

文章编号:1001-9081(2006)07-1676-03

基于数据复制技术实现移动数据同步

王文琴¹, 费贤举¹, 鞠时光²

(1. 常州工学院 计算机信息工程学院, 江苏 常州 213002;

2. 江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

(wwqin@czu.cn)

摘要:在移动计算环境中,数据同步使人们能够使用各种各样的无线或者移动设备处理和存储各种数据。介绍了一个使用通用同步协议 SyncML、基于数据复制技术实现移动数据同步的系统的设计和实现方法。

关键词:数据同步; SyncML; 冲突; 同步服务器

中图分类号: TP311.132 **文献标识码:** A

Implementation of mobile data synchronization based on data replication technology

WANG Wen-qin¹, FEI Xian-ju¹, JU Shi-guang²

(1. School of Computer & Information Engineering, Changzhou Institute of Technology, Changzhou Jiangsu 213002, China;

2. School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: Data synchronization is becoming a new important technology in the mobile computing environment. Based on it, people can process and store data in various wireless devices or mobile devices. A mobile data synchronization system was designed and implemented. It used common synchronization protocol SyncML and was based on data replication technology.

Key words: data synchronization; SyncML; conflict; synchronization sever

0 引言

移动计算技术的不断发展,给人们提供了越来越多的便利。一个最明显的好处就是,人们可以随时随地地使用最方便的设备将个人信息记录下来,以免忘记。例如,当人们在办公室工作时,可以将个人信息记录在 PC 或笔记本电脑上;当人们在旅途中,可以将自己的个人信息记录在手持 PDA 或手机中。这一切都标志着移动计算正在得到越来越多的应用,但如果不能实现移动设备之间的数据同步,那么这种便利就无法真正发挥作用。

移动计算环境的一个显著特点是移动终端与服务器之间的连接是一种弱连接,即低带宽、长延迟、不稳定和经常性的断开。为了支持用户在弱环境下对数据库的操作,现在普遍采用乐观复制方法(Optimistic replication 或 Lazy replication)。即移动端将需要的数据先复制到移动设备的存储介质上,形成移动节点上的本地数据部分,在移动节点和网络断开连接时允许用户对本地缓存上的数据副本进行操作;待移动节点和网络重新连接后再与数据库服务器交换数据修改信息,并通过冲突检测和协调来恢复数据的一致性。本文介绍了使用 SyncML 协议、基于数据复制技术实现移动数据同步的解决方案。

1 数据同步业务模型

数据同步业务是一种支持两数据源间进行数据同步的机制,由于个人用来存放信息的电子设备种类繁多,而这些电子设备之间物理连接的方式也是多样的,例如:红外、蓝牙、GPRS、CDMA、电缆等。所以,数据同步业务不应该对具体的

承载方式进行约定。

本文的实现采用了如图 1 所示的模型。在这个环境中,网络分为固定网络和无线网络两部分,相对于可靠性不高的无线网络单元,我们将固定网络部分称为可信部分^[1]。我们假设整个系统由 4 类节点组成:1)数据库服务器,由各主流商用数据库服务器组成;2)固定客户端,通过固定网络直接和数据库服务器连接;3)移动客户端,具有移动性的主机,经常与服务器断接,通过无线网络和同步服务器进行通信,这一层无线网络可以采用多种无线通讯技术,例如无线局域网、GSM、GPRS 等;4)同步服务器,负责整个系统的同步。同步服务器是移动节点连接到固定网络的唯一接口,所有移动节点和中心数据库服务器之间的数据流都要通过同步服务器来传递。

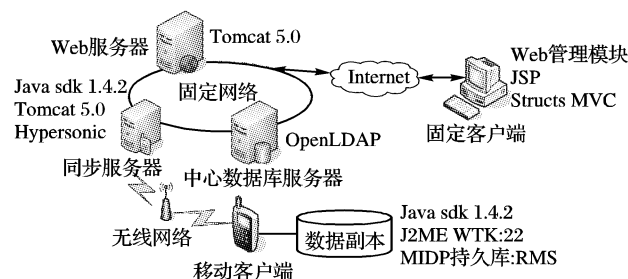


图 1 系统模型

同步服务器在整个移动数据库系统的体系结构中处于中间服务层,是连接移动节点和固定网络的桥梁,也是整个系统的关键部件。正是由于同步服务器中间层的引入,使得整个系统具有较灵活的可扩充性。而且同步服务器通过标准的数据库接口如 JDBC 和中心数据库通讯,使得中心数据库系统

