



(12) 发明专利申请

(10) 申请公布号 CN 102043853 A

(43) 申请公布日 2011. 05. 04

(21) 申请号 201010601118. 1

(22) 申请日 2010. 12. 22

(71) 申请人 中国农业银行股份有限公司  
地址 100005 北京市东城区建国门内大街  
69 号

(72) 发明人 薛俊鹏 吴会 张宝强 李艳成  
孙洪江 张硕

(74) 专利代理机构 北京集佳知识产权代理有限  
公司 11227  
代理人 逯长明 王宝筠

(51) Int. Cl.  
G06F 17/30 (2006. 01)

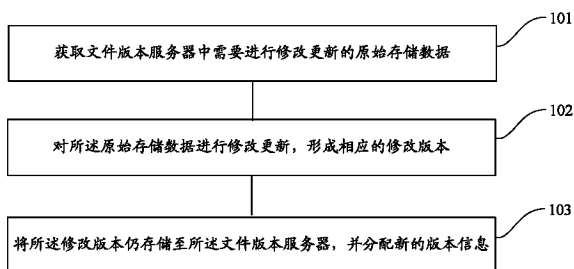
权利要求书 1 页 说明书 7 页 附图 4 页

(54) 发明名称

一种存储数据的更新方法和装置

(57) 摘要

本发明实施例公开了一种存储数据的更新方法和装置,其中,所述方法包括:获取文件版本服务器中需要进行修改更新的原始存储数据;对所述原始存储数据进行修改更新,形成相应的修改版本;将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。通过本发明,便于实现对数据库中数据的修改,当数据修改过程中发现错误时,便于查找错误的出处和原因,使得数据库中配置数据修改更新的效率大大提高。



1. 一种存储数据的更新方法,其特征在于,所述方法包括:  
获取文件版本服务器中需要进行修改更新的原始存储数据;  
对所述原始存储数据进行修改更新,形成相应的修改版本;  
将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。
2. 根据权利要求1所述的存储数据的更新方法,其特征在于,所述获取需要进行修改更新的原始存储数据,具体包括:  
获取数据库中未经过修改更新的初始存储数据;或者,获取所述文件版本服务器中已有的完成修改更新的数据版本。
3. 根据权利要求2所述的存储数据的更新方法,其特征在于,当所述原始存储数据为文件版本服务器中已有的完成修改更新的数据版本时,对所述原始存储数据进行修改更新具体包括:  
在所述数据版本已有的修改操作基础之上,对所述数据版本继续执行新的修改操作。
4. 根据权利要求1所述的存储数据的更新方法,其特征在于,所述方法还包括:  
将所述需要进行修改更新的原始存储数据存储至本地文件系统;  
则对所述原始存储数据进行修改更新具体为:在所述本地文件系统对所述原始数据进行修改更新。
5. 根据权利要求1~4中任一项所述的存储数据的更新方法,其特征在于,所述方法还包括:  
利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。
6. 一种存储数据的更新装置,其特征在于,所述装置包括:  
原始数据获取模块,用于获取需要进行修改更新的原始存储数据;  
数据更新模块,用于对所述原始存储数据进行修改更新,形成相应的修改版本;  
更新存储模块,用于将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。
7. 根据权利要求6所述的存储数据的更新装置,其特征在于,所述原始存储数据具体包括:  
数据库中未经过修改更新的初始存储数据;或者,所述文件版本服务器中已有的完成修改更新的数据版本。
8. 根据权利要求7所述的存储数据的更新装置,其特征在于,当所述原始存储数据为文件版本服务器中已有的完成修改更新的数据版本时,所述数据更新模块用于:在所述数据版本已有的修改操作基础之上,对所述数据版本继续执行新的修改操作。
9. 根据权利要求6所述的存储数据的更新装置,其特征在于,所述装置还包括:  
转发模块,用于将所述需要进行修改更新的原始存储数据存储至本地文件系统;  
则数据更新模块用于在所述本地文件系统对所述原始数据进行修改更新。
10. 根据权利要求6~8中任一项所述的存储数据的更新装置,其特征在于,所述装置还包括:  
库更新模块,用于利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。

