

文章编号:1671-9352(2008)11-0061-06

# 基于 P2P 的异构数据库数据同步研究

孟雷<sup>1</sup>, 孙彦杰<sup>2</sup>

(1. 山东大学威海分校信息工程学院, 山东 威海 264209;

2. 山东大学计算机科学与技术学院, 山东 济南 250101)

**摘要:**企业信息化在高速发展的同时也产生了很多需求,异构数据库间的数据集成和数据同步就是其中一个亟待解决的问题。基于 extensible markup language 实现了异构数据同步模型,通过应用上的分析,设计了在传输速率和传输稳定方面改善性能的方法,从而建立了基于 P2P 的异构数据同步模型,该模型采用 extensible markup language 技术和面向服务技术屏蔽数据的异构性,通过触发器和控制表的方式实现了差异更新,并利用基于最短路径 P2P 算法实现增量提取和传输优先级,从而增强了传输的效率和可靠性。

**关键词:**数据同步; extensible markup language; P2P; 最短路径算法

**中图分类号:** TP311      **文献标志码:** A

## Research on P2P-based heterogeneous database data synchronization

MENG Lei<sup>1</sup>, SUN Yan-jie<sup>2</sup>

(1. School of information engineering, Shandong University At Weihai, Weihai 264209, Shandong, China;

2. School of Computer Science & Technology, Shandong University, Jinan 250101, Shandong, China)

**Abstract:** The existing problems in data integration and data synchronization were analyzed, especially the requirements of heterogeneous data synchronization and efficient transport. A heterogeneous database data synchronization model was given. By transferring into Extensible markup language files, the difference between heterogeneous data sources was shielded. A P2P-based optimal algorithm ensures real-time transports, which is essential to business applications. A data synchronization tool was developed and used in real applications. By comparing the system performance, it was shown that this tool can solve a heterogeneous problem and can keep high speed transmission when the network situation was not so good.

**Key words:** data synchronization; extensible markup language; peer to peer; shortest path algorithm

## 0 引言

随着信息化的发展和网络技术的普及,许多企业在业务发展中逐渐实现了计算机管理。但是由于历史或者技术原因,企业可能在不同时期使用了不同数据模型的数据库系统,而且计算应用系统也逐步从单机版走向 c/s, b/s 架构,这就造成企业内不同系统无法有效地进行交互和数据共享,从而制约了企业整体的信息化建设。异构数据集成主要是解决异构数据库之间的数据集成和数据通信问题。而

数据同步作为数据集成中的 1 个部分,又具有自身的一些特殊需求,如数据同步对时效性的要求较高,要求能够有比较好的数据传输效率,又比如数据同步中要求源数据与目标数据实施保持一致性,这些都是—些具有很高的理论和使用价值的研究课题。

目前针对数据库的同步,已经有不少的解决方案,常用的如数据库管理系统自身的数据复制技术<sup>[1]</sup>,但是这种方式对于异构数据节点,特别是不同类型的数据库系统之间,以及结构化数据库系统与文件数据之间的同步则无能为力,或者需要进行大量的配置工作,在现实应用中具有很大的局限性。

对等网络<sup>[2]</sup>(peer to peer, P2P)是一种分布式网络,网络的参与者共享他们所拥有的一部分硬件资源,这些共享资源需要由网络提供服务和内容,能被其它对等节点直接访问而无需经过中间实体。在此网络中的参与者既是资源(服务和内容)提供者,又是资源(服务和内容)获取者。P2P网络模型与C/S模型相比,具有非中心化、可扩展性、健壮性、高性能/价格比、隐私保护、负载均衡等特点,它在有效利用网络上大量的信息资源、存储空间、处理器周期等闲置资源,避免服务器带来的瓶颈问题,降低服务器成本等方面有明显优势<sup>[3,4]</sup>。

本文提出的模型从数据模式统一,传输2个方面,提出了基于 Extensible markup language 技术和 P2P 技术的解决方案,可以解决数据同步面临的差异更新困难和传输效率低等问题,灵活性很强,在实际应用中取得了比较好的应用效果。

