



(12) 发明专利申请

(10) 申请公布号 CN 104935634 A

(43) 申请公布日 2015. 09. 23

(21) 申请号 201510205983. 7

(22) 申请日 2015. 04. 27

(71) 申请人 南京大学

地址 210046 江苏省南京市栖霞区仙林大道
163 号

(72) 发明人 黄宇 张绍文

(74) 专利代理机构 南京苏高专利商标事务所
(普通合伙) 32204

代理人 李玉平

(51) Int. Cl.

H04L 29/08(2006. 01)

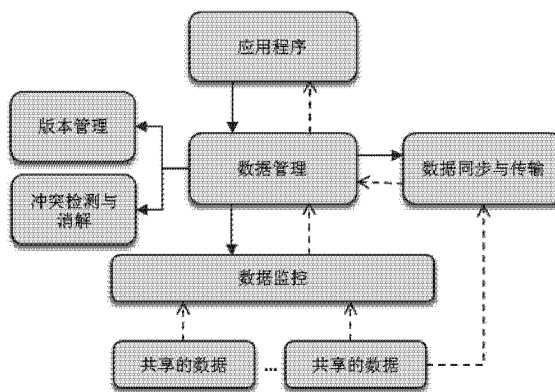
权利要求书2页 说明书7页 附图3页

(54) 发明名称

基于分布共享存储的移动设备数据共享方法

(57) 摘要

本发明公开了一种基于分布共享存储的移动设备数据共享平台的构建技术。首先在移动设备上设置共享的数据,数据监控进程对数据进行监控,记录数据变化信息;数据管理进程对保存的信息进行分析,利用版本控制模块更新数据的版本信息,并将版本信息发送给数据同步进程;数据同步进程依照设定的同步策略,将需要同步的数据,通过数据传输模块与其它移动设备进行同步;数据传输模块利用TCP技术进行数据传输;其它设备在接收到同步信息时,利用数据管理进程中的冲突检测与消解模块,对接收到的数据进行处理,最终达到数据的同步。本发明中的技术能够有效的在不同移动设备上构建起一个数据共享平台,并允许其它应用基于该平台进行快速开发。



1. 一种基于分布共享存储的移动设备数据共享方法,其特征在于,主要包括以下步骤:

1) 在各个移动设备上设置需要共享的数据,利用数据监控进程对共享的数据进行监控,记录数据变化信息;

2) 数据管理进程生成一个同文件系统对应的树形数据结构来管理对应的文件数据,这个树形数据结构称为文件树;

当得到来自监控进程的变化信息时,对其进行分析,依靠文件树找到对应的文件节点,并利用版本控制模块更新该节点的数据版本信息,将发生的变化信息发送给数据同步进程;

3) 数据同步进程依照设定的同步策略,将树形数据结构的变化以及需要同步的数据,通过数据传输模块与其它移动设备进行同步;

4) 接收到同步信息时,利用数据管理进程中的冲突检测与消解模块,对接收到的数据进行处理;

5) 在网络连接出现问题时利用断连处理模块进行处理,并在设备重新建立连接后进行重连处理。

2. 根据权利要求 1 所述的基于分布共享存储的移动设备数据共享方法,其特征在于,步骤 1) 的具体过程为:

11) 在局域网中部署需要使用的移动设备;

12) 设定需要同步的数据;

13) 利用 Android 系统的系统 API,对需要同步的数据进行监控,记录下数据变化的信息,并将信息发送到全局消息队列中。

3. 根据权利要求 1 所述的基于分布共享存储的移动设备数据共享方法,其特征在于,步骤 2) 的具体过程为:

21) 依据共享的文件数据,构建一个树形数据结构来表示共享的数据,该结构称为文件树;文件树的每个节点同共享的数据文件是一一对应的,它不仅包含了共享数据的路径,以便依据路径找到真实数据,还包括了数据的基本信息,也包括了数据的版本信息,除了拥有版本号之外,还拥有一个向量时钟 VC, VC 是一组向量 $[t_1, t_2, \dots, t_n]$, 每个元素 t_i 对应于该数据在 i 设备上的版本号,其初始值为 $[0, 0 \dots 0]$;

22) 分析每种数据操作所对应的数据变化信息;

23) 由保存的数据变化信息反推出在共享数据上的数据操作,包括了修改,复制,删除,重命名等;