## 一种存储数据的更新方法和装置

### 技术领域

[0001] 本发明涉及计算机技术领域,更具体地说,涉及一种存储数据的更新方法和装置。

### 背景技术

[0002] 随着计算机技术的发展,数据库在各种生产环境中的应用越来越为广泛,用于存储各种应用数据。

[0003] 通常,可以将各种应用程序划分为后台应用程序和前端应用程序。后台应用程序所需要的配置数据,一般存储在关系数据库中。实际应用中,往往需要对存储在关系数据库中的配置数据进行开发和维护,该过程中需要对相应配置数据进行修改更新。

[0004] 现有技术中,实现对后台关系数据库中的配置数据的修改更新通常采用如下方式:利用前端应用程序直接对关系数据库的数据进行修改,相应地采用两种前端应用程序,一种是利用关系数据库厂商提供的客户端工具,比如 Oracle Client、Sybase Sql Advantage 等;另一种是由操作人员实际开发新的前端应用程序,利用 ado.net、jdbc 等技术访问数据库,并提供对于数据的图形化操作界面。

[0005] 通过发明人的研究发现,上述现有技术中的配置数据的修改更新操作至少具有以下缺陷:

[0006] 当直接采用关系数据库厂商提供的客户端工具时,要求开发人员熟练掌握各种不同数据库的工具使用;精通 sql 语句的编写;熟悉数据库中的每一个表结构,可见,该种方式只适用于数据库表比较少,或者少量数据的增删改查;

[0007] 而采用由操作人员实际开发新的前端应用程序时,由于开发过程中,数据通常是需反复修改的,而现有技术中的关系数据库由于版本问题,数据库表中只能保存一份数据记录,不能保留同一份数据的多组不同历史版本,因此,修改过程中每次修改后形成的数据不能保留在数据库中,只能保留最后一次修改后的数据,这样,当如果发现最终修改的数据有误时,将由于无法判断错误发生在修改过程中的哪次修改,而导致对配置数据的修改更新操作效率低下。

### 发明内容

[0008] 有鉴于此,本发明实施例提供一种存储数据的更新方法和装置,以提高数据库中配置数据修改更新的效率。

[0009] 本发明实施例提供一种存储数据的更新方法,所述方法包括:

[0010] 获取文件版本服务器中需要进行修改更新的原始存储数据;

[0011] 对所述原始存储数据进行修改更新,形成相应的修改版本;

[0012] 将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。

[0013] 优选的,所述获取需要进行修改更新的原始存储数据,具体包括:

[0014] 获取数据库中未经过修改更新的初始存储数据;或者,获取所述文件版本服务器中已有的完成修改更新的数据版本。

[0015] 优选的,当所述原始存储数据为文件版本服务器中已有的完成修改更新的数据版本时,对所述原始存储数据进行修改更新具体包括:

[0016] 在所述数据版本已有的修改操作基础之上,对所述数据版本继续执行新的修改操作。

[0017] 优选的,所述方法还包括:

[0018] 将所述需要进行修改更新的原始存储数据存储至本地文件系统;

[0019] 则对所述原始存储数据进行修改更新具体为:在所述本地文件系统对所述原始数据进行修改更新。

[0020] 优选的,所述方法还包括:

[0021] 利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。

[0022] 一种存储数据的更新装置,所述装置包括:

[0023] 原始数据获取模块,用于获取需要进行修改更新的原始存储数据;

[0024] 数据更新模块,用于对所述原始存储数据进行修改更新,形成相应的修改版本;

[0025] 更新存储模块,用于将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。

[0026] 优选的,所述原始存储数据具体包括:

[0027] 数据库中未经过修改更新的初始存储数据;或者,所述文件版本服务器中已有的完成修改更新的数据版本。

[0028] 优选的,当所述原始存储数据为文件版本服务器中已有的完成修改更新的数据版本时,所述数据更新模块用于:在所述数据版本已有的修改操作基础之上,对所述数据版本继续执行新的修改操作。

[0029] 优选的,所述装置还包括:

[0030] 转发模块,用于将所述需要进行修改更新的原始存储数据存储至本地文件系统;

[0031] 则数据更新模块用于在所述本地文件系统对所述原始数据进行修改更新。

[0032] 优选的,所述装置还包括:

[0033] 库更新模块,用于利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。

[0034] 同现有技术相比,本发明实施例提供的技术方案通过在前端应用程序和数据库之间开发设置文件版本服务器,利用该文件版本服务器存储开发过程中对数据库中相关配置数据的修改版本,在后续修改时直接调用文件版本服务器中相应的修改版本,从而便于实现对数据库中数据的修改,当数据修改过程中发现错误时,便于查找错误的出处和原因,使得数据库中配置数据修改更新的效率大大提高。