## 1 相关研究

数据集成是对各种异构数据提供统一的标识,存储和管理,提供用户对多种异构数据源的透明,一致和实时访问。目前数据集成采用的体系结构有3种:联邦数据库系统,中间件系统和数据仓库。其中最主要的是数据仓库,该方法在用户端和数据源之间建立一个存储数据的功能层,它把多个数据源的数据进行统一转换,形成统一的模式存储。这种方法的优点是数据高度集中,客户端扩展比较容易;缺点是数据集成是单向的,而且差异更新困难。

在 P2P 发展过程中,每个阶段的划分是以搜索算法为标志的。第一阶段以 Napster 为代表的集中目录式结构,带有中心服务器,但服务器上只存放每个节点的目录信息。该搜索的优点在于能够进行快速查询。其缺点是当服务器接受过多的请求,其搜索效率会逐渐下降。第二阶段以 Kazaa 为代表的纯 P2P 模式,对等节点之间的内容查询和内容共享都是直接通过相邻节点广播接力传递。其搜索算法优点是在很短时间里搜索到的节点呈几何级数增长。其缺点是以泛洪方式进行搜索,请求信息的泛滥消耗了大量带宽并造成网络拥塞。第三代 P2P 以 eDonkey, emule 为代表,采用“分散式哈希表”<sup>[5-7]</sup>的混合网络模型,它引入了超级节点的概念。超级节点与其临近的若干普通节点构成一个簇,簇内采用基于集中目录式的 P2P 模式,而不同簇之间再通过纯 P2P 模式将超级节点相连。搜索算法的优点是有效消除泛洪算法带来的网络拥塞、搜索迟缓等缺点,

同时能确保一些恶意的攻击行为在网络局部得到控制。其缺点是超级节点本身的脆弱性可能导致其簇内节点处于孤立状态。

## 2 基于 Extensible markup language 的异构数据同步模型

首先描述一下异构数据集成模型(见图1),该模型在用户和异构数据源之间建立一个数据存储中心,用户通过接口访问数据存储中心即可访问到所有异构数据源的数据。数据同步模块负责数据存储中心和异构数据源之间数据的同步更新。

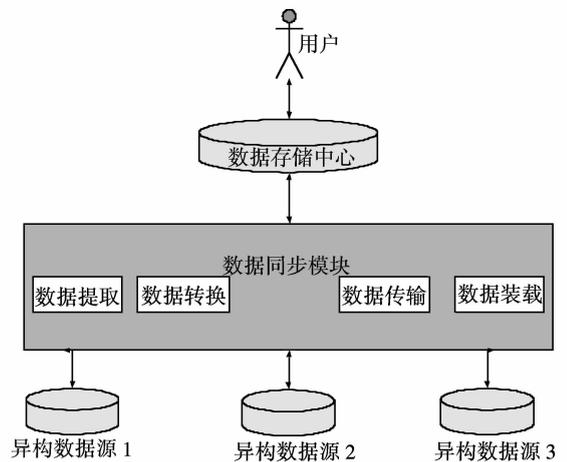


图1 异构数据源集成模型

Fig.1 Model of heterogeneous data integration

针对数据集成中的数据同步模块,提出了异构数据同步模型(DSM)(见图2),这里的异构可以有3种体现方式:平台层,系统层和数据语义层,DSM模型讨论的是系统层上的异构。该模型包括数据提取器,数据解析器两大部分。数据提取器主要用来异构数据源交互,用来取得要同步的数据。数据解析器主要用来把同步过来的数据集成到数据存储中心。

在进行数据同步时,不同的异构数据源可能是由不同数据模式构成,比如关系型,对象型等,不同数据的表现形式不一样。即使是模式相同,也可能存在数据结构和表现形式上的差异。这些模式或者结构上的差异会导致同步上的困难。因而,在同步过程中,必须将异构的数据模式统一为1个公共的标准的数据模式。DSM模型采用的是 extensible markup language 转换方式。extensible markup language 具有自描述性,平台无关性,能描述不同复杂程度的数据,可扩展性等优点,用 extensible markup language 可以建立一个异构数据的统一模式。

