



(12)发明专利申请

(10)申请公布号 CN 107682431 A

(43)申请公布日 2018.02.09

(21)申请号 201710895377.1

(22)申请日 2017.09.28

(71)申请人 成都路行通信息技术有限公司
地址 610041 四川省成都市高新区天府大道中段530号东方希望天祥广场一幢1402

(72)发明人 陈剑波 汪亮

(51)Int.Cl.
H04L 29/08(2006.01)

权利要求书1页 说明书6页 附图2页

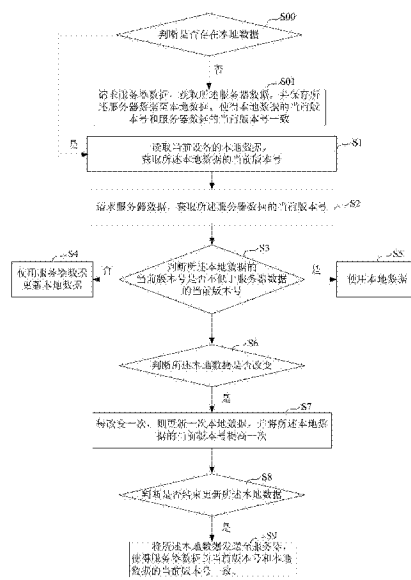
(54)发明名称

一种多设备间数据同步优化方法及系统

(57)摘要

本发明公开一种多设备间数据同步优化方法,包括读取当前设备的本地数据,获取其当前版本号。请求服务器数据,获取其当前版本号。判断两个版本号高低,使用较高版本的数据。判断本地数据是否改变,如果是,则每改变一次,只更新一次本地数据,并将所述本地数据的当前版本号提高一次;更新本地数据后,用户即可看到最新修改后的界面。且因数据对实时性要求不高,当结束更新才上传最高版本号的本地数据至服务器也可。因此,即使多次修改,也无需频繁访问服务器,减轻服务器的压力,降低因网络中断造成数据丢失的概率。而且,在多个设备之间切换时,如果在前一设备上做了修改,在更换设备后可以通过访问服务器将数据同步到本设备的本地数据上。

CN 107682431 A



1. 一种多设备间数据同步优化方法,其特征在于,应用于设备端,所述方法包括:
读取当前设备的本地数据,获取所述本地数据的当前版本号;
请求服务器数据,获取所述服务器数据的当前版本号;
判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号,如果否,则使用服务器数据更新本地数据,如果是,则使用本地数据;
判断所述本地数据是否改变,如果是,则每改变一次,则更新一次本地数据,并将所述本地数据的当前版本号提高一次;
判断是否结束更新所述本地数据,如果是,则将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。
2. 根据权利要求1所述的多设备间数据同步优化方法,其特征在于,所述将所述本地数据的当前版本号提高一次具体指,将本地数据的当前版本号数值加一。
3. 根据权利要求1所述的多设备间数据同步优化方法,其特征在于,在所述读取当前设备的本地数据,获取所述本地数据的当前版本号之前还包括:
判断是否存在本地数据,如果不存在,则请求服务器数据,获取所述服务器数据,并保存所述服务器数据至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。
4. 一种多设备间数据同步优化系统,其特征在于,包括:
本地数据获取模块:用于读取当前设备的本地数据,获取所述本地数据的当前版本号;
服务器数据获取模块:用于请求服务器数据,获取所述服务器数据的当前版本号;
版本号差值判断模块,要不判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号;
本地数据更新模块:用于当服务器数据的当前版本号不低于本地数据的当前版本号时,使用服务器数据更新本地数据;
本地数据使用模块,用于当服务器数据的当前版本号不低于本地数据的当前版本号时,读取本地数据;
本地数据改变判断模块,用于判断所述本地数据是否改变;
本地数据更新模块:用于当本地数据改变时,更新本地数据,并根据更新次数提高本地数据的当前版本号;
本地数据更新结束判断模块:判断是否结束更新所述本地数据;
服务器数据更新模块,用于当所述本地数据更新结束时,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。
5. 根据权利要求4所述的一种多设备间数据同步优化系统,其特征在于,还包括:
本地数据储存模块:用于设备上不存在时,将服务器数据储存至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。