#### 附图说明

[0035] 为了更清楚地说明本发明实施例的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

- [0036] 图 1 为本发明实施例提供的一种存储数据的更新方法步骤示意图；
- [0037] 图 2 为本发明实施例提供的另一种存储数据的更新方法步骤示意图；
- [0038] 图 3 为本发明实施例提供的顺序开发模式示意图；
- [0039] 图 4 为本发明实施例提供的并行开发模式示意图；
- [0040] 图 5 为本发明实施例提供的一种存储数据的更新装置结构示意图；
- [0041] 图 6 为本发明实施例提供的另一种存储数据的更新装置结构示意图；
- [0042] 图 7 为本发明实施例提供的又一种存储数据的更新装置结构示意图。

### 具体实施方式

[0043] 下面将结合本发明实施例中的附图,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。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动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。

[0044] 本发明实施例提供一种存储数据的更新方法及装置,以提高数据库中配置数据修改更新的效率。为了便于对本发明实施例技术方案的充分理解,使得本发明的目的、技术方案及优点更加清楚明白,下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整的描述。

[0045] 本发明实施例提供了一种存储数据的更新方法,如图 1 所示,为该方法的步骤流程,该方法可以包括以下步骤:

[0046] 步骤 101、获取文件版本服务器中需要进行修改更新的原始存储数据;

[0047] 步骤 102、对所述原始存储数据进行修改更新,形成相应的修改版本;

[0048] 该步骤中,修改版本是针对原始存储数据执行修改更新操作之后,生成的某一数据版本。

[0049] 步骤 103、将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。

[0050] 该步骤中,对存入文件版本服务器中的每一修改版本都分配特定的版本信息,通过该版本信息对不同的修改版本进行标识,这是由于本发明实施例中,文件版本服务器中可能同时存在多个针对某一原始存储数据生成的修改版本。

[0051] 本发明实施例提供的技术方案通过在前端应用程序和数据库之间开发设置文件版本服务器,利用该文件版本服务器存储开发过程中对数据库中相关配置数据的修改版本,在后续修改时直接调用文件版本服务器中相应的修改版本,从而便于实现对数据库中数据的修改,当数据修改过程中发现错误时,便于查找错误的出处和原因,使得数据库中配置数据修改更新的效率大大提高。

[0052] 在具体实施时,本发明实施例中所述的需要进行修改更新的原始存储数据,具体可以包括以下几种形式:

[0053] (1) 数据库中未经过修改更新的初始存储数据;

[0054] (2) 文件版本服务器中已有的完成修改更新的数据版本;

[0055] (3) 由本地提供的数据版本。

[0056] 对于第一种情形,所述初始存储数据是数据库中存储的从未经过修改的初始配置数据,当需要对这部分数据进行修改更新时,由于不能直接在数据库中执行操作,因此,需

要将这部分数据另外存储在文件版本服务器中,在文件版本服务器中对这部分数据进行修改更新;

[0057] 对于第二种情形,文件版本服务器中已有的完成修改更新的数据版本将作为当前修改更新的基础版本,因此,当前的修改更新是在该数据版本已有的修改操作基础之上进一步增加的新的修改操作。

[0058] 对于第三种情形,由本地提供的数据版本是原数据库中没有的配置数据,也不是根据文件版本服务器中已有的数据版本进行更新的版本,是开发人员在本地提供的全新配置数据。

[0059] 为了保证文件版本服务器中数据的稳定性,通常不允许直接在文件版本服务器中对相应数据执行修改更新操作。这种情形下,可以将所述需要进行修改更新的原始存储数据存储在本地文件系统,在所述本地文件系统对所述原始数据进行修改更新,更新完毕时,再将生成的相应数据版本发送至文件版本服务器中进行存储。

[0060] 所述文件系统以文件形式存储各种配置数据,通常位于每一个开发人员的磁盘上。

[0061] 为了执行对修改数据正确性的验证,或者,为了后台应用程序能够准确调用经过修改更新的数据版本,需要将相应的数据版本发布至数据库,以实现在数据库中运行相应的数据。因此,在本发明的另一个实施例中,如图 2 所示,所述存储数据的更新方法还可以包括以下步骤:

[0062] 步骤 104、利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。

[0063] 本发明实施例中,通过后台应用程序,读取数据库中的配置数据,运行相应的业务逻辑。

[0064] 数据库中仍然只保留有一份数据记录,因此,即使在开发过程中,针对数据库中某一初始数据同时存在多个修改版本,数据库中最终的配置数据仅为最终发布至数据库的完成修改更新的数据版本,其他的修改版本将存储在文件版本服务器中,不同的修改版本通过版本信息进行区分。

[0065] 需要说明的是,文件版本服务器中的修改版本以文件形式存储,当需要将某一修改版本发布至数据库中时,可以采用以下实现方式:

[0066] 从文件版本服务器中读取相应数据到内存,以实现利用文件 I/O 方法,读取文件中的内容;然后,将内存中的数据存储在数据库中,利用 jdbc 等数据库访问技术,对数据库表执行数据插入操作。

[0067] 在本发明实施例中,对于文件版本服务器中数据版本的管理,可以通过开源的版本管理工具 SVN(SubVersion)、商业化的版本管理工具 firefly 等实现,它们提供了相应的插件,供第三方应用程序集成。集成后的应用程序作为版本管理工具的客户端,可以与版本管理工具的服务器端(SVN 或者 firefly)通信,执行版本的上传、读取、合并等操作。

[0068] 本发明实施例仍利用前端应用程序调用文件版本服务器中存储的数据。因此,在开发过程中,需要将前端应用程序设置为文件版本服务器的客户端,从而,通过该“客户端”能够从文件版本服务器中检出相应存储数据的历史修改版本或者最新版本进行修改,并通过该“客户端”将修改后生成的修改版本长传至文件版本服务器;当利用文件系统执行数据

的修改时,需要在前端应用程序开发可视化的界面,通过该界面读写文件系统中文件的内容,并将文件系统中完成修改的数据上传至文件版本服务器中存储。

[0069] 针对某一原始存储数据执行修改操作时,通常可以采用顺序开发模式和并行开发模式,分别如图 3 和图 4 所示。

[0070] 如图 3 所示,开发人员 1 在文件版本服务器中获取需要进行修改更新的数据版本 1,在本地文件系统中对数据版本 1 进行修改编辑,形成文件 a,将文件 a 上传至文件版本服务器,形成数据版本 2;继而,由开发人员 2 获取数据版本 2,在本地文件系统中对数据版本 2 进行修改编辑,形成文件 a+,将文件 a+ 上传至文件版本服务器,形成数据版本 3。可见,采用顺序开发模式时,后续开发人员实际是在之前开发人员提供的修改版本基础之上进一步执行新的修改更新操作。

[0071] 如图 4 所示,采用并行开发模式时,开发人员 1 在文件版本服务器中获取需要进行修改更新的数据版本 1,在本地文件系统中对数据版本 1 进行修改编辑,形成文件 a,将文件 a 上传至文件版本服务器,形成数据版本 2;开发人员 2 在文件版本服务器中获取需要进行修改更新的数据版本 1,在本地文件系统中对数据版本 1 进行修改编辑,形成文件 a+;由于通常情况下,当文件版本服务器中形成数据版本 2 后,文件版本服务器会将该版本作为最新版本优先提供给开发人员,因此,开发人员 2 可以在文件版本服务器尚未生成数据版本 2 时获取数据版本 1。在开发人员 2 将当前的修改版本上传至文件版本服务器时,需要将已生成的数据版本 1 和数据版本 2 进行文件合并,形成文件 a++,并将文件 a++ 上传至文件版本服务器,形成数据版本 3。可见,采用并行开发模式时,后续开发人员需要整合之前开发人员提供的修改版本,以形成新的修改版本。

[0072] 可见,本发明实施例中,由于操作的是本地文件版本服务器中的数据文件,所以支持离线开发,开发过程不需要连接数据库,只有发布配置数据时才启动数据库联机。

[0073] 相应上述存储数据的更新方法实施例,本发明实施例还提供了一种存储数据的更新装置,如图 5 所示,所述装置包括:

[0074] 原始数据获取模块 501,用于获取需要进行修改更新的原始存储数据;

[0075] 数据更新模块 502,用于对所述原始存储数据进行修改更新,形成相应的修改版本;

[0076] 更新存储模块 503,用于将所述修改版本仍存储至所述文件版本服务器,并分配新的版本信息。

[0077] 其中,修改版本是针对原始存储数据执行修改更新操作之后,生成的某一数据版本。更新存储模块对存入文件版本服务器中的每一修改版本都分配特定的版本信息,通过该版本信息对不同的修改版本进行标识,这是由于本发明实施例中,文件版本服务器中可能同时存在多个针对某一原始存储数据生成的修改版本。