24) 每当本地修改数据,使得数据发生变化时,都会依靠文件树找到对应的文件节点,更新数据的基本信息;在更新版本信息时,除了更新版本号外,还会增加本地向量时钟 VC 中对应于本地的部分,即 VC 中相应维数的值加 1,同时将变化信息以消息的形式发送给数据同步进程;在接收到远端的数据更新时,同样需要将本地的 VC 进行更新;

25) 当出现新建,删除,移动文件使得文件结构发生变化时,文件树的结构也要得到对应的变化,并将变化信息发送给数据同步进程。

4. 根据权利要求 1 所述的基于分布共享存储的移动设备数据共享方法,其特征在于,步骤 3) 的具体过程为:

- 31) 设计数据同步策略,包括何时同步以及如何同步;
- 32) 利用 TCP 协议在需要通信的移动设备间建立连接;
- 33) 利用 Socket 在不同设备间传输需要发送的信息以及需要同步的数据;

在进行同步时,先发送数据的基本信息与版本信息,当对方确实需要数据时,再传输文件数据;

由于基本信息与版本信息很小,避免了每次都发送数据,从而有效地提高了数据同步效率。

5. 根据权利要求 1 所述的基于分布共享存储的移动设备数据共享方法,其特征在于,步骤 4) 的具体过程为:

41) 对于接收到的数据同步信息,将其中的数据版本信息交付给冲突检测模块,同本地的版本信息进行比较;

42) 设本地的版本信息时钟向量为 $[t_1, t_2, \dots, t_n]$,其中本地及远端的对应维度向量值为 t_i, t_j ;收到的版本信息时钟向量为 $[y_1, y_2, \dots, y_n]$,其中本地及远端的对应维度向量值为 y_i, y_j ;

- a) 若 $t_i > y_i, t_j = y_j$,远端版本低于本地版本;
- b) 若 $t_j > y_j, t_i = y_i$,本地版本低于远端版本;
- c) 若 $t_i > y_i, t_j < y_j$,发生冲突;
- d) 若 $t_i < y_i$ 或者 $t_j > y_j$,理论上不应该存在这种情况;

冲突检测模块利用上述规则进行判断处理;

43) 如果出现上述的 a), b) 情况,由数据同步进程进行处理;如果出现 c),则由冲突消解模块进行处理;

44) 冲突消解模块依照开发者设定的冲突消解方法,对发生冲突的数据进行处理;

冲突消解的方法可以是对于冲突的数据,进行重命名操作;或者向用户抛出冲突信息,并停止对于冲突数据的同步;

45) 如果接受到的是数据操作命令,则由数据管理进程进行处理;

数据操作命令主要是数据请求操作;

46) 在上述各种操作中,如果涉及了本地数据的更新,都需要利用版本管理模块对于版本信息进行更新;如果涉及了文件结构的变化,则要修改文件树。

6. 根据权利要求 1 所述的基于分布共享存储的移动设备数据共享方法,其特征在于,步骤 5) 的具体过程为:

51) 数据传输模块维护设备间的连接,利用发送心跳帧来判断是否发生断连;

52) 当设备间连接发生中断时,断连处理模块将发生断连时的状态进行记录,主要有在发生断连时,同远端共享的数据的版本号,名称,路径等;如果在双方发送信息时发生断连,还需要将信息保存下来,留待重连后再发送;

53) 当设备间重新建立连接后,断连处理模块利用之前保存的信息,同远端进行通信,并将接收到的信息交给数据管理进程,进而将需要同步的数据进行同步。

基于分布共享存储的移动设备数据共享方法

[0001]

技术领域

[0002] 本发明涉及一种基于分布共享存储的移动设备数据共享方法,属于计算机应用技术领域,具体涉及在局域网络中的移动设备上构建一个数据共享平台。

背景技术

[0003] 随着移动智能设备的性能越来越高,存储量越来越大,日常生活已离不开移动设备的参与。在这种情况下,在不同移动设备间进行数据的共享就成为了人们的迫切需求。目前,移动设备间的数据共享主要有两种方式:Hub-and-Spoke 以及 Cloud Storage。