一种多设备间数据同步优化方法及系统

技术领域

[0001] 本发明涉及通信技术领域,具体涉及一种多设备间数据同步优化方法及系统。

背景技术

[0002] APP中有很多功能,但是出于操作方便和界面简洁的原因,在APP首首页上一般只放置有限个数的功能入口。例如,支付宝首页可以有的功能入口包括充值中心,收钱,我的快递,蚂蚁森林、转账,蚂蚁借呗和/或蚂蚁森林等。用户访问进入某个功能入口的过程,是向服务器请求返回数据的过程,按照目前的技术手段,每进入一次,需要服务器响应一次。用户操作过快,会服务器压力较大,同时电量和流量都耗费较多,如果网络较差,会出现界面刷新不及时,体验较差。

[0003] 根据用户的个性化需求,可以在APP首页上选择添加或删除某些功能入口,或者在功能入口内更改个人数据,例如,将常用的“我的快递”添加,删除不常用的“蚂蚁借呗”,并把“我的快递”调整到首页的第一个入口位置,然后在“我的快递”中添加了新的收货地址,一共4个操作。这些操作均为修改操作,修改的过程是向服务器上传数据,并通过服务器响应后更新返回新数据并的过程,按照目前的技术手段,每修改一次,需要服务器影响一下。在网络较差或异常中断时,可能会造成某些数据丢失,后果下次登陆APP后展现的不是修改后的最新的。

[0004] 可见,按照目前的技术手段,数据的获取和更新都依赖服务器完成,在同一个设备上登陆APP或在不同设备之间交替登陆APP,均可能造成体验不好或下次登陆时不是最新界面的后果。同一设备上数据优化,和多设备间数据同步优化,均是需要解决的问题,亟需一种新的多设备间数据同步优化方法。

发明内容

[0005] 本发明提供了一种多设备间数据同步优化方法和系统,以解决现有的数据在多个设备之间同步不及时或错误的问题。

[0006] 第一方面,一种多设备间数据同步优化方法,应用于设备端,所述方法包括:

[0007] 读取当前设备的本地数据,获取所述本地数据的当前版本号;

[0008] 请求服务器数据,获取所述服务器数据的当前版本号;

[0009] 判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号,如果否,则使用服务器数据更新本地数据,如果是,则使用本地数据;

[0010] 判断所述本地数据是否改变,如果是,则每改变一次,则更新一次本地数据,并将所述本地数据的当前版本号提高一次;

[0011] 判断是否结束更新所述本地数据,如果是,则将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。

[0012] 优选的,在上述多设备间数据同步优化方法中,所述将所述本地数据的当前版本号提高一次具体指,将本地数据的当前版本号数值加一。

[0013] 优选的,在上述多设备间数据同步优化方法中,,在所述读取当前设备的本地数据,获取所述本地数据的当前版本号之前还包括:

[0014] 判断是否存在本地数据,如果不存在,则请求服务器数据,获取所述服务器数据,并保存所述服务器数据至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。

[0015] 第二方面,本发明提供了一种多设备间数据同步优化系统,包括:

[0016] 本地数据获取模块:用于读取当前设备的本地数据,获取所述本地数据的当前版本号;

[0017] 服务器数据获取模块:用于请求服务器数据,获取所述服务器数据的当前版本号;

[0018] 版本号差值判断模块,要不判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号;

[0019] 本地数据更新模块:用于当服务器数据的当前版本号不低于本地数据的当前版本号时,使用服务器数据更新本地数据;

[0020] 本地数据使用模块,用于当服务器数据的当前版本号不低于本地数据的当前版本号时,读取本地数据;

[0021] 本地数据改变判断模块,用于判断所述本地数据是否改变;

[0022] 本地数据更新模块:用于当本地数据改变时,更新本地数据,并根据更新次数提高本地数据的当前版本号;

[0023] 本地数据更新结束判断模块:判断是否结束更新所述本地数据;

[0024] 服务器数据更新模块,用于当所述本地数据更新结束时,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。

[0025] 优选的,在上述多设备间数据同步优化系统中,还包括:

[0026] 本地数据储存模块:用于设备上不存在时,将服务器数据储存至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。

[0027] 可见,当在同一设备上,登陆APP时,本发明提供的方法首先使用本地数据来展示页面,同时向服务器请求数据,请求到服务器数据后与本地数据对比,对比的是两者的版本号,如果不一致,以版本号高的数据为准重新刷新页面,这样,如果用户不做修改操作时,只需在该设备上第一次登陆APP时,才需要访问服务器,能够避免多次访问服务器,减轻服务器的压力,降低因网络中断造成数据丢失的概率。而且只要在设备上登陆过APP,就会形成本地数据。在多个设备之间切换登陆时,均会在本设备上形成本地数据。