[0078] 本发明实施例通过在前端应用程序和数据库之间开发设置文件版本服务器,利用该文件版本服务器存储开发过程中对数据库中相关配置数据的修改版本,在后续修改时直接调用文件版本服务器中相应的修改版本,从而便于实现对数据库中数据的修改,当数据修改过程中发现错误时,便于查找错误的出处和原因,使得数据库中配置数据修改更新的效率大大提高。

[0079] 由于本发明实施例中所述的需要进行修改更新的原始存储数据,具体可以包括以

下两种形式：

[0080] (1) 数据库中未经过修改更新的初始存储数据；

[0081] (2) 文件版本服务器中已有的完成修改更新的数据版本。

[0082] 对于第一种情形,所述初始存储数据是数据库中存储的从未经过修改的初始配置数据,当需要对这部分数据进行修改更新时,由于不能直接在数据库中执行操作,因此,需要数据更新模块将这部分数据另外存储在文件版本服务器中,在文件版本服务器中对这部分数据进行修改更新；

[0083] 对于第二种情形,文件版本服务器中已有的完成修改更新的数据版本将作为当前修改更新的基础版本,因此,数据更新模块中所进行的当前修改更新是在该数据版本已有的修改操作基础之上进一步增加的新的修改操作。

[0084] 为了保证文件版本服务器中数据的稳定性,通常不允许直接在文件版本服务器中对相应数据执行修改更新操作。因此,在本发明的另一个实施例中,如图 6 所示,所述装置还可以包括：

[0085] 转发模块 504,用于将所述需要进行修改更新的原始存储数据存储至本地文件系统；

[0086] 则数据更新模块 502 用于在所述本地文件系统对所述原始数据进行修改更新。

[0087] 本发明实施例中,通过转发模块将所述需要进行修改更新的原始存储数据存储至本地文件系统,在所述本地文件系统对所述原始数据进行修改更新,更新完毕时,再将生成的相应数据版本发送至文件版本服务器中进行存储。

[0088] 所述文件系统以文件形式存储各种配置数据,通常位于每一个开发人员的磁盘上。

[0089] 为了执行对修改数据正确性的验证,或者,为了后台应用程序能够准确调用经过修改更新的数据版本,需要将相应的数据版本发布至数据库,以实现在数据库中运行相应的数据。因此,在本发明的另一个实施例中,如图 7 所示,所述存储数据的更新装置还可以包括：

[0090] 库更新模块 505,用于利用所述文件版本服务器中相应的某一修改版本,更新数据库中已有的数据信息。

[0091] 数据库中仍然只保留有一份数据记录,因此,即使在开发过程中,针对数据库中某一初始数据同时存在多个修改版本,数据库中最终的配置数据仅为最终发布至数据库的完成修改更新的数据版本,其他的修改版本将存储在文件版本服务器中,不同的修改版本通过版本信息进行区分。

[0092] 以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的单元可以是或者也可以不是物理上分开的,作为单元显示的部件可以是或者也可以不是物理单元,即可以位于一个地方,或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部模块来实现本实施例方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0093] 对所公开的实施例的上述说明,使本领域专业技术人员能够实现或使用本发明。对这些实施例的多种修改对本领域的专业技术人员来说将是显而易见的,本文中所定义的一般原理可以在不脱离本发明实施例的精神或范围的情况下,在其它实施例中实现。因此,