[0004] 在 hub-and-spoke 系统中,用户需要指定一台特定的机器作为整个系统的“主节点”,即 hub device,它保存了所有共享的数据的副本。其他设备如果要进行通信,必须首先同“主节点”进行连接,将需要发送的数据首先拷贝到“主节点”上,然后“主节点”再将数据推送到其他的设备上。在进行连接时,可以利用快速的本地连接,比如直接连接或者局域网。由其特点可知,在 hub-and-spoke 模型中,只需要查看 hub device,就可以很容易的得到所有共享的数据。此外,由于所有操作都需要通过唯一的 hub device 来进行,因此,对于同步问题,该模型也可以很容易处理。但是,该系统也有其缺陷:在一个 hub-and-spoke 系统中, hub device 必须存在;一旦 hub device 崩溃或者断连,则整个系统无法正常工作;它仅在星型的拓扑结构中高效运行,对于其他拓扑结构,它的效率就不高了;如果两台设备有直接连接,它们直接的通信仍需要通过 hub device 来进行;Hub device 保存了所有共享的数据副本,它的存储容量成为整个系统的瓶颈。

[0005] 另一种共享数据的方式 Cloud Storage,也就是常说的云存储方式,它是利用存在于互联网中的多台服务器来替代 hub-and-spoke 中的 hub device。这些服务器基于分布式系统进行构建,但对于用户而言,它们的拓扑结构并不重要。用户只需要登录一个网站,就可以查看存储在各个服务器上的文件,而不需要知道文件具体存储在哪个服务器上。相较于 hub-and-spoke 模型,云存储更可靠。由于存在多台服务器,且数据往往都会有冗余,因此,即使其中一些服务器无法运行,其他服务器仍能提供服务。此外,由于有整个互联网中服务器的支持,在 hub-and-spoke 中存储容量的限制,在云存储中几乎不存在。但是云存储也带来了一些新的限制:服务提供商可能会限制支持的数据类型;设备仍然无法直接进行通信,即使它们共处于一个快速的局域网中;文件操作需要通过网络服务提供商来进行,受到了网络带宽等的限制;数据存储在服务中,可能出现安全与隐私问题。

发明内容

[0006] 发明目的:本发明要解决的技术问题是在局域网中提供一种基于分布共享存储的移动设备数据共享平台的构建方法,该平台不需要一台集中的服务器来支持,能够供开发者设计自己的数据同步与冲突消解策略,并提供其它应用基于该平台进行快速开发的接

口。

[0007] 技术方案：一种基于分布共享存储的移动设备数据共享方法，其主要包括以下步骤：

1) 在各个移动设备上设置需要共享的数据，利用数据监控进程对共享的数据进行监控，记录数据变化信息；

2) 数据管理进程为了能够管理共享的文件，会生成一个同文件系统对应的树形数据结构来管理对应的文件数据，这个树形数据结构称为文件树。操作系统的文件系统是用来管理本地文件，并且提供接口来让其他应用对文件执行读写等操作的。而数据管理进程利用文件树，也能够实现对共享文件的管理。当得到来自监控进程的变化信息时，对其进行分析，依靠文件树找到对应的文件节点，并利用版本控制模块更新该节点的数据版本信息，将发生的变化信息发送给数据同步进程；

3) 数据同步进程依照设定的同步策略，将树形数据结构的变化以及需要同步的数据，通过数据传输模块与其它移动设备进行同步；

4) 接收到同步信息时，利用数据管理进程中的冲突检测与消解模块，对接收到的数据进行处理；

5) 在网络连接出现问题时利用断连处理模块进行处理，并在设备重新建立连接后进行重连处理。

[0008] 上述步骤 1) 的具体过程为：

11) 在局域网络中部署需要使用的移动设备；

12) 设定需要同步的数据；

13) 利用 Android 系统的系统 API，对需要同步的数据进行监控，记录下数据变化的信息，并将信息发送到全局消息队列中。

[0009] 在 Android 系统中，利用系统 API 进行数据的监控时，一旦数据发生变化，都会抛出一系列的响应事件。对于哪些事件感兴趣，就可以在系统 API 中进行注册。

[0010] 上述步骤 2) 的具体过程为：

21) 依据共享的文件数据，构建一个树形数据结构来表示共享的数据，该结构称为文件树。文件树的每个节点同共享的数据文件是一一对应的，它不仅包含了共享数据的路径，以便依据路径找到真实数据，还包括了数据的基本信息，包括大小，作者，修改时间等，也包括了数据的版本信息，除了拥有版本号之外，还拥有有一个向量时钟 VC，VC 是一组向量 $[t_1, t_2, \dots, t_n]$ ，每个元素 t_i 对应于该数据在 i 设备上的版本号，其初始值为 $[0, 0 \dots 0]$ ；