[0028] 当进行修改操作时,每次修改,只更新本地数据,并升高其版本号,而且更新本地数据后,用户即可看到最新修改后的界面。因为数据对实时性要求不高,所以最后在用户退出App的时候才上传修改后的最高版本号的本地数据至服务器,这样即使多次修改,也无需频繁访问服务器并上传数据,减轻服务器的压力,降低因网络中断造成数据丢失的概率。而且,在多个设备之间切换时,如果在前一台设备上做了修改,在更换设备后可以通过访问服务器将数据同步到本设备的本地数据上。

附图说明

[0029] 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,显而易见地,对于本领域普通技术人员而言,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0030] 图1为本发明提供多设备间数据同步优化方法的流程示意图;

[0031] 图2为本发明提供多设备间数据同步优化系统的结构示意图。

具体实施方式

[0032] 本发明提供一种多设备间数据同步优化方法和系统,以解决现有的数据在多个设备之间同步不及时或错误的问题。以下以在设备A(华为手机)和设备B(苹果手机)上,登陆支付宝APP时,应用本发明提供的方法同步数据的过程为例,以实施例的方式进行说明。

[0033] 实施例一

[0034] 第一次在在设备A上登陆支付宝APP,预进行的操作为查看余额—查看本月账单——退出。

[0035] 按照本发明提供的方法,在步骤S1中,读取当前设备的本地数据,获取本地数据的当前版本号。在步骤S2中,请求服务器数据,获取服务器数据的当前版本号。由于是第一次登陆,结果是不存在本地数据。因此,直接从服务器下载数据,获取服务器数据的当前版本号为1。

[0036] 然后,在步骤S3中,判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号,如果否,则在步骤S4中,使用服务器数据更新本地数据,如果是,则在步骤S5中,使用本地数据。可见,在本实例中,当进行操作查看余额时,是从服务器下载数据后更新至本地数据,服务器当前本地数据的版本号为1。

[0037] 在步骤S6中,判断所述本地数据是否改变,如果是,则在步骤S7中,每改变一次,则更新一次本地数据,并将所述本地数据的当前版本号提高一次。因为操作为查看余额和查看本月账单,因此,本地数据并未改变,因此版本号不做改变,仍为1。而且,当在步骤S4后,当进行查看本月账单时,不需要访问服务器下载数据,可以直接使用本地数据刷新界面。

[0038] 然后,在步骤S8中,判断是否结束更新所述本地数据,如果是,则在步骤S9中,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。当查看本月账单的操作结束后,下个操作为退出APP,设备收到本地数据更新结束的命令,则将设备A上的本地数据发送至服务器,服务器形成版本号同样为1的服务器数据。可见,如果用户连续操作时,只需要访问本地数据即可,无需每次都从服务器拉取数据,减轻了服务器压力,也节省了用户的流量,在网络不佳时,也能提升用户体验。

[0039] 实施例二

[0040] 在实施例一一之后,再次在设备A上登陆支付宝APP,预进行的操作是设置扣款顺序——在首页上添加蚂蚁森林入口——在首页上删除淘票票入口——退出APP。

[0041] 按照本发明提供的方法,在步骤S1中,读取当前设备的本地数据,获取本地数据的当前版本号。在步骤S2中,请求服务器数据,获取服务器数据的当前版本号。由于是第二次登陆,存在本地数据。

[0042] 然后,在步骤S3中,判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号,如果否,则在步骤S4中,使用服务器数据更新本地数据,如果是,则在步骤S5中,使用本地数据。可见,在本实例中,首先读取本地数据,同时,获取服务器数据以及当前版本

号也为1。服务当前本地数据的版本号为1。并获得本地数据的当前版本号为1。两者的版本号相等,因此,当进行操作设置扣款顺序时,直接读取本地数据。

[0043] 在步骤S6中,判断所述本地数据是否改变,如果是,则在步骤S7中,每改变一次,则更新一次本地数据,并将所述本地数据的当前版本号提高一次。将本地数据的当前版本号提高一次可以是,将本地数据的当前版本号数值加一。本实例中的三个操作设置扣款顺序——在首页上添加蚂蚁森林入口——在首页上删除淘票票入口,均改变了本地数据。因此,当进行完这三个操作时,本地数据的版本号升高为4。