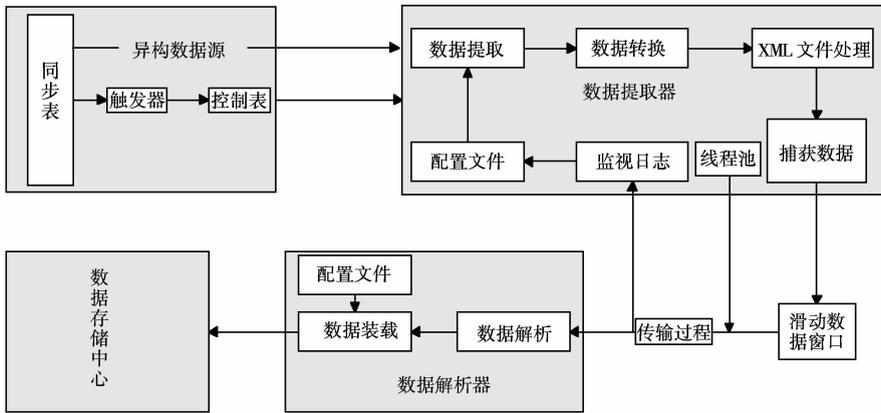


图 2 异构数据同步模型  
Fig.2 Model of heterogeneous data synchronization

模型的工作流程大致如下:

首先在异构数据源(客户端)和服务端上分别部署数据提取器和数据解析器,并根据要求配置文件。

其次,定时或者手工同步开始时,数据提取器查看控制表的信息,如果不为空,则生成 Extensible markup language 文件传输,同时系统根据配置文件里的信息,查询出同步中数据转换成 Extensible markup language 文件传输,传输过程中发生网络错误,监视日志记录下相关信息以便于错误信息重传。

最后,数据解析器收到客户端的同步数据之后,解析出 Extensible markup language 文件中的数据,根据配置文件插入到数据存储中心。

### 2.1 异构数据的配置

用 extensible markup language 可扩展标识语言定义的描述异构数据源的描述性文件。很多同步系统利用触发器和存储过程实现数据库变更事件的实时获取和 Extensible markup language 数据生成,但是如果同步的表多,配置数据库的操作就太过繁琐,如果还要实现差异更新的话,那每个同步表就需要写 3 个触发器,可见扩展性不强。而采用配置文件方式,只需要在配置文件中配置一次,就可以实现任意表的同步。配置同步表的过程中,还可以通过排同步表的先后顺序来实现传输的优先级。

数据源数据提取器中的配置文件描述了如下信息:

- (1)如何连接异构数据源;
- (2)如何从异构数据源同步表中取出要同步数据;
- (3)设置定时同步的信息;
- (4)代理服务器的 IP 地址。因为数据同步工具的服务端和数据存储中心可能是在 2 台服务器上,所以需要提供服务端所在代理服务器的 IP 地址。目标数据解析器中的配置文件描述了如下信息:

- (1)如何连接数据存储中心;
- (2)如何把解析出来的数据集集成到数据存储中心。

### 2.2 数据提取器

部署在异构数据源端的服务程序,利用配置文件中的相关信息,来实现对所在异构数据源上的数据库进行查询操作,并将查询的数据转换为 extensible markup language 文件,以及对 extensible markup language 文件处理和监视日志的维护。查询的数据有 2 种,一种是增量提取的数据,即根据配置文件里的上次同步结束时间取出数据库里新增加的数据。第二种是差异更新的数据,上次同步结束后,同步表中可能某条记录被更新或者删除,这些记录的信息由触发器记录到控制表中,下次同步时,系统读取控制表中的信息,查询出相关数据后转化为 extensible markup language 文件。数据提取器的流程见图 3。