每个文件同文件树中的节点都是一一对应的关系。每个文件节点都包含了基本信息，版本信息等，同时还提供了读写等操作供其他模块使用。

[0011] 22) 分析每种数据操作所对应的数据变化信息；

23) 由保存的数据变化信息反推出在共享数据上的数据操作，包括了修改，复制，删除，重命名等；这些文件数据操作，每一个都对应于一个或多个系统事件。其中文件移动和文件夹移动都对应了多个事件序列，这是由它们的起始目录和目标目录是否被监听所决定的。

[0012] 24) 每当本地修改数据，使得数据发生变化时，都会依靠文件树找到对应的文件节点，更新数据的基本信息。在更新版本信息时，除了更新版本号外，还会增加本地向量时

钟 VC 中对应于本地的部分,即 VC 中相应维数的值加 1,同时将变化信息以消息的形式发送给数据同步进程;在接收到远端的数据更新时,同样需要将本地的 VC 进行更新。

[0013] 25) 当出现新建,删除,移动文件使得文件结构发生变化时,文件树的结构也要得到对应的变化,并将变化信息发送给数据同步进程。

[0014] 上述步骤 3) 的具体过程为:

- 31) 设计数据同步策略,包括何时同步以及如何同步;
- 32) 利用 TCP 协议在需要通信的移动设备间建立连接;
- 33) 利用 Socket 在不同设备间传输需要发送的信息以及需要同步的数据。

[0015] 在进行同步时,先发送数据的基本信息与版本信息,当对方确实需要数据时,再传输文件数据。由于基本信息与版本信息很小,避免了每次都发送数据,从而有效地提高了数据同步效率。

[0016] 上述步骤 4) 的具体过程为:

41) 对于接收到的数据同步信息,将其中的数据版本信息交付给冲突检测模块,同本地的版本信息进行比较;

42) 设本地的版本信息时钟向量为 $[t_1, t_2, \dots, t_n]$,其中本地及远端的对应维度向量值为 t_i, t_j ;收到的版本信息时钟向量为 $[y_1, y_2, \dots, y_n]$,其中本地及远端的对应维度向量值为 y_i, y_j 。

[0017] a) 若 $t_i > y_i, t_j = y_j$,远端版本低于本地版本;

b) 若 $t_j > y_j, t_i = y_i$,本地版本低于远端版本;

c) 若 $t_i > y_i, t_j < y_j$,发生冲突;

d) 若 $t_i < y_i$ 或者 $t_j > y_j$,理论上不应该存在这种情况。

[0018] 冲突检测模块利用上述规则进行判断处理;

对于大于两台设备的环境下,由于设备间的通信也是两两进行的,因此在两两之间利用上述规则进行判断。

[0019] 43) 如果出现上述的 a), b) 情况,由数据同步进程进行处理;如果出现 c),则由冲突消解模块进行处理;

44) 冲突消解模块依照开发者设定的冲突消解方法,对发生冲突的数据进行处理。冲突消解的方法可以是对于冲突的数据,进行重命名操作;或者向用户抛出冲突信息,并停止对于冲突数据的同步等;

45) 如果接受到的是数据操作命令,则由数据管理进程进行处理。数据操作命令主要是数据请求操作等;

46) 在上述各种操作中,如果涉及了本地数据的更新,都需要利用版本管理模块对于版本信息进行更新。如果涉及了文件结构的变化,则要修改文件树。

[0020] 上述步骤 5) 的具体过程为:

51) 数据传输模块维护设备间的连接,利用发送心跳帧来判断是否发生断连;

52) 当设备间连接发生中断时,断连处理模块将发生断连时的状态进行记录,主要有在发生断连时,同远端共享的数据的版本号,名称,路径等;如果在双方发送信息时发生断连,还需要将信息保存下来,留待重连后再发送;