本发明实施例将不会被限制于本文所示的这些实施例,而是要符合与本文所公开的原理和新颖特点相一致的最宽的范围。

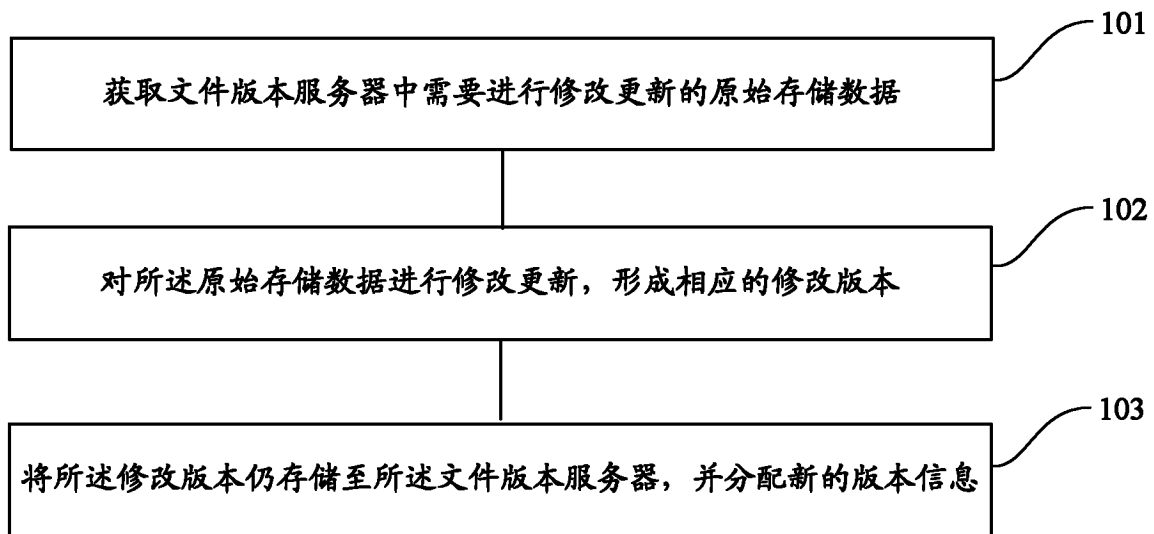


图 1

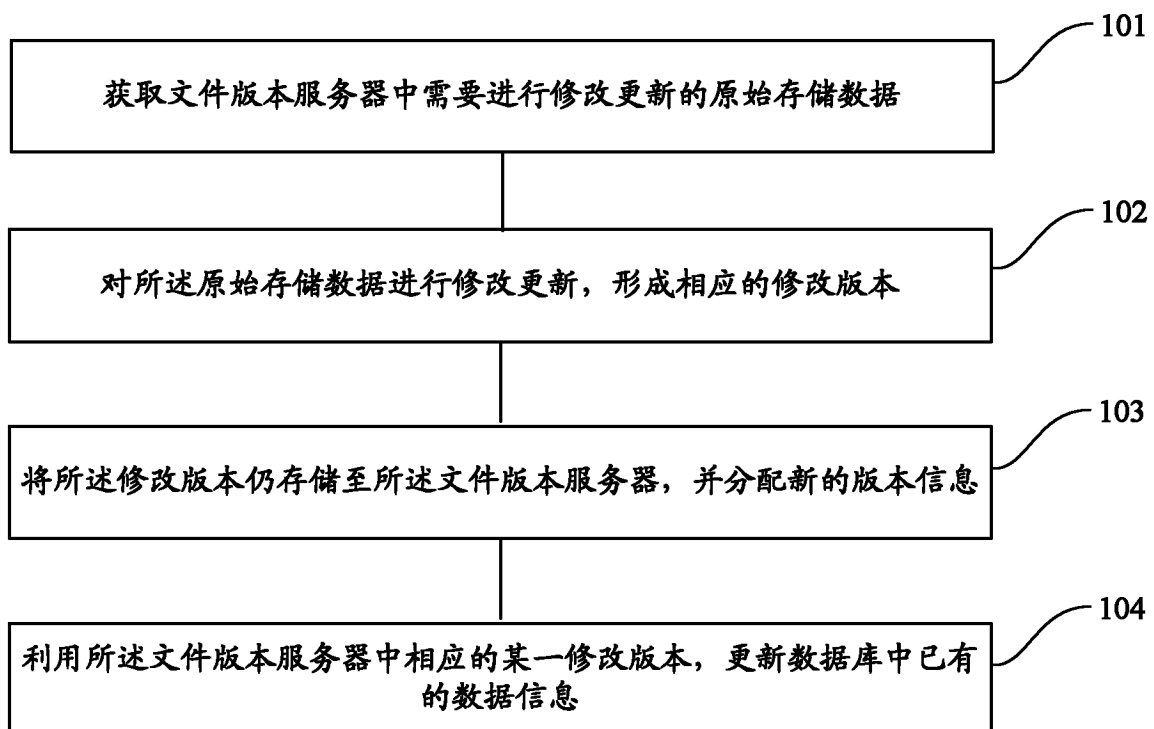