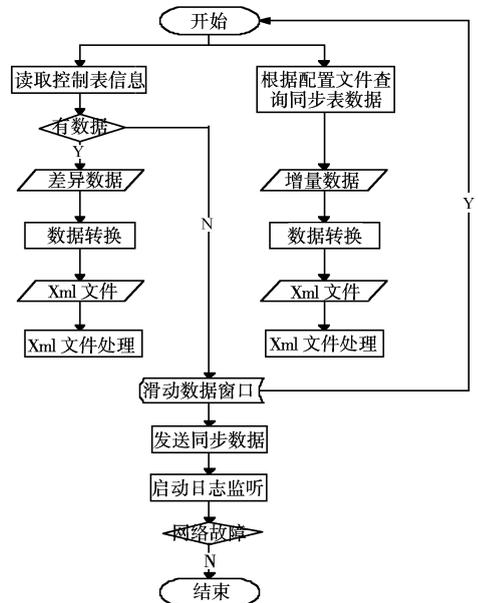


图 3 数据提取器流程  
Fig.3 Process of data extraction

## 2.4 数据解析器

部署在服务端的服务程序,收到异构数据源客户端传过来的 Extensible markup language 文件之后解析出数据,同时利用配置文件中的相关信息,把数据集成到数据存储中心(流程如图4)。

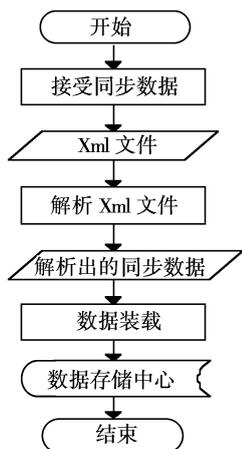


图4 数据解析器工作流程  
Fig.4 Process of data analysis

## 2.5 数据一致性控制

采用控制表的方法实现差异更新功能,在数据库里建立 manageInfo 表,设六个字段:

ID: 主键名

tableName: 表名

keyName: 主键名

keyValue: 主键值

manageTag: 更新类型

modifyValue: 更新值

同步结束后,如果某条记录发生改变,设置的触发器就会在 manageInfo 中插入一条信息,当下次同步开始时,查看控制表的信息,根据 manageTag 里的 modify 或者 delete 来形成不同的操作语句,然后采用 dom4j 技术转化成 Extensible markup language 文件用于传输。

采用日志监视器实现错误重传功能。监视日志是一个自定义格式的日志,同步数据开始传输后,传输过程中可能出现网络故障等,从而导致传输失败,这些错误都会写到日志里,系统下次同步时会再传一次,或者用户也可以选择手工传一次,这样就不会有数据漏传,保持了数据的一致性。

## 3 基于 P2P 的传输优化

### 3.1 改进数据同步效率的方法

数据同步对实时性的要求很高,特别在商务系统中,市场机会瞬息万变,要求能够在尽可能短的时

间内把数据传输到目标系统中。采用一些技术手段可以有效地提高数据传输速度,比如采用滑动数据窗口和多线程相结合的思想。

从异构数据源取出的数据经过数据转换为 Extensible markup language 文件后,使用滑动窗口可以限定一次送至网络的数据单元的个数。也可以将数据先暂存在滑动数据窗口中,一个窗口的数据传输成功后,窗口向后移动,使用这个机制有效地利用了网络带宽,在负载量很大的网络中是非常有效的。

同时,配合采用多线程的传输,线程池里装载着等待的线程,当滑动数据窗口中的数据单元已满时,用线程池中的多个线程进行传输。线程的数目可以根据要传输的数据量大小灵活的设置,这样不但节省了系统资源,还可以保证数据快速传输。

但是,在 2 个节点间的传输,毕竟还受到网络带宽和服务器自身性能的限制,基于 P2P 技术,构建传输 P2P 网络,并采用优化的 P2P 传输算法,可以更好地解决传输效率问题。

### 3.2 基于限速的 P2P 传输算法

在点对点传输中,为防止数据发送节点因为忙于数据传输和建立连接而崩溃,当发送节点连接的外部节点过多时,就需要限制每个连接的传输速度。在这里根据当前连接数量,动态调整传输速度,具体思想为:在服务端节点用一个全局变量记录当前建立连接的数量,根据连接数量和节点的传输能力确定每发送一个数据包后的等待时间。连接数量越多,表明节点繁忙,则等待时间也越长,每个节点的传输速度降低;当连接数量减少后,节点的压力减小,等待时间也会自动减少,可以加快传输速度。其最终目的是保证系统提供正常服务的前提下,尽可能地达到节点传输能力的上限。