53) 当设备间重新建立连接后,断连处理模块利用之前保存的信息,同远端进行通信,

并将接收到的信息交给数据管理进程,进而将需要同步的数据进行同步。

[0021] 有益效果:本发明提出的一种在局域网络中构建基于分布共享存储的移动设备数据共享平台的构建技术,有别于传统的 Hub-and-Spoke 以及 Cloud Storage 技术,利用版本管理以及向量时钟,在不需集中服务器的情况下,能够在多台设备间建立数据的同步共享,并能够对数据的冲突情况进行处理。同时,与现有的分布共享存储技术所不同的是,现有的分布共享存储技术的存储同步对象都是孤立的,而本平台所存储同步的对象之间是有关联的,即在进行同步时,同步的是一个树形的数据结构。在同步的过程中,都是以围绕着平台所维护的文件树来进行的。

附图说明

[0022] 图 1 平台整体结构图;

图 2 Android 系统文件基本事件;

图 3 文件树的结构及其同共享文件的对应关系;

图 4 Android 系统文件操作对应的基本事件序列;

图 5 向量时钟的更新及冲突检测及版本的更新判断;

图 6 传输信息的格式。

具体实施方式

[0023] 下面结合具体实施例,进一步阐明本发明,应理解这些实施例仅用于说明本发明而并不用于限制本发明的范围,在阅读了本发明之后,本领域技术人员对本发明的各种等价形式的修改均落于本申请所附权利要求所限定的范围。

[0024] 1) 数据变化的获取

数据变化的获取在本发明中起了较为重要的作用。只有能够及时准确的获取数据的变化信息,才能在不同移动设备进行数据的共享更新。

[0025] 在 Android 系统中,利用系统 API 进行数据的监控时,一旦数据发生变化,都会抛出一系列的响应事件。这些基本的事件如图 2 所示。对于哪些事件感兴趣,就可以在系统 API 中进行注册。通过继承 Android 系统提供的 FileObserver 类,用户可以定义自己的 FileObserver。在使用时,只需将要监听的文件(文件夹)路径作为参数传入,启动监听,Android 系统即可对该文件(文件夹)进行监听。当监听的文件发生变化时,系统即会发送一个事件响应,该事件响应会被相应的 FileObserver 捕获,获得的参数包括了事件类型以及路径信息两方面。由于 FileObserver 类无法对子文件夹中的文件数据进行监听,因此,需要为每一个需要监听的文件夹设置一个 FileObserver 对象。为了保存 FileObserver 对象的引用(必须保存,不然系统无法继续监听),需要一个管理模块来对其进行管理。

[0026] 2) 文件树的组织与维护

平台在运行过程中,数据管理模块始终维护着一棵文件树,它是共享文件数据的一个镜像,树上的每一个节点都与需要同步的文件数据一一对应。文件树同共享文件的对应关系如图 3 所示,左边是文件系统中共享文件的结构,右边是平台维护的文件树。每个文件同文件树中的节点都是一一对应的关系。每个文件节点都包含了基本信息,版本信息等,同时还提供了读写等操作供其他模块使用。

[0027] 当本地文件发生修改时,依靠文件树可以快速定位到对应的文件节点,并对文件的基本信息以及版本信息进行更新;当文件被删除或者移动时,也需要找到对应的文件节点,将其删除或者修改移动;当有新的文件被建立时,同样需要在对应位置生成一个新的文件节点,并进行初始化。在进行同步时,首先会同步文件树,再按照不同的同步策略来同步文件数据。

[0028] 同时,文件树需要提供读、写、删除、移动等接口来供其他模块使用。当通过这些接口来进行读、写等操作时,文件树同样也需要得到更新与维护。

[0029] 3) 数据版本信息的管理

由于在本数据共享平台中,并没有一台集中式的服务器来管理各种共享的数据,设备间的数据就是依靠各台设备上所保存的各种数据版本信息来进行同步的。因此,数据版本信息在实现数据的同步以及数据的冲突消解中都起着至关重要的作用。

[0030] 数据的版本信息主要包括了数据的基本信息以及数据的版本号两部分。其中数据的基本信息主要包括了数据的大小,修改日期,路径等。当数据被修改或者更新时,这些基本信息也要进行对应的更新。数据的版本号则主要包括了一个版本信息时钟向量。在更新版本信息时,除了更新版本号外,还会增加本地向量时钟 VC 中对应于本地的部分,即 VC 中相应维数的值加 1,同时将变化信息以消息的形式发送给数据同步进程;

VC 的变化情况如图 5 所示。在接收到远端的数据更新时,同样需要将本地的 VC 进行更新。下面给出版本信息时钟向量的计算方法:

假设在局域网中有 n 台设备共享一份数据 A,则在每台设备上都保存着一份关于数据 A 的版本信息时钟向量 $[t_1, t_2, \dots, t_n]$ 。

[0031] 31) 当本地设备对数据 A 进行修改时,就对本地所对应的分量进行 +1 的操作,并将更新后的时钟向量发送给其他设备。

[0032] 32) 当本地接收到来自其他设备的关于 A 数据的时钟向量 $[y_1, y_2, \dots, y_n]$ 时,将本地的时钟向量同接收到的时钟向量进行比较。不妨设在 $[t_1, t_2, \dots, t_n]$ 中本地及远端设备的值为 t_i, t_j ,在 $[y_1, y_2, \dots, y_n]$ 中本地及远端设备的值为 y_i, y_j 。按照下面的方式进行比较:

- a) 若 $t_i > y_i, t_j = y_j$,远端版本低于本地版本;
- b) 若 $t_j > y_j, t_i = y_i$,本地版本低于远端版本;
- c) 若 $t_i > y_i, t_j < y_j$,发生冲突;
- d) 若 $t_i < y_i$ 或者 $t_j > y_j$,理论上不应该存在这种情况。

[0033] 33) 当本地设备从其他设备上获得更新的数据后,需要将本地保存的时钟向量对应的值进行更新。如图 4 所示。

[0034] 34) 这样,维护的时钟向量就可以用来对不同的文件数据版本进行比较与判断。如图 5 所示。

[0035] 4) 设备的连接管理与数据传输

由于移动设备自有的可移动性,在数据平台的运行过程中,设备之间很有可能会发生网络断连的情况。当设备之间发生断连时,由于无法进行通信,则之后的数据更新将无法进行同步。为了能够在重新连接后进行同步,必须在断开连接时记录下当时的版本信息,并以文件的形式保存在本地。

[0036] 设备间利用 TCP 进行长连接,并通过发送心跳帧的形式来判断设备间是否发生了断连。当设备间连接发生中断时,断连处理模块将发生断连时的状态进行记录,主要有在发生断连时,同远端共享的数据的版本号,名称,路径等,并以文件的形式保存到移动设备上;如果在双方发送信息时发生断连,还需要将信息保存下来,留待重连后再发送。

[0037] 由于在设备间传输的信息包括了数据本身以及其他信息(包括了版本信息,数据操作命令等)。因此,为了区分不同的数据信息,在利用 socket 进行传输时,需要定义信息格式,以此来判断接收到的信息是什么类型,如图 6 所示。不同的信息其类型字段不同,由于不同的信息其参数个数不定,参数间利用特殊符号进行分割。数据字段则只有发送的是数据时才拥有。

[0038] 以下通过具体的实施例来对本发明的技术方案进行详细的说明。

[0039] 首先描述一个典型的应用场景。

[0040] 在小组研讨会上,将黑板上的讨论内容拍摄了下来,希望能够在其它设备上也能查看拍摄的照片。

[0041] 下面具体按照上面的步骤来详细说明本场景中的具体行为。

[0042] 1. 将需要同步的设备接入相同的局域网中,并通过 TCP 进行连接。摄影软件将照片所在文件夹的路径以参数的形式传递给数据共享平台。平台创建一个新的FileObserver来对该文件夹进行监控,此时会生成一棵文件树,并生成默认的基本信息以及版本信息,如图 3 所示。在选择需要共享的数据时,也可以由用户手动选择。

[0043] 2. 当有新的照片拍摄后,按照图 4 中所示的文件操作与事件序列的对应关系,文件监控能够发现新文件的生成。同样的,对拍摄的照片进行重命名或者美化裁剪等操作时,按照图 4 中所示的文件操作与事件序列的对应关系,文件监控同样能够发现这些信息。文件监控会将数据的变化消息发送到数据管理模块,并由数据管理模块依靠文件树找到对应的文件节点,更新数据信息与版本信息。如果对应的文件节点不存在,也就是生成了新的文件,就会在对应的位置生成一个新的文件节点并初始化数据。数据管理模块在接收到数据变化的信息并更新了文件树后,会将该信息发送到数据同步模块,并由数据同步模块按照设定的同步策略来进行文件树与数据的同步。进行同步时,先同步文件树再同步数据。只有当对方同步了文件树并请求该数据时,才利用数据传输模块将对方请求的数据传过去。由于同步策略可能会因为不同的需求而有所不同,为了满足这种需求,需要设计一个同步策略的接口类。当需要设计一个特定的策略时,只要实现这个接口就可以了。