图 2

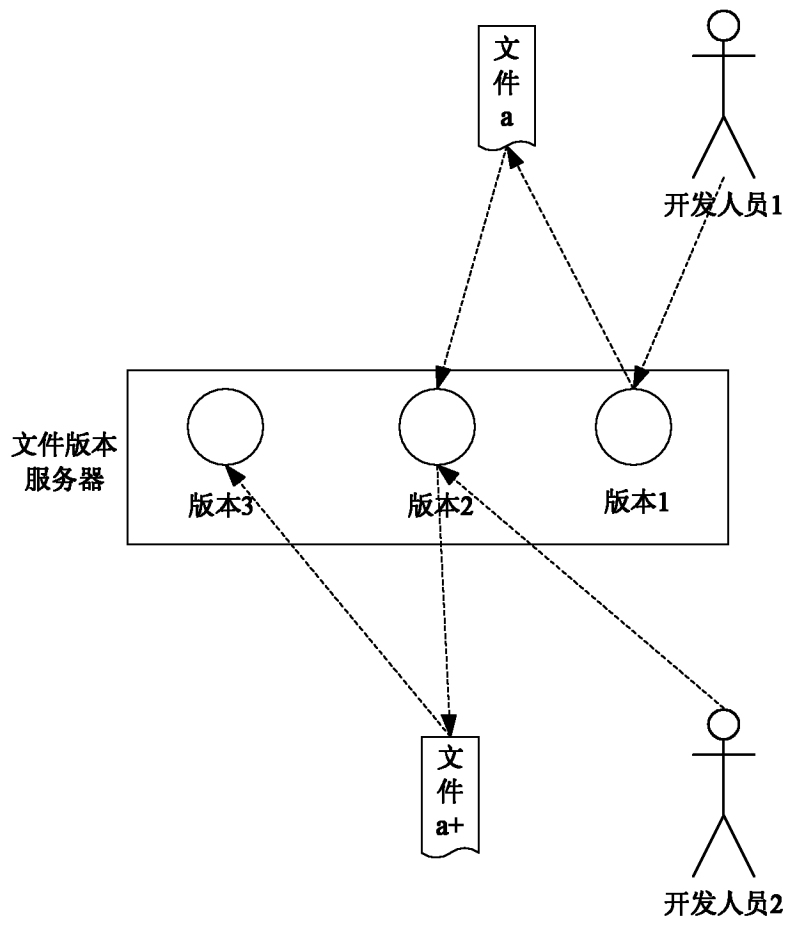


图 3

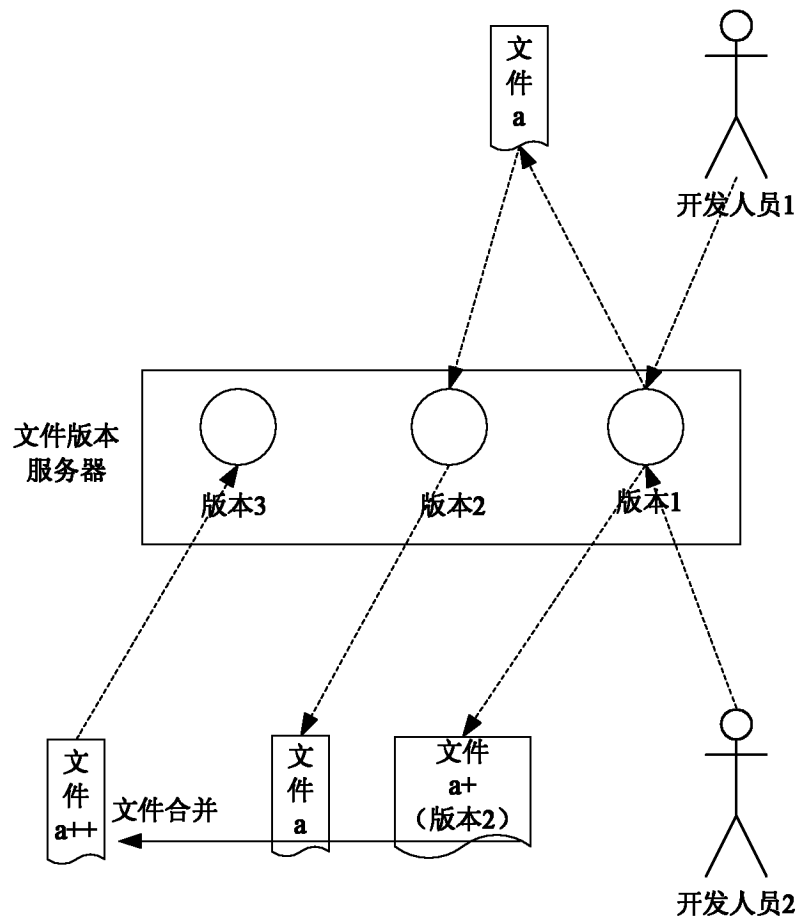


图 4