在实现过程中,首先确定节点的传输能力上限。假设服务节点对外只有一个连接且不限速时,  $F_{\max}$  代表每秒最大传输流量(单位字节)。那么在服务节点同时建立多个连接以后,其每秒的最大传输流量都不会超过  $F_{\max}$ , 因为建立多个连接引发上下文切换会消耗一定时间,另外多个连接由于每发送一个数据包后都会有等待时间,这就无法保证每个时间片都有发送数据的操作。因此可以用  $F_{\max}$  代表系统每秒最大的传输流量。假设服务节点同时对外的连接数为  $N$ , 每个数据包传输的字节数为  $P$ (单位字节),可以使用如下公式确定每发送一个数据包后等待的时间(单位为 ms):

$$t = 1000 / (F_{\max} / (N * P)) = \frac{1000 * N * P}{F_{\max}}$$

该公式假设节点每秒的传输流量为  $F_{\max}$ , 这样每个连接在 1 s 内的发送数据量为  $F_{\max}/N$ , 进而每个连接在 1 s 内需要发送  $F_{\max}/(N * P)$  个数据包, 最后得到每 2 个数据包之间的时间间隔。该公式保证了节点在提供正常服务的前提下, 尽可能地达到节点传输能力的上限。

在实际应用中每个数据包传输 48 KB, 测试发现, 系统的  $F_{\max}$  为 4 500 KB, 当前连接线程数和等待时间关系见表 1。

表 1 线程数和等待时间

Table 1 Contrast of the number of threads and the waiting time

线程数	等待时间/ms
1	10.7
2	21.3
3	32.0
4	42.7
5	53.3
6	64.0

通过测试发现, 与不限速相比, 使用上面的方法, 可以使服务端同时连接的线程数明显增多, 增强了节点的健壮性。达到了服务端动态负载均衡的目的。

### 3.3 基于最短路径的 P2P 传输优化算法

上述算法讲述了如何计算出一个节点传输数据包要等待的时间, 可以进一步利用这个等待时间作为权值, 通过最短路径的算法求得一条权值最小的路径(即最短路径), 然后再进行传输, 这样不但提高了速度, 而且有效地利用了带宽, 减少了网络负载。以下是算法描述:

假设:

节点有  $m$  个待传输的数据包, 在每个传输时刻只能传送一个数据包;

目标值: 求出节点传输完这  $m$  个数据包的总等待时间。

术语:

传输时刻: 选择一个数据包发送出去的時刻;

$P$ : 传输节点的可用性概率。在 P2P 网络环境下, 经常会发生节点失效, 失效的原因主要是用户退网或者网络中出现路由错误等;

$K$ : 数据包此时能用此节点传输的概率。在 P2P 传输中, 有大量的数据包会传到某节点上, 比如说洪泛算法就会产生大量的数据包, 其中有些数据包会因为生命周期等原因已经失效, 也有可能因为优先级的原因要提前或者退后发送;

$M$ : 节点上共有  $m$  个待传输的数据包;

$T$ : 传输一个数据包平均等待时间。

模型:

在第一个传输时刻, 第一个数据包被发送的概率是  $p_1 = p * k$ , 第二个数据包被发送的概率是  $(1 - p_1) * p * k$ , 那么第  $n$  个数据包被发送的概率是  $(1 - \sum_{i=1}^{n-1} p_i) * p * k$ 。  $\sum_{i=1}^{n-1} p_i$  表示数据包被发送出去的概率,  $1 - \sum_{i=1}^{n-1} p_i$  代表数据包没被发送出去的概率, 可以看出数据包是否被发送符合 0-1 分布, 如表 2 所示。