[0044] 3. 接收方在接收到发送方发送的数据版本信息后,将本地的版本信息时钟向量和接收到的时钟向量进行比较,并依据结果进行相应的处理,如图 5 所示。当获取了来自发送方的新版本数据后,对应的需要更新文件树中保存的本地数据的基本信息以及版本信息。

[0045] 4. 在同步的过程中,有可能设备因为离开了局域网或者出现了其他意外从而断开了与其他设备的连接。这时候连接管理模块会发现发生了断连,并利用断连处理模块来进行处理,它会将共享的照片的版本号,名称,路径等以文件的形式保存到移动设备上。当重新建立连接后,利用之前保存的版本信息来进行重新的同步。同步的步骤同样是发送照片的版本信息,通过比较来进行相应的处理。

[0046] 5. 当发生冲突时,依照设定的冲突消解策略对冲突的照片进行处理。冲突策略的设定可以依照不同的需求进行定制。为了满足这种需求,需要设计一个冲突消解的接口类。

当需要设计一个特定的策略时,只要实现这个接口即可。

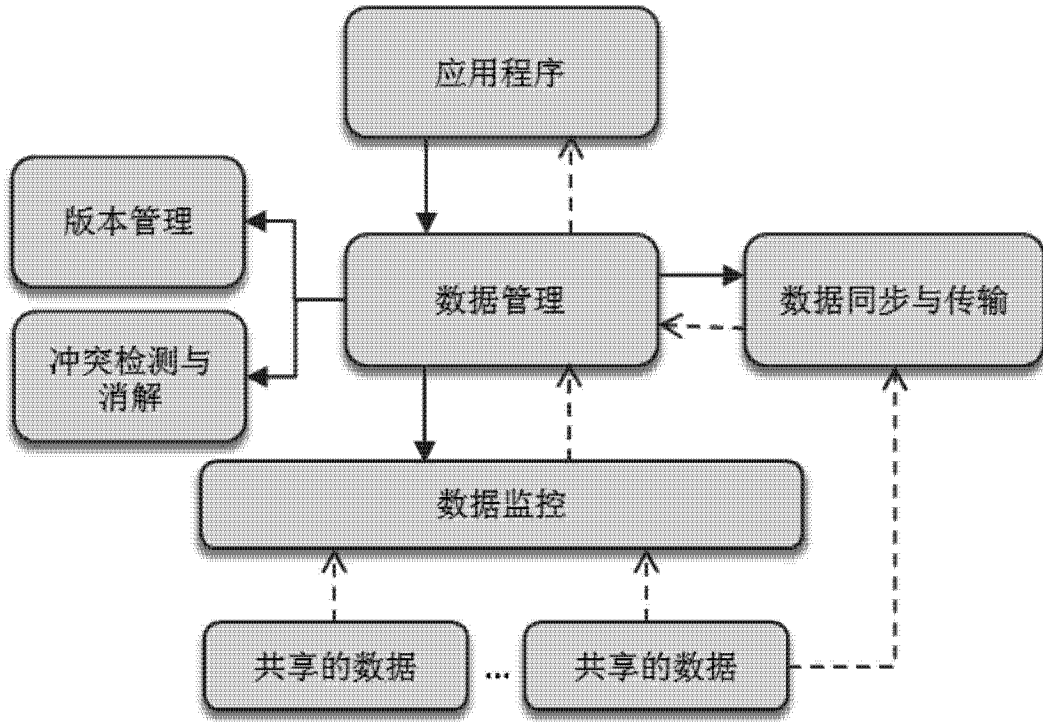


图 1

Type	Meaning
ACCESS	Data was read from a file
ATTRIB	Metadata(permissions,owner,timestamp)was changed explicitly
CLOSE_NOWRITE	Someone had a file or directory open read-only,and closed it
CLOSE_WRITE	Someone had a file or directory for writing,and closed it
CREATE	A new file or subdirectory was created under the monitored directory
DELETE	A file was deleted from the monitored directory
DELETE_SELF	The monitored file or directory was deleted
MODIFY	Data was written to a file
MOVED_FROM	A file or subdirectory was moved from the monitored directory
MOVED_TO	A file or subdirectory was moved to the monitored directory
MOVE_SELF	The monitored file or directory was moved
OPEN	A file or directory was opened

