应用升级时的灵活性。



1. 一种应用升级方法,所述方法包括以下步骤:
应用平台获取应用信息并上传至脚本服务器;
所述脚本服务器根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并下发给应用平台;
所述应用平台根据所述升级脚本对应用进行升级。
2. 根据权利要求 1 所述的应用升级方法,其特征在于,所述升级脚本中包括升级指令序列;
所述应用平台根据所述升级脚本对应用进行升级的步骤具体为:
所述应用平台解析所述升级脚本获取升级指令序列,并依次解释执行所述升级指令序列中的升级指令进行升级。
3. 根据权利要求 2 所述的应用升级方法,其特征在于,所述应用信息至少包括应用标识和应用版本号;
所述脚本服务器根据接收到的应用信息查找与应用信息对应的升级脚本,并下发给应用平台的步骤包括:
所述脚本服务器根据所述应用标识查找与所述应用标识对应的升级脚本,获取查找到的升级脚本中的应用版本号比接收到的应用版本号高的升级脚本列表。
4. 根据权利要求 3 所述的应用升级方法,其特征在于,所述脚本服务器根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并下发给应用平台的步骤还包括:
所述脚本服务器获取应用平台选择的需升级的应用标识,根据所述应用标识从所述升级脚本列表中提取对应的升级脚本,并将所述升级脚本下发给应用平台。
5. 根据权利要求 2 至 4 中任意一项所述的应用升级方法,其特征在于,所述升级指令序列包括用于从数据服务器下载升级数据的升级指令;
所述依次解释执行所述升级指令序列中的升级指令的步骤包括:
根据所述升级指令从数据服务器下载升级数据。
6. 一种应用升级系统,其特征在于,包括应用平台和脚本服务器,所述应用平台包括:
应用信息获取模块,用于获取应用信息;
应用信息上传模块,用于将所述应用信息获取模块获取到的应用信息上传至所述脚本服务器;
所述脚本服务器包括:
脚本查找模块,用于根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并将查找到的所述升级脚本下发给所述应用平台;
所述应用平台还包括:
升级模块,用于根据所述升级脚本对应用进行升级。
7. 根据权利要求 6 所述的应用升级系统,其特征在于,所述升级脚本包括升级指令序列;
所述升级模块还用于解析所述升级脚本获取升级指令序列,并依次解释执行所述升级指令序列中的升级指令对应用进行升级。
8. 根据权利要求 7 所述的应用升级系统,其特征在于,所述应用信息至少包括应用标识和应用版本号;

所述查找模块用于根据所述应用标识查找与所述应用标识对应的升级脚本,获取查找到的升级脚本中的应用版本号比接收到的应用版本号高的升级脚本列表。

9. 根据权利要求 8 所述的应用升级系统,其特征在于,所述脚本查找模块还用于获取应用平台选择的需升级的应用标识,根据所述应用标识从所述升级脚本列表中提取对应的升级脚本。

10. 根据权利要求 7 至 9 中任意一项所述的应用升级系统,其特征在于,所述系统还包括用于存储升级数据的数据服务器;

所述升级指令序列包括用于从数据服务器下载升级数据的升级指令;

所述升级模块还用于根据所述升级指令从所述数据服务器下载升级数据。

应用升级方法及系统

【技术领域】

[0001] 本发明涉及网络技术领域,特别涉及一种应用升级方法及系统。

【背景技术】

[0002] 应用平台是一种能够聚合各种第三方软件提供商所开发的应用(俗称 app),使得应用对用户的表现形式标准化、规范化的平台类服务软件。第三方应用提供商通过应用平台对外开放的应用程序接口(API)与应用平台对接,为应用平台提供相应的功能。应用平台则为应用程序提供基础的配套服务,如空间存储、平台调用、网络传输等。

[0003] 传统的应用平台中,平台上的每个应用在发布成功(添加到应用平台的应用数据库中)之后均被存储为一份独立的数据包,并在应用列表中注册该应用的应用标识。平台上的应用在升级和部署时,先在应用平台的应用列表中查找该应用标识,然后下载与该应用标识对应的数据包,通过解包完成应用的升级和部署。

[0004] 然而,传统技术中的应用平台上的应用在升级时,需要将新版本的应用的整个数据包完全重新下载一遍,并覆盖本地所有文件,因此应用升级的方法单一,仅仅包括文件替换的方式。当在应用升级过程中,某些文件需要系统注册时,则该应用升级方法无法满足需求,从而使得传统技术中的应用升级方法灵活性不足。且由于需要重新下载,使得传统技术中的应用升级方法的网络带宽利用率不高。