[0044] 然后,在步骤S8中,判断是否结束更新所述本地数据,如果是,则在步骤S9中,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。当在首页上删除淘票票入口的操作结束后,下个操作为退出APP,设备收到本地数据更新结束的命令,则将设备A上的本地数据发送至服务器,服务器形成版本号同样为4的服务器数据。而且,当在步骤S5后,当进行在首页上添加蚂蚁森林入口——在首页上删除淘票票入口操作时,不需要访问服务器下载数据,可以直接使用本地数据刷新界面。可见,如果用户连续操作时,只需要访问本地数据即可,无需每次都从服务器拉取数据,减轻了服务器压力,也节省了用户的流量,在网络不佳时,也能提升用户体验。

[0045] 为了更加优化数据获取的方法,在本实施例中的步骤S1之前,还可以包括步骤S00,判断是否存在本地数据,如果不存在,则在步骤S01中,请求服务器数据,获取所述服务器数据,并保存所述服务器数据至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。可见,本实施例中,步骤S00的判断结果是存在,并且进入步骤S1进行读取本地数据。

[0046] 实施例三

[0047] 在实施例一以后,更换设备B第一次登陆支付宝APP,预进行的操作是查看蚂蚁花呗——更换预留手机号——退出APP。

[0048] 按照本发明提供的方法,在步骤S00,判断是否存在本地数据,如果不存在,则在步骤S01中,请求服务器数据,获取所述服务器数据,并保存所述服务器数据至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。可见,本实施例中,步骤S00的判断结果是不存在,则在步骤S2中,请求服务器数据,获取服务器数据的当前版本号。可见,在本实例中,当进行操作查看查看蚂蚁花呗前,是从服务器下载数据后更新至本地数据,服务器当前本地数据的版本号为4,设备B的本地数据的当前版本数为4。

[0049] 然后,在步骤S3中,判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号,如果否,则在步骤S4中,使用服务器数据更新本地数据,如果是,则在步骤S5中,使用本地数据。因为本发明提供的多设备间数据同步优化方法,无论是在任一设备上,修改APP的本地数据后均在退出APP时,将最新的修改上传至服务器,所以,即使更换设备后,第一次登陆此APP,也能看到最新的修改界面,不会因为更换设备而造成数据丢失。

[0050] 在步骤S6中,判断所述本地数据是否改变,如果是,则在步骤S7中,每改变一次,则更新一次本地数据,并将所述本地数据的当前版本号提高一次。当进行操作查看蚂蚁花呗时,直接读取本地数据。换预留手机号,改变了本地数据。因此,本地数据的版本号升高为5。

[0051] 然后,在步骤S8中,判断是否结束更新所述本地数据,如果是,则在步骤S9中,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。

当操作为退出APP,设备收到本地数据更新结束的命令,则将设备B上的本地数据发送至服务器,服务器形成版本号同样为5的服务器数据。

[0052] 从上述实施例一到三可知,当在同一设备上,登陆APP时,本发明提供的方法首先使用本地数据来展示页面,同时向服务器请求数据,请求到服务器数据后与本地数据对比,对比的是两者的版本号,如果不一致,以版本号高的数据为准重新刷新页面,这样,如果用户不做修改操作时,只需在该设备上第一次登陆APP时,才需要访问服务器,能够避免多次访问服务器,减轻服务器的压力,降低因网络中断造成数据丢失的概率。而且只要在设备上登陆过APP,就会形成本地数据。在多个设备之间切换登陆时,均会在本设备上形成本地数据。

[0053] 当进行修改操作时,每次修改,只更新本地数据,并升高其版本号,而且更新本地数据后,用户即可看到最新修改后的界面。因为数据对实时性要求不高,所以最后在用户退出App的时候才上传修改后的最高版本号的本地数据至服务器,这样即使多次修改,也无需频繁访问服务器并上传数据,减轻服务器的压力,降低因网络中断造成数据丢失的概率。而且,在多个设备之间切换时,如果在前一台设备上做了修改,在更换设备后可以通过访问服务器将数据同步到本设备的本地数据上。

[0054] 本发明还提供了与上述方法相对应的一种多设备间数据同步优化系统,请参考图1包括:

[0055] 本地数据获取模块:用于读取当前设备的本地数据,获取所述本地数据的当前版本号;

[0056] 服务器数据获取模块:用于请求服务器数据,获取所述服务器数据的当前版本号;

[0057] 版本号差值判断模块:要不判断所述本地数据的当前版本号是否不低于服务器数据的当前版本号;

[0058] 本地数据更新模块:用于当服务器数据的当前版本号不低于本地数据的当前版本号时,使用服务器数据更新本地数据;