收稿日期:2006-01-10;修订日期:2006-03-07

作者简介:王文琴(1970-),女,江苏镇江人,讲师,硕士研究生,主要研究方向:数据库应用、移动数据库; 费贤举(1975-),男,安徽巢湖人,讲师,硕士研究生,主要研究方向:软件工程、数据库、知识库系统; 鞠时光(1955-),男,江苏南通人,教授,博士生导师,主要研究方向:数据库、可视化语言、信息安全。

可以使用通用的商业数据库服务器。

2 SyncML 数据同步协议

本文在同步服务器和移动设备之间采用 SyncML 技术进行数据同步。SyncML 是一种具有行业标准性质的同步协议,利用 SyncML 技术,用户可以对大量类型的数据进行同步。SyncML 基于 XML 定义,使用 XML 消息来添加、删除和修改数据单元。它也允许交换安全信息,这样节点就可以执行身份验证。

SyncML 不仅定义了一种格式,而且定义了一种概念上的数据同步框架和数据同步协议。在图 2 所示的 SyncML 协议框架结构中,虚线框内是 SyncML 框架的范围,虚线框外的部分为数据同步协议。在整个 SyncML 同步协议的框架中,SyncML 包和底层的链路传输是分离的,因此从理论上讲 SyncML 消息可以在任何一种网络链接上进行同步操作。目前同步协议定义的传输绑定协议有 HTTP, WSP, OBEX 3 种,在传输层绑定的这 3 种传输协议基本上覆盖了大部分网络链接和标准短程链接^[2,3]。

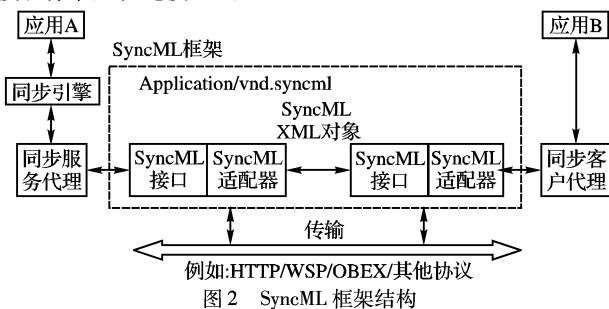


图2 SyncML 框架结构

3 同步处理策略

3.1 ID 处理

| 移动设备 | | 服务器设备 | |
|---------|---------|------------|---------|
| 客户端数据库: | | 服务器端数据库: | |
| LUID | Data | GUID | Data |
| 1 | Tom | 10001 | John |
| 2 | John | 10002 | Rebecca |
| 3 | Rebecca | 10003 | Tom |
| | | 数据项的ID映射表: | |
| | | GUID | LUID |
| | | 10001 | 2 |
| | | 10002 | 3 |
| | | 10003 | 1 |

图3 LUID, GUID 及映射表在通信实体上的分布

作为复制的基本需求,我们需要唯一地标识将要复制的每个数据单元,可以采用各种各样的方案。对于单向复制,主数据库可以要求为数据单元生成 ID;对于双向复制,需要一个能够标识数据库之间对应数据单元的映射方案。在本文介绍的系统中,采用了服务器端和移动端分别为数据项分配本地唯一标识号的办法。其中服务器端对数据库记录的唯一标识称为 GUID(Global Unique Identifier),客户端对数据库记录的唯一标识称为 LUID(Locally Unique Identifier)。这样,相同的一条记录,共存在两个 ID,所以整个系统中还需存在一个 GUID 和 LUID 的映射表。考虑到同步客户端通常是内存受限设备,所以这个 LUID 和 GUID 的映射表被服务器所维护。

3.2 数据库的更新检测算法

由于移动设备的频繁断接性和资源限制,本系统采用了 Session-Based 的同步方式,而且在每个同步 Session 中传递的内容主要是事务日志,这种方式只需交换数据库的更新部分即可,不必交换整个数据库,使得复制过程更加快捷。对于数据库的更新部分的检测,本文在客户端和服务端分别采用

了如下算法。

3.2.1 同步服务器端

1)同步服务器端通过一个修改日志记录服务器上数据的修改情况,修改日志上保存的信息有:被修改记录的键值、操作类型(新增 N、更新 U、删除 D)以及记录修改时间,在每次数据同步完成后修改日志被清空。

2)考虑系统应用于多用户环境下,为了减少同步时服务器端处理日志的时间,服务器端将日志根据用户进行分片,每个用户都拥有属于自己的独享的日志。

3)考虑每个用户可能使用多个移动设备与中心数据库进行数据共享,此时同一用户账号的所有设备在与服务器同步时,对应的数据源、访问的日志是相同的。如果采用上面的日志方式,当某个设备和服务器数据同步后,服务器端的日志信息被清空,其他设备将无法获取服务器上数据的修改信息,这样的方式显然不合适;但如果不清空日志,经过多次同步后,服务器端日志处理效率将会很低。因此为了支持同一用户的多设备同步,在修改日志中增加了一个修改次数项。改进后的单条日志格式为:

记录键值 = 修改次数 | 修改类型 | 修改时间

4)当固定客户端通过固定连接数据库服务器直接修改数据项 O_i 时,服务器端在此用户的修改日志中添加一条日志信息,其中修改次数的初始值为该用户拥有的移动设备数量(DeviceNum);当用户的某个移动设备通过移动网络与同步服务器同步,修改数据项 O_i 时,服务器在此用户的修改日志中也添加一条日志信息,其中修改次数的初始值为 DeviceNum - 1。以下为日志文件中的一条日志信息,其中数据项 O_i 的键值被编码后保存。

YWJj = 3U1135133748890

5)假设用户 i 的一个移动设备 Device(i),上次与系统同步的时间为 LastAnchor(Device(i))。当此设备与系统连接执行同步时,服务器端首先读取该用户的日志信息,并将其修改时间 > LastAnchor(Device(i)) 的日志所对应的数据项的完整信息发送到移动设备端进行处理;其次将处理过的日志的修改次数自动减 1。

6)当此用户的最后一个移动设备与系统同步后,此时该日志的修改次数为零,该条修改日志将被自动删除。

3.2.2 移动设备端

在移动设备端,由于信息比较单一、数据量较小,因此通过采用附加信息(状态标志位)来记录数据单元的状态。移动设备端记录结构为:

记录键值 | 修改状态位 | 记录内容

其中修改状态位为: N(新增记录)、U(记录被更新)、D(记录被删除)、S(记录已被同步过),且系统执行时不显示修改标志位为 D 的记录。当同步开始时,系统中所有修改状态位不为 S 的记录均为修改记录,其完整信息封装在同步数据包中,发送到同步服务器端等待处理;同步结束后,系统将所有同步过的记录的修改状态位改为 S,并将状态标志位为 D 的记录从系统删除。

3.3 相似性判断算法

在慢同步方式下或当同步数据包中包含追加记录时,服务器端或移动设备端在将此记录追加到本地数据库前,有必要先在本地数据库上查找是否有此记录的相似项,以免造成不必要的重复记录。一般情况下我们可以通过查找记录的 ID 号是否相同来判断。但在移动数据同步系统中,由于同一条记录可能在不同的移动终端上添加,从而分配有不同的 ID 号,因此这种方法不能实现。另外假设在一个通讯簿中,对于具有相同 Name 的记录、或具有相同手机号码的记录、或具有

相同 E-mail 的记录,其记载的是否是同一对象的信息?在本文中采用的算法通过授权用户为每个字段分配不同的相似权重值,从而实现了具有自适应性的相似性判断算法。以下通过一个模拟的通讯簿数据表为例说明了此算法的实现方法。

1) 参照如表 1 所示的方法为数据表中各个字段设置相似权重值及最低相似点数。

表 1 字段相似性权重值

| i | Field(i) | Identical(i) | Different(i) |
|-----|-----------------|--------------|--------------|
| 1 | First Name | +10 | -20 |
| 2 | Last Name | +10 | -40 |
| 3 | Email; Internet | +10 | -20 |
| 4 | Phone; Home | +10 | -20 |
| 5 | Phone; Work | +10 | -20 |
| ... | ... | ... | ... |

PointsNeeded(最低相似点数):25

2) 将需要比较的两个记录 ItemA 和 ItemB 的每个字段参照表 1 进行比较,获得两个记录的相似点数 SimilarPoints,其计算方法如下:

i = 1

Do Until i > 字段总数

If ItemA.Fields(i) Is Not Null and ItemB.Fields(i) Is Not Null Then

If ItemA.Fields(i) = ItemB.Fields(i) Then

SimilarPoints = SimilarPoints + Identical(i)

Elseif ItemA.Fields(i) < > ItemB.Fields(i) Then

SimilarPoints = SimilarPoints + Different(i)

End If

End If

i = i + 1

Loop

If SimilarPoints > PointsNeeded Then

ItemA 与 ItemB 记载的是同一对象信息

Else

ItemA 与 ItemB 记载的是不同对象的信息

End If

3.4 数据复制中的问题与解决方案

理想的数据复制应该是一种异步复制,即允许移动设备在断接的情况下在其本地副本上执行读写操作,从而造成系统短暂的不一致。重新连接时进行数据同步处理,将移动设备上的数据修改上传到固定网络上的中心数据库,同时将中心数据库上所作的修改信息下载到移动设备端,从而使系统重新达到一致的状态。在此过程中,必须进行冲突处理,因为系统中不同节点可能对同一数据项执行了冲突操作,于是产生数据一致性问题^[4]。本系统中的同步服务器按照表 2 所示规则,对服务器端和移动客户端数据库中记载的对同一记录的操作状态进行比较,从而判断是否存在冲突。