【发明内容】

[0005] 基于此,有必要提供一种更加灵活的应用升级方法。

[0006] 一种应用升级方法,所述方法包括以下步骤:

[0007] 应用平台获取应用信息并上传至脚本服务器;

[0008] 所述脚本服务器根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并下发给应用平台;

[0009] 所述应用平台根据所述升级脚本对应用进行升级。

[0010] 优选的,所述升级脚本中包括升级指令序列;

[0011] 所述应用平台根据所述升级脚本对应用进行升级的步骤具体为:

[0012] 所述应用平台解析所述升级脚本获取升级指令序列,并依次解释执行所述升级指令序列中的升级指令进行升级。

[0013] 优选的,所述应用信息至少包括应用标识和应用版本号;

[0014] 所述脚本服务器根据接收到的应用信息查找与应用信息对应的升级脚本,并下发给应用平台的步骤包括:

[0015] 所述脚本服务器根据所述应用标识查找与所述应用标识对应的升级脚本,获取查找到的升级脚本中的应用版本号比接收到的应用版本号高的升级脚本列表。

[0016] 优选的,所述脚本服务器根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并下发给应用平台的步骤还包括:

- [0017] 所述脚本服务器获取应用平台选择的需升级的应用标识,根据所述应用标识从所述升级脚本列表中提取对应的升级脚本,并将所述升级脚本下发给应用平台。
- [0018] 优选的,所述升级指令序列包括用于从数据服务器下载升级数据的升级指令;
- [0019] 所述依次解释执行所述升级指令序列中的升级指令的步骤包括:
- [0020] 根据所述升级指令从数据服务器下载升级数据。
- [0021] 此外,还有必要提供一种更加灵活的应用升级系统。
- [0022] 一种应用升级系统,包括应用平台和脚本服务器,所述应用平台包括:
- [0023] 应用信息获取模块,用于获取应用信息;
- [0024] 应用信息上传模块,用于将所述应用信息获取模块获取到的应用信息上传至所述脚本服务器;
- [0025] 所述脚本服务器包括:
- [0026] 脚本查找模块,用于根据接收到的所述应用信息查找与所述应用信息对应的升级脚本,并将查找到的所述升级脚本下发给所述应用平台;
- [0027] 所述应用平台还包括:
- [0028] 升级模块,用于根据所述升级脚本对应用进行升级。
- [0029] 优选的,所述升级脚本包括升级指令序列;
- [0030] 所述升级模块还用于解析所述升级脚本获取升级指令序列,并依次解释执行所述升级指令序列中的升级指令对应用进行升级。
- [0031] 优选的,所述应用信息至少包括应用标识和应用版本号;
- [0032] 所述查找模块用于根据所述应用标识查找与所述应用标识对应的升级脚本,获取查找到的升级脚本中的应用版本号比接收到的应用版本号高的升级脚本列表。
- [0033] 优选的,所述脚本查找模块还用于获取应用平台选择的需升级的应用标识,根据所述应用标识从所述升级脚本列表中提取对应的升级脚本。
- [0034] 优选的,所述系统还包括用于存储升级数据的数据服务器;
- [0035] 所述升级指令序列包括用于从数据服务器下载升级数据的升级指令;
- [0036] 所述升级模块还用于根据所述升级指令从所述数据服务器下载升级数据。
- [0037] 上述应用升级方法和系统,脚本服务器根据应用平台获取并上传的应用信息查找与应用信息对应的预设的升级脚本,并下发给应用平台,应用平台根据升级脚本进行升级。由于升级脚本与应用信息对应,即预先可根据应用的类型设置不同的升级脚本,在升级时获取应用对应的升级脚本实现升级即可,从而提高了应用升级方法和系统的灵活性。

【附图说明】

- [0038] 图 1 为一个实施例中应用升级方法的流程图;
- [0039] 图 2 为一个实施例中应用升级系统的结构示意图;
- [0040] 图 3 为另一个实施例中应用升级系统的结构示意图。

【具体实施方式】

- [0041] 如图 1 所示,在一个实施例中,一种应用升级方法,包括以下步骤:
- [0042] 步骤 S102,应用平台获取应用信息并上传至脚本服务器。

[0043] 应用平台可以是浏览器,也可以是桌面程序。当应用平台启动时,可自动获取应用平台内已存储的应用信息或根据用户选择的需升级的应用,然后将应用信息上传到脚本服务器。具体的,当应用平台是浏览器时,应用平台可在浏览器缓存中获取应用信息;当应用平台为桌面程序时,应用平台可在硬盘上的应用存储区获取应用信息。