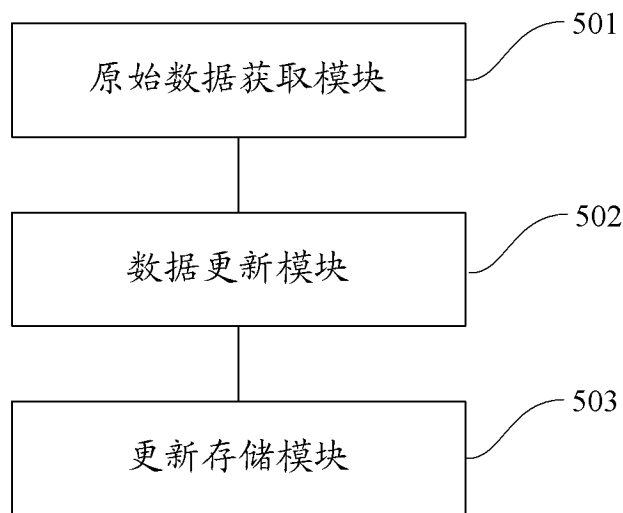


图 5

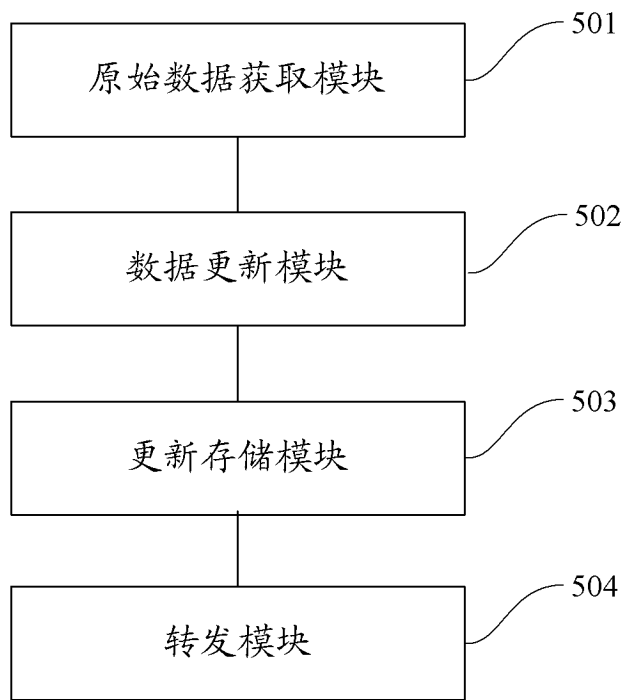


图 6

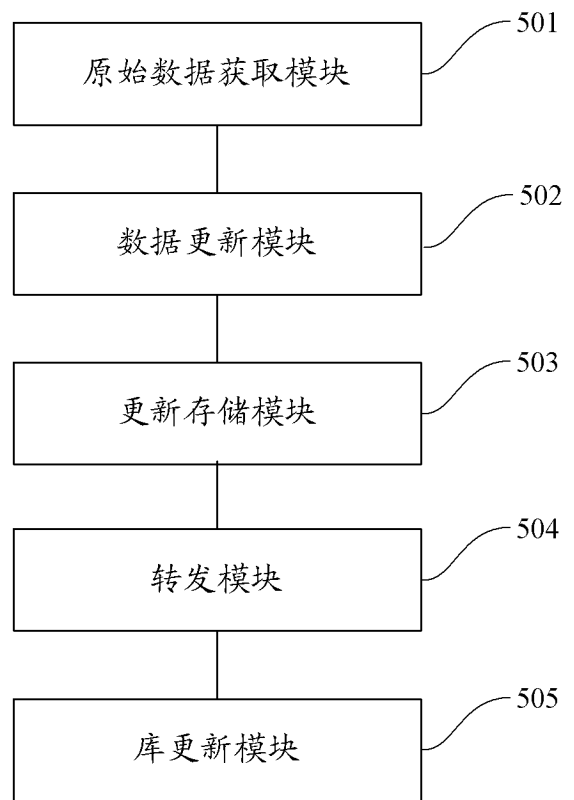


图 7