表 2 冲突检测规则

| 数据库 A | 数据库 B | | | | |
|-------|-------|----|----|-----|-----|
| | 新增 | 删除 | 更新 | 已同步 | 不存在 |
| 新增 | C | C | C | C | B |
| 删除 | C | × | C | D | × |
| 更新 | C | C | C | B | B |
| 已同步 | C | D | A | = | B |
| 不存在 | A | × | A | A | × |

表中各符号的意义如下:数据库 A,客户端数据库;数据库 B,服务器端数据库;×,不做任何操作;A,用 A 端的数据修改 B 端的数据;B,用 B 端的数据修改 A 端的数据;C,冲突;D,从所属数据源中删除。

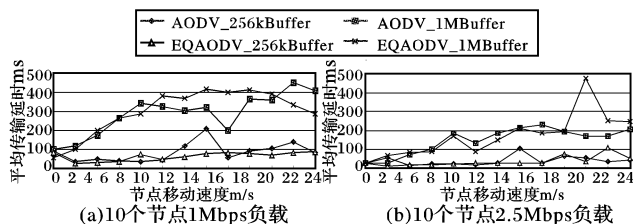
当冲突发生的时候,需要采用适当的方法来消除冲突,以保证数据的一致性。因此需要在同步服务器中设立冲突仲裁机制对冲突做出最后的裁决。常用方法有以下几种:回送信息给用户由用户决定、移动端优先、服务器端优先、以数据操作的先后顺序来解决冲突等^[5]。考虑到移动设备的不稳定性,本文中采用了服务器端优先的策略来解决冲突问题。

参考文献:

- [1] SERRANO-ALVARADO P, RONCANCIO C. A Survey of Mobile Transactions[J]. Distributed and Parallel Databases, 2004, 16(2): 193-230.
- [2] SyncML Representation Protocol Version 1.2[S/OL]. http://www.openmobilealliance.org/release_program/docs/DS/V1_2_C/OA-SyncML-DataSyncRep-v1_2-20040601-C.pdf. Open Mobile Alliance, 2004-06.
- [3] SyncML Sync Protocol Version 1.2[S/OL]. http://www.openmobilealliance.org/release_program/docs/DS/V1_2_C/OA-SyncML-DataSyncProtocol-V1_2-20040601-C.pdf. Open Mobile Alliance, 2004-06.
- [4] 王珊,丁治明,张孝. 移动数据库及其应用[J]. 计算机应用, 2000, 20(9): 1-4.
- [5] TÜRKER C, ZINI G. A Survey of Academic and Commercial Approaches to Transaction Support in Mobile Computing Environments[EB/OL]. <http://www.inf.ethz.ch/publications>, 2003-11.

更正

本刊2006年第6期第1288页中的图10、图11有错,图10原为:



应改为:

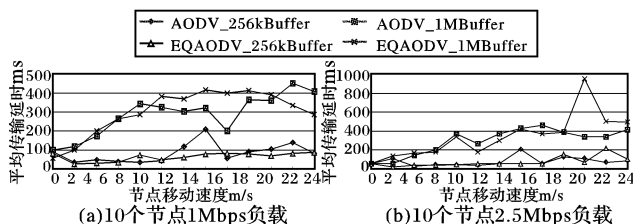
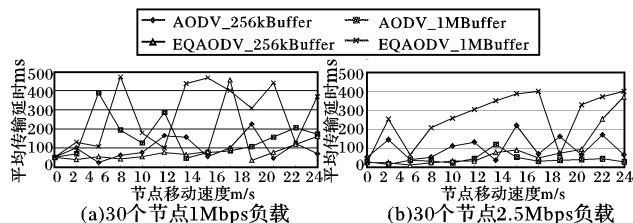
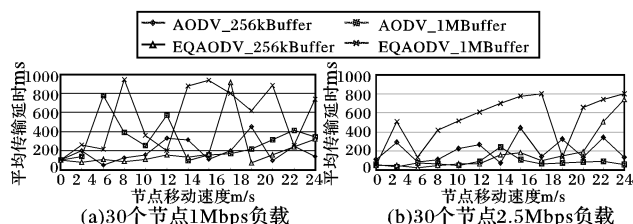


图11原为:



应改为:



本文第三作者鄢楚平的职称应为正研级高级工程师。因工作疏忽,给读、作者带来不便,谨致歉意!

《计算机应用》编辑部