表 2 数据包发送的 0-1 分布

Table 2 The 0-1 distribution of packets sending

是否被发送	1	0
概率	$\sum_{i=1}^n p_i$	$1 - \sum_{i=1}^n p_i$

定义  $Y(P, K, m, t)$  为第  $m$  个数据包要等待的时间,  $F(P, K, m)$  为第  $m$  个数据包被发送的概率。

$$Y(P, K, m, t) = t * \sum_{i=1}^m f(P, K, i). \quad (1)$$

假如定义:

$$A^i = \sum_{i=1}^i p_i, \quad (2)$$

则得到:

$$f(P, K, 1) = A^n. \quad (3)$$

当第二个传输时刻时,  $f(P, K, 2)$  有 2 种可能性, 第一个传输时刻成功发送出数据包, 或者没有成功发送数据包。即:

$$f(P, K, 2) = f(P, K, 1) * A^{n-1} + (1 - f(P, K, 1)) * A^n. \quad (4)$$

类似推理, 第  $m$  个传输时刻

$$f(P, K, m) = f(P, K, m-1) * A^{n-m+1} + (1 - f(P, K, m-1)) * A^{n-m+2}. \quad (5)$$

把公式(5)带入公式(1)中即可以得到求一个节点的权值的公式。

上面的算法可以求得一个节点的权值, 这个值可以用来衡量一个节点的传输能力, 权值越少, 传输能力越好, 点对点传输时, 假如每个负责的传输节点都有这样一个权值, 就可以在发送点执行 Dijkstra 算法, 得到一个总权值最少节点的集合, 传输的时候按照这个路径来传输, 将得到最快传输速度以及最佳网络带宽应用。

## 4 实现与应用

异构数据同步模型(DSM)已经被应用于实际的异构数据集成架构中。在应用场景中(图 5), 需要将来自不同主机厂的数据同步到供应商业务管理系

统(SBM)中,需要的同步系统2端之间分布于不同的城市,且数据格式和数据库管理系统均是异构的,需要同步的数据是企业的生产计划、订单等业务信息,实时性要求很高,数据首先通过数据访问服务(DAS)的方式集成到数据集成服务(DIS)中,SBM通

过调用DIS服务获取相关数据,在业务层次上提供了一个信息共享服务;然后,供应商可以登录SBM的服务界面,检索信息和执行业务,避免了供应商需要分别查询多个主机厂信息的繁杂重复工作,加快了供应商的供货流程。

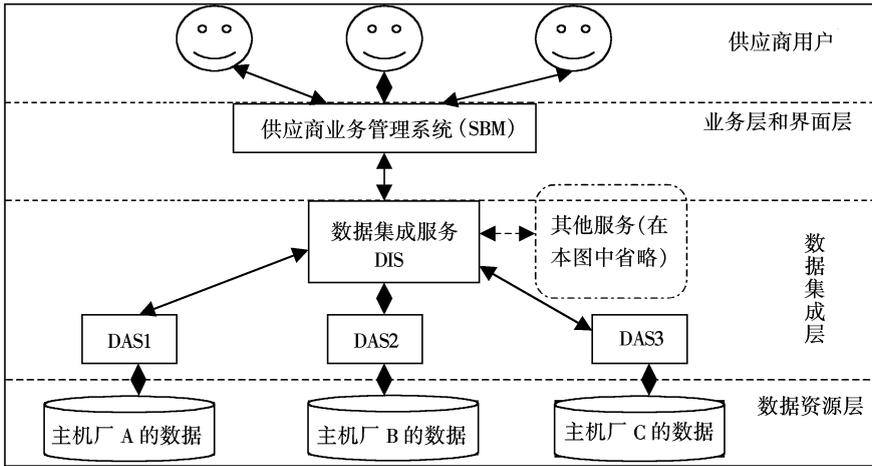


图5 数据集成在供应商业务管理系统中的应用

Fig.5 The role of data integration supplier in the business management system

从实际应用来看,文章中提出的异构数据同步模型可以成功地解决异构数据源集成的问题,该模型不但可以用于不同类型数据库系统之间的同步,还可以用于非结构化数据的同步,比如文件和图片等。在服务器端和客户端分别安装数据解析器和数据提取器。通过配置界面(图6),实现异构数据集成的灵活设置和部署,具有很好的动态性和扩展性。客户端主要配置如何连接异构数据源,如何从异构数据源同步表中取出同步数据,设置定时同步的信息,代理服务器的IP地址。服务器端主要配置如何连接数据存储中心以及如何把解析出来的数据集成到数据存储中心。配置的信息会保存在 extensible markup language 格式的文件中。服务器端和客户端配置都完成之后,可以在客户端的图形化页面中配置定时信息,到达某个设定时间同步就会自动进行。