[0044] 在一个实施例中,应用平台还可根据用户输入的选取指令获取应用信息并上传至脚本服务器。

[0045] 在一个实施例中,应用信息至少包括应用标识和应用版本号。则步骤 S102 中,应用平台获取应用标识和应用版本号并上传至脚本服务器。应用标识可以是 UUID(Universally Unique Identifier,全局唯一识别码),也可以是数字。在应用发布时,应用标识可由脚本服务器自动为应用分配,用于唯一标识应用。

[0046] 步骤 S104,脚本服务器根据接收到的应用信息查找与应用信息对应的升级脚本,并下发给应用平台。

[0047] 具体的,脚本服务器存储了预先设置好的升级脚本。脚本服务器接收到应用平台上传的应用信息后,可先从应用信息中提取出应用标识,并根据应用标识查找与该应用标识对应的升级脚本,获取查找到的升级脚本中的应用版本号比接收到的应用版本号高的升级脚本列表。

[0048] 在一个实施例中,脚本服务器获取到升级脚本列表后,可自动将升级脚本列表中的升级脚本下发给应用平台。在另一个实施例中,脚本服务器获取到升级脚本列表后,还可将升级脚本列表返回给应用平台,等待用户选择需升级的应用标识,然后根据该应用标识从升级脚本列表中提取对应的升级脚本,并将该升级脚本下发给应用平台,实现手动升级应用。

[0049] 步骤 S106,应用平台根据升级脚本对应用进行升级。

[0050] 升级脚本可以以文件的形式存储在脚本服务器上,也可以以二进制流的形式存储于脚本服务器中。升级脚本中可包含与应用信息对应的基本属性字段。在一个实施例中,基本属性字段可包括应用标识、应用名称、应用版本号、应用类型、应用描述信息中的一种或多种。其中,应用类型用于标识应用所属类型,包括 Web 应用(网页类型的应用)、Exe 应用(应用平台类型的应用)、Dll 应用(插件类型的应用)、Flash 应用(Flash 动画类型的应用)等。

[0051] 在一个实施例中,升级脚本中还可包括升级指令序列。升级指令序列是由升级指令按预设的顺序排列组成的序列。基本的升级指令可包括下载指令、解压指令、移动文件指令、替换文件指令、创建文件指令、删除文件指令、文件重命名指令、注册文件指令、删除缓存文件中的一种或多种。

[0052] 进一步的,可根据实际升级需要在基本的升级指令库中选取恰当的升级指令,然后将升级指令排列成升级指令序列,并记录在升级脚本中。

[0053] 在一个实施例中,应用平台可解析升级脚本获取升级指令序列,并依次解释执行升级指令序列中的升级指令进行升级。

[0054] 例如,应用平台接收到的升级脚本,并从升级脚本中获取的升级指令序列如表 1 所示:

[0055] 表 1

[0056]

```
AppID: 200002001
AppName: Dll 应用
AppVersion: 1.0.0
AppType: dll
AppDesc: Dll 应用描述
DownloadFile http://appstore.qplus.com/200002001/config.json
ReplaceFile config.json config.json
DownloadFile http://appstore.qplus.com/200002001/icon_48.png
ReplaceFile icon_48.png icon_48.png
DownloadFile http://appstore.qplus.com/200002001/main.dll
ReplaceFile main.dll main.dll
RegFile main.dll
NewFile Setup.ini
```

[0057] 其中,AppID 表示应用标识,AppName 表示应用名称,AppVersion 表示应用版本号,AppType 表示应用类型。DownloadFile 表示下载文件指令,ReplaceFile 表示替换文件指令,RegFile 表示注册文件指令,newFile 表示创建文件指令。

[0058] 应用平台在解释执行该升级指令序列时,先执行 DownloadFile 指令,下载 config.json 文件,然后将本地的 config.json 文件替换。然后继续下载 icon_48.png 文件,将本地的 icon_48.png 文件替换。然后下载 main.dll 文件,先替换本地的 main.dll 文件,并将 main.dll 在操作系统中注册。最后创建 setup.ini 文件,通知应用平台升级完毕。

[0059] 进一步的,在一个实施例中,升级指令序列包括用于从数据服务器下载升级数据的升级指令。应用平台依次解释执行所述升级指令序列中的升级指令时,则根据用于该升级指令从数据服务器下载升级数据。