图 2

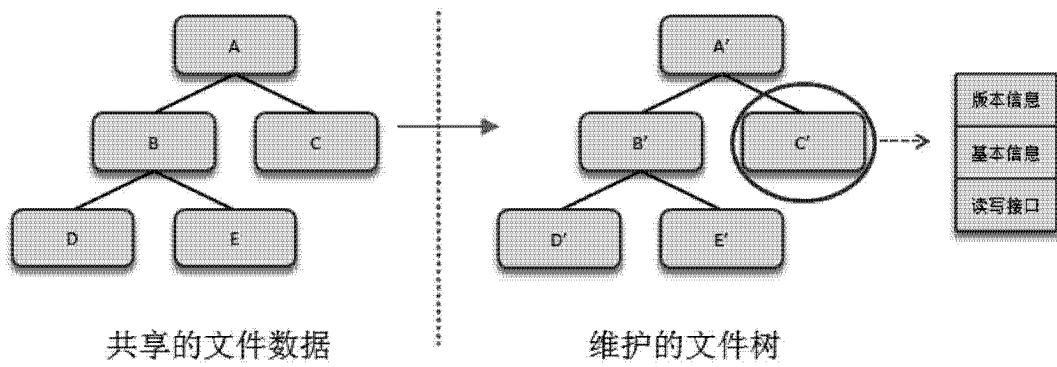


图 3

操作	简化的事件序列
文件夹创建/复制	CREATE
文件创建/复制	CREATE + MODIFY+ CLOSE_WRITE
文件夹删除	DELETE + MOVE_FROM+ DELETE_SELF
文件删除	DELETE
文件移动1	MOVE_FROM
文件移动2	MOVE_TO
文件移动3	MOVE_FROM+ MOVE_TO
文件夹移动1	MOVE_FROM+ MOVE_SELF
文件夹移动2	MOVE_TO
文件夹移动3	MOVE_TO+ MOVE_SELF
文件夹移动4	MOVE_FROM+ MOVE_TO+MOVE_SELF

图 4

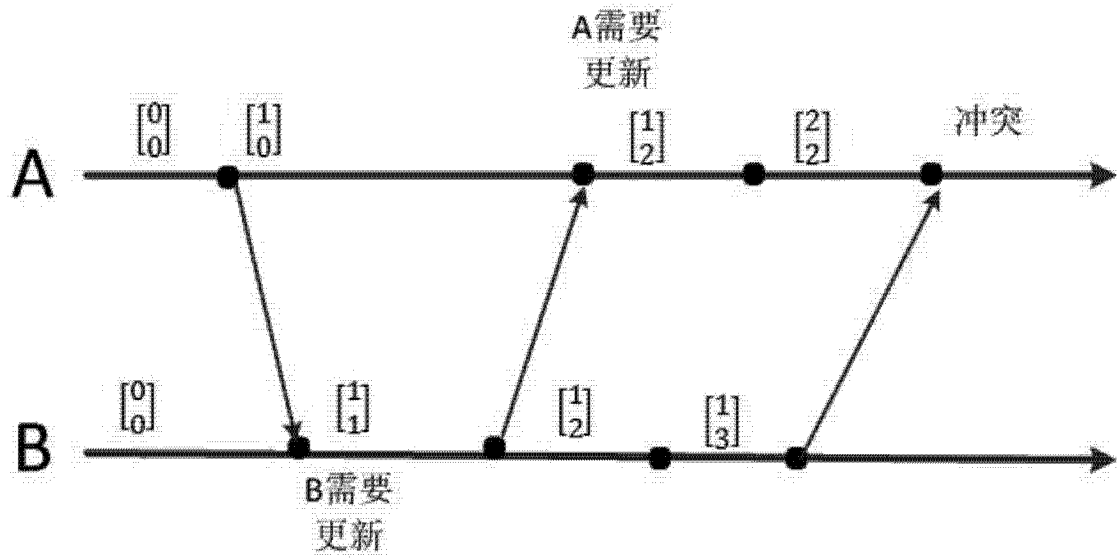


图 5

类型	参数1	参数2	...	参数n	数据
----	-----	-----	-----	-----	----

图 6