在传输效率方面,从分布于2个城市之间异构系统间执行同步,采用定时同步的方式时,每天同步24次,数据量在5000条记录左右;采用即时同步时,单次同步数据量在6000条左右,峰值数据量在20000条左右;均能够实现实时的同步传输,经过近一年的实际运行,显示了数据集成服务的稳定性和令用户满意的效率。

异构数据同步模型(DSM)的第二个应用是在山东省制造业信息化服务平台地区节点的建设中。每个地区定制 portal, portal 部署在当地的服务器上,为了让地区 portal 有主题和地域特色,总的 portal 会产生包括各个地区的 portal 数据,每天定时同步到地区 portal 中,同时地区节点新产生的数据也会每天同步到总 portal 中。

目前异构数据同步模型已经在总 portal 和潍坊 portal 之间实现了双向同步,同步的数据包括结构化的数据和非结构化数据,如文件图片等。每天定时同步2次,同步的数据库表有109个以及大量的文件图片等,从运行情况来看,效率和稳定性都达到了用户的要求。

### 5 总结与展望

本文提出的异构数据库同步模型(DSM)解决了现实中关于数据同步的异构性和传输效率的要求。功能比较完善,灵活性强,配置也 (下转第71页)

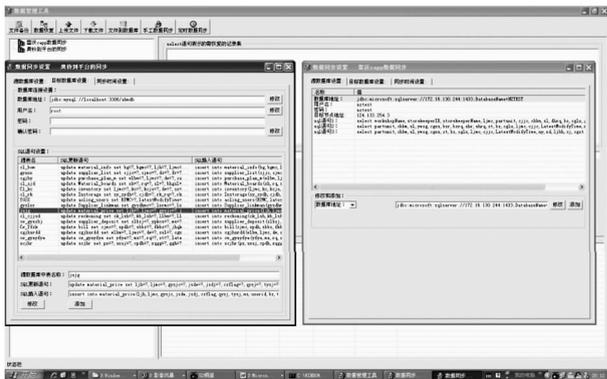


图6 数据同步工具配置界面

Fig.6 Configuration interface of data synchronization tool

(上接第 66 页) 比较简洁方便,并已成功地运用到面向服务的异构数据集成中,并取得了很好的效果。但是也存在一些问题,比如在定时同步中,目前是通过补偿机制实现数据的弱一致性,在点对点传输的优化算法中,目前只实现了基于静态权值的优化,没有考虑节点权值的动态性。这些有待于在今后的工作中进一步优化。

#### 参考文献:

- [1] 刘智勇. Sql Server 2005 宝典[M].北京:电子工业出版社, 2005.
- [2] SING Li. Making P2P interoperable: the JXTA command shell [M]. Birmingham: Wrox Press, 2001.
- [3] Robert Flenner, Michalel Abbott. Java P2P unleashed[M]. [S.1]: Sams, 2003.
- [4] Jameela AlJaroodi, Nader Mohamed, Hong Jiang. Agent-based parallel computing in Java proof of concept[J]. Technical Report TR-UNL-CSE-2001-1004, Summer, 2001: 5-12.
- [5] 包林玉,袁平,彭春燕,等. 基于 JMS 和 XML 的异构数据集成技术的设计与实现[J]. 电脑知识与技术, 2008, 13: 684-685.
- [6] 王辉等,李贤媚,周英梅. 异构数据库间主动式差异数据同步模型研究[J]. 河南大学学报: 自然科学版, 2008 (6): 1672-6871.
- [7] 李铁军,郝文宁,王伟一. 一个轻量级的异构数据库同步系统[J]. 计算机与信息技术, 2007, 6: 65-66.

(编辑:孙培芹)