[0060] 本实施例中,当应用平台上的应用具有升级内容需要应用平台进行升级时,将升级脚本存入脚本服务器,将升级数据存入数据服务器。数据服务器中的升级数据可通过包含 URL 形式的链接的升级指令获取。在解释执行升级指令时,可通过执行包含 URL 形式的链接的升级指令从数据服务器获取升级数据。

[0061] 本实施例中,将升级脚本和升级数据分开存储,可以实现升级数据的资源共享。若多个应用在升级时,均使用了同一文件,则可以在数据服务器中仅保存该文件的一个副本,然后在与多个应用对应的多个升级脚本中使用链接表示该文件,从而节省了存储空间。

[0062] 在一个实施例中,升级脚本中还包括顺序控制指令。顺序控制指令用于限定升级指令序列的执行顺序。顺序控制指令包括循环指令和分支指令。应用平台依次解释执行升级指令序列中的升级指令时,可根据循环指令循环执行多条升级指令,也可根据分支指令

在执行过程中进行条件判断,选择与条件判断结果相符的升级指令解释执行。

[0063] 在升级脚本中添加了顺序控制指令后可以对升级指令的执行顺序进行更加详细的限定。例如,在执行升级指令时,可通过执行分支指令对应用平台的运行环境进行判断,然后再选择与应用平台的运行环境相符合的升级指令执行。从而使得升级指令的执行不再限于顺序执行顺序,使得应用升级方法更加灵活。

[0064] 如图 2 所示,在一个实施例中,一种应用升级系统,包括应用平台 100 和脚本服务器 200,其中:

[0065] 应用平台 100 包括应用信息获取模块 102、应用信息上传模块 104 和升级模块 106,其中:应用信息获取模块 102 用于获取应用信息;应用信息上传模块 104 用于将应用信息获取模块 102 获取到的应用信息上传到脚本服务器 200。

[0066] 应用平台 100 可以是浏览器,也可以是桌面程序。当应用平台 100 启动时,应用信息获取模块 102 可用于自动获取应用平台 100 内已存储的应用信息,应用信息上传模块 104 可用于将该信息上传至脚本服务器 200。当应用平台 100 是浏览器时,应用信息获取模块 102 可用于在浏览器缓存中获取应用信息;当应用平台 100 为桌面程序时,应用信息获取模块 102 可用于在硬盘上的应用存储区获取应用信息。

[0067] 在一个实施例中,应用信息获取模块 102 还可用于根据用户输入的选取指令获取应用信息,并通过应用信息上传模块 104 将应用信息上传到脚本服务器 200。

[0068] 在一个实施例中,应用信息至少包括应用标识和应用版本号,则应用信息获取模块 102 获取应用标识和应用版本号。应用标识可以是 UUID(Universally Unique Identifier,全局唯一识别码),也可以是数字。在应用发布时,应用标识可由脚本服务器 200 自动为应用分配,用于唯一标识应用。

[0069] 脚本服务器 200 包括脚本查找模块 202,其中:脚本查找模块 202 用于根据接收到的应用信息查找与应用信息对应的升级脚本,并将查找到的升级脚本下发给应用平台 100。

[0070] 具体的,脚本服务器 200 存储了预先设置好的升级脚本。脚本查找模块 202 用于接收到应用平台上传的应用信息后,可先从应用信息中提取出应用标识,并根据应用标识查找与该应用标识对应的升级脚本,获取查找到的升级脚本中对应的应用版本号比接收到的应用版本号高的升级脚本列表。

[0071] 在一个实施例中,脚本查找模块 202 获取到升级脚本列表后,可由脚本查找模块 204 自动将升级脚本列表中的升级脚本下发给应用平台 100。在另一个实施例中,脚本查找模块 202 获取到升级脚本列表后,还可由脚本查找模块 204 将升级脚本列表返回给应用平台 100,等待用户选择需升级的应用标识,然后根据该应用标识从升级脚本列表中提取对应的升级脚本,并将该升级脚本下发给应用平台 100,实现手动升级应用。

[0072] 本实施例中,升级模块 106 用于根据升级脚本对应用进行升级。

[0073] 升级脚本可以以文件的形式存储在脚本服务器 200 上,也可以以二进制流的形式存储于脚本服务器 200 中。升级脚本中可包含与应用信息对应的基本属性字段。在一个实施例中,基本属性字段可包括应用标识、应用名称、应用版本号、应用类型、应用描述信息中的一种或多种。其中,应用类型用于标识应用所属类型,包括 Web 应用(网页类型的应用)、Exe 应用(应用平台类型的应用)、Dll 应用(插件类型的应用)、Flash 应用(Flash 动画类型的应用)等。