[0059] 本地数据使用模块:用于当服务器数据的当前版本号不低于本地数据的当前版本号时,读取本地数据;

[0060] 本地数据改变判断模块:用于判断所述本地数据是否改变;

[0061] 本地数据更新模块:用于当本地数据改变时,更新本地数据,并根据更新次数提高本地数据的当前版本号;

[0062] 本地数据更新结束判断模块:判断是否结束更新所述本地数据;

[0063] 服务器数据更新模块:用于当所述本地数据更新结束时,将所述本地数据发送至服务器,使得服务器数据的当前版本号和本地数据的当前版本号一致。

[0064] 为了进一步提高优化效率,上述系统还可以包括本地数据储存模块:用于设备上不存在时,将服务器数据储存至本地数据,使得本地数据的当前版本号和服务器数据的当前版本号一致。

[0065] 上述系统的工作过程可以参考方法技术方案,此不赘述。

[0066] 以上所述仅是本发明的具体实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视为本发明的保护范围。

[0067] 本说明书中各个实施例之间相同相似的部分互相参见即可。尤其,对于……实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例中的说明即可。

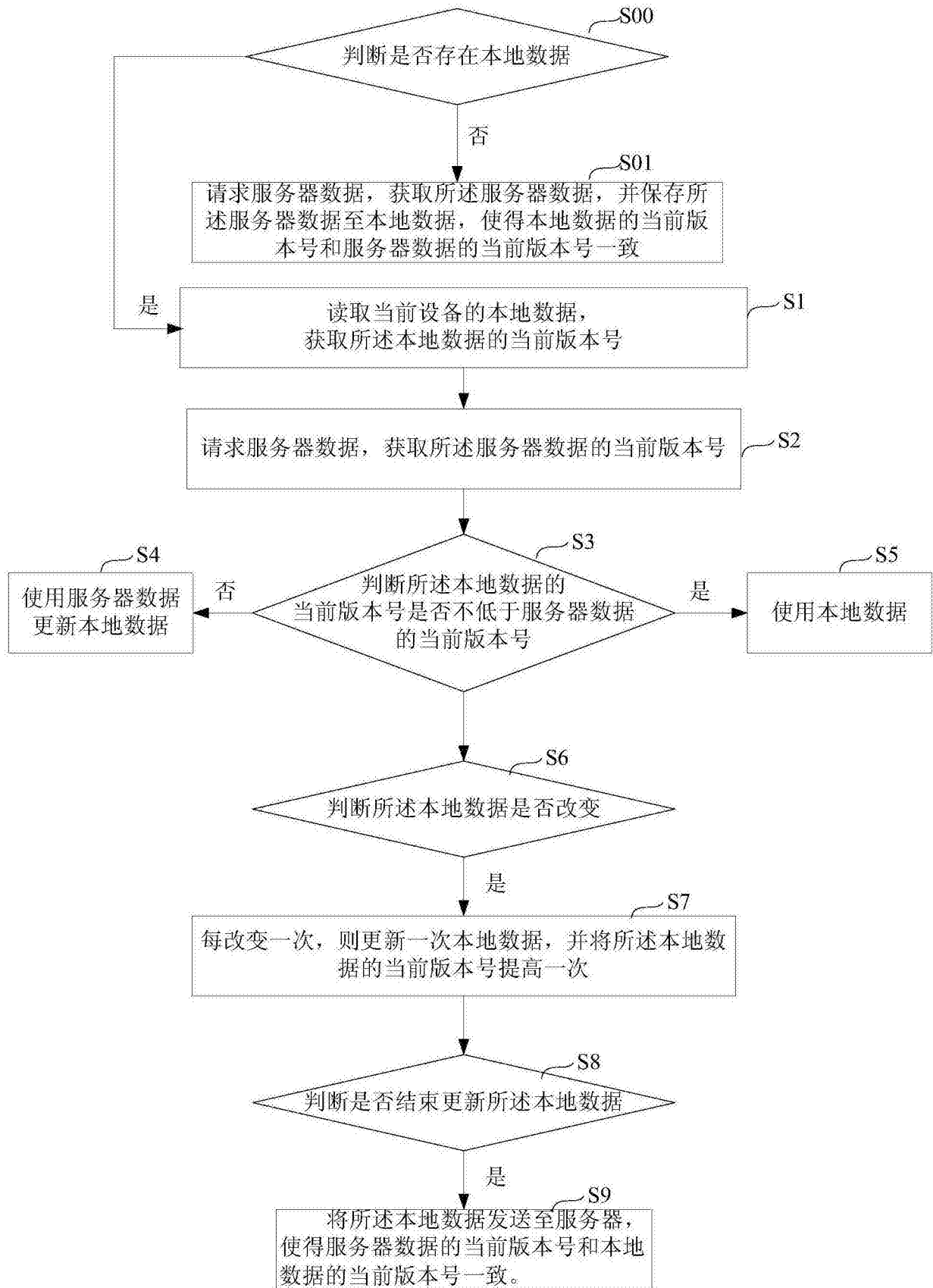


图1

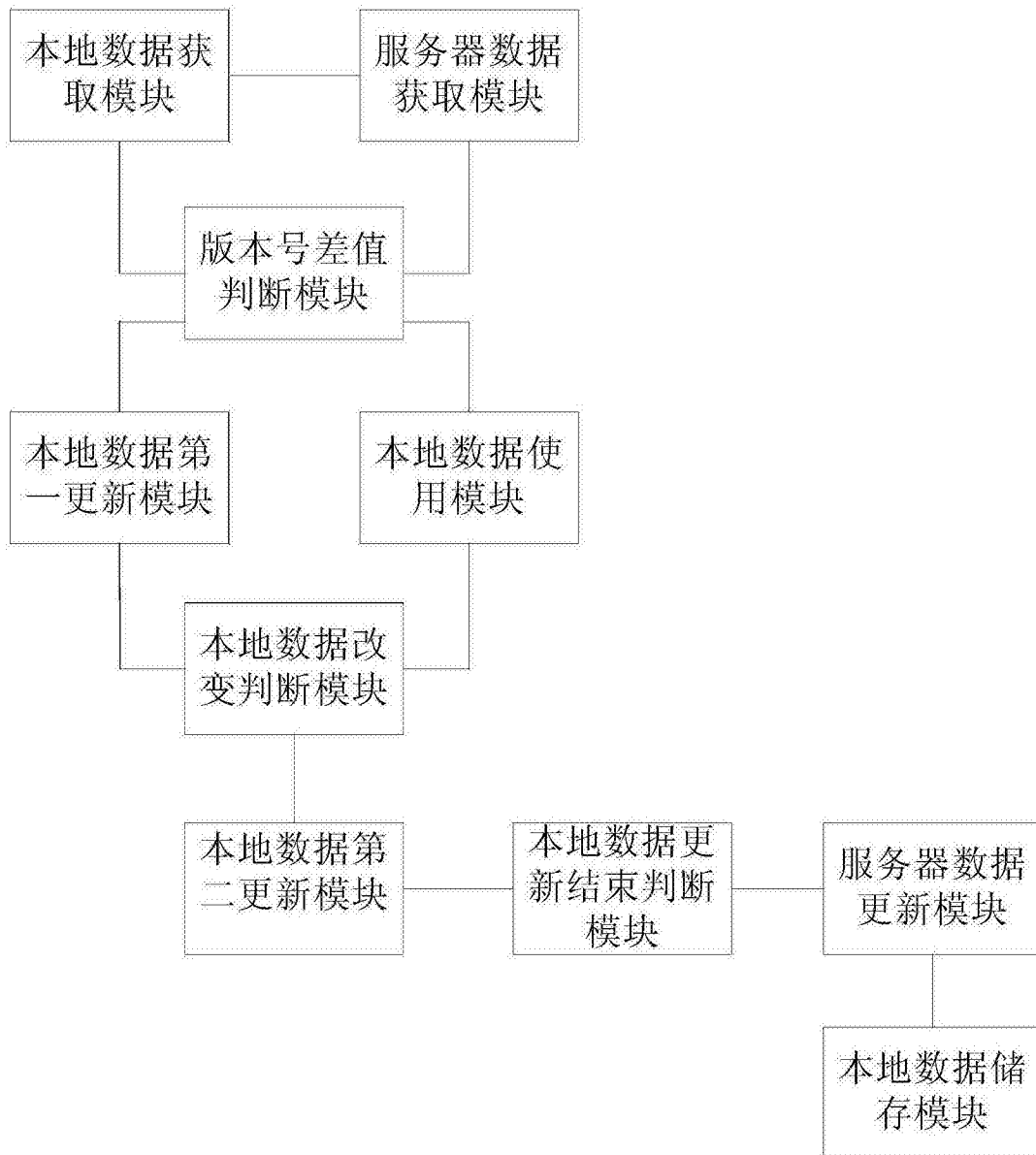


图2