[0074] 在一个实施例中,升级脚本中还可包括升级指令序列。升级指令序列是由升级指令按预设的顺序排列组成的序列。基本的升级指令可包括下载指令、解压指令、移动文件指令、替换文件指令、创建文件指令、删除文件指令、文件重命名指令、注册文件指令、删除缓存文件中的一种或多种。

[0075] 进一步的,可根据实际升级需要在基本的升级指令库中选取恰当的升级指令,然后将升级指令排列成升级指令序列,并记录在升级脚本中。

[0076] 在一个实施例中,升级模块 106 可用于解析升级脚本获取升级指令序列,并依次解释执行升级指令序列中的升级指令进行升级。

[0077] 进一步的,在一个实施例中,如图 3 所示,应用升级系统还包括用于存储升级数据的数据服务器 300。升级指令序列包括用于从数据服务器 300 下载升级数据的升级指令。本实施例中,升级模块 106 还用于根据用于升级指令从数据服务器 300 下载升级数据。

[0078] 本实施例中,将升级脚本存入脚本服务器 200,将升级数据存入数据服务器 300。数据服务器 300 中的升级数据可通过包含 URL 形式的链接的升级指令获取。在解释执行升级指令时,可通过执行包含 URL 形式的链接的升级指令从数据服务器 300 获取升级数据。

[0079] 本实施例中,将升级脚本和升级数据分开存储,可以实现升级数据的资源共享。若多个应用在升级时,均使用了同一文件,则可以在数据服务器 300 中仅保存该文件的一个副本,然后在与多个应用对应的多个升级脚本中使用链接表示该文件,从而节省了存储空间。

[0080] 在一个实施例中,升级脚本中还包括顺序控制指令。顺序控制指令用于限定升级指令序列的执行顺序。顺序控制指令包括循环指令和分支指令。升级模块 106 依次解释执行升级指令序列中的升级指令时,可用于根据循环指令循环执行多条升级指令,也可用于根据分支指令在执行过程中进行条件判断,选择与条件判断结果相符的升级指令解释执行。

[0081] 在升级脚本中添加了顺序控制指令后可以对升级指令的执行顺序进行更加详细的限定。例如,在执行升级指令时,升级模块 106 可用于通过执行分支指令对应用平台 100 的运行环境进行判断,然后再选择与应用平台的运行环境相符合的升级指令执行。从而使得升级指令的执行不再限于顺序执行顺序,使得应用升级方法更加灵活。

[0082] 上述应用升级方法和系统,脚本服务器根据应用平台获取并上传的应用信息查找与应用信息对应的预设的升级脚本,并下发给应用平台,应用平台根据升级脚本进行升级。由于升级脚本与应用信息对应,即预先可根据应用的类型设置不同的升级脚本,在升级时获取应用对应的升级脚本实现升级即可,从而提高了应用升级方法和系统的灵活性。

[0083] 以上所述实施例仅表达了本发明的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对本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,在不脱离本发明构思的前提下,还可以做出若干变形和改进,这些都属于本发明的保护范围。因此,本发明的保护范围应以所附权利要求为准。

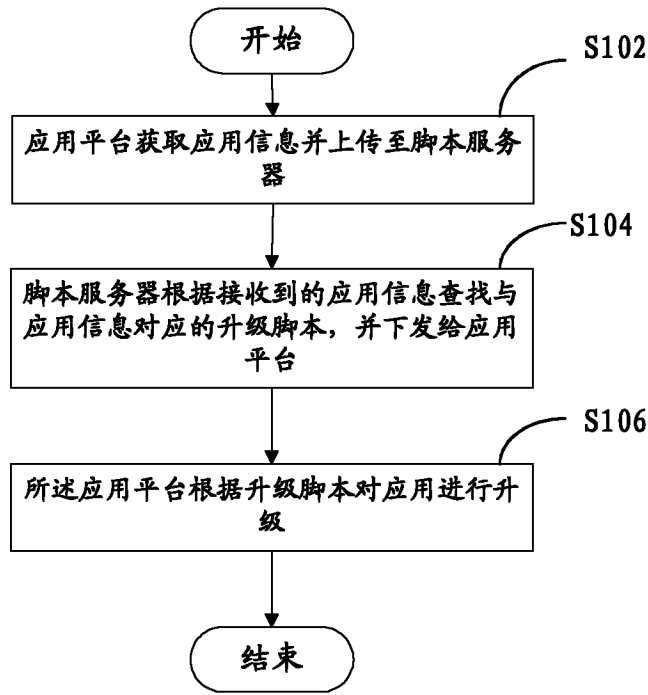


图 1

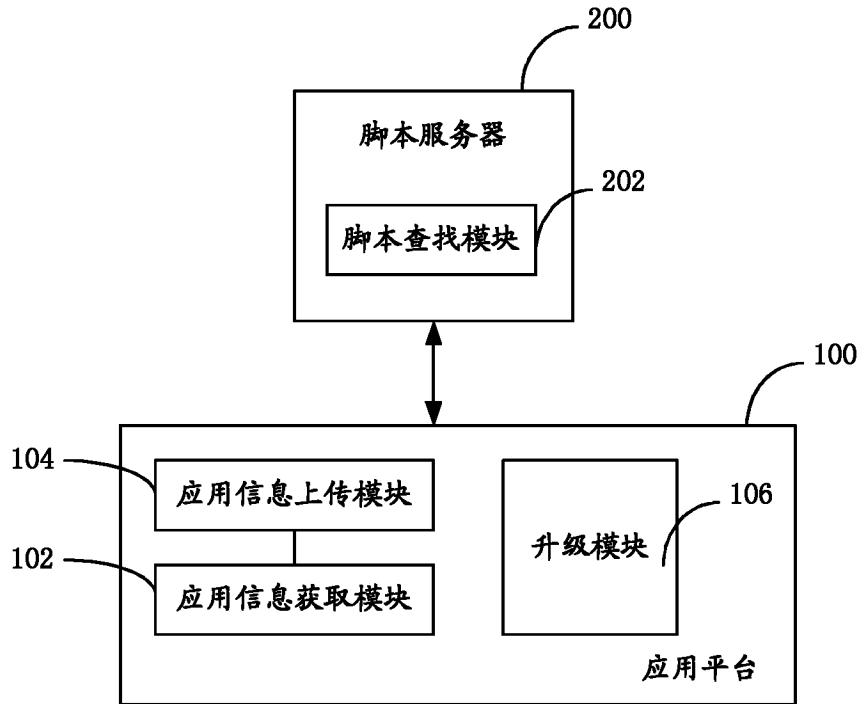


图 2

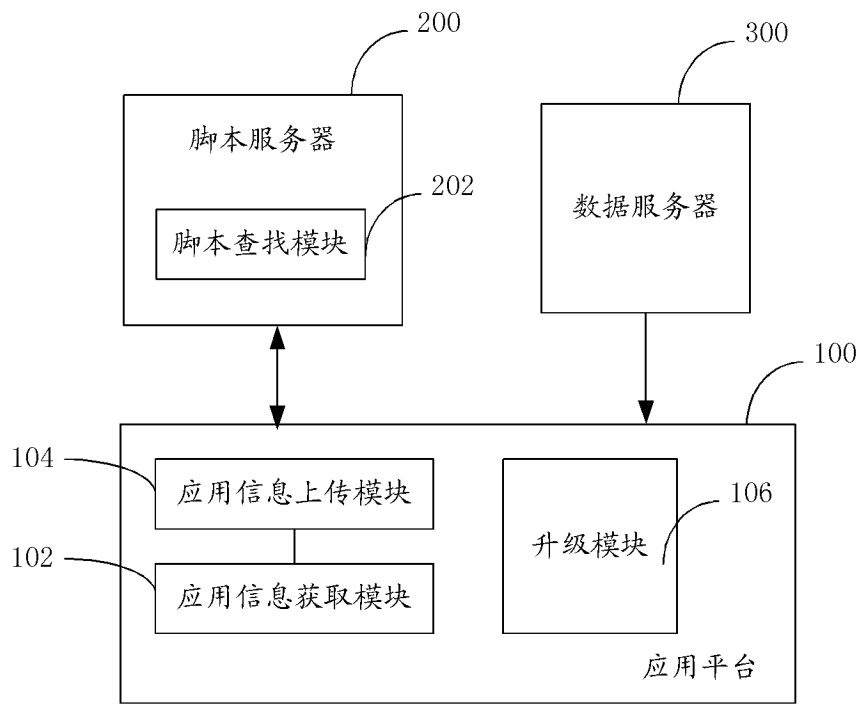


图 3