第16卷 第11期
2006年11月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol. 16 No. 11
Nov. 2006

用JDOM和XML实现异构数据库的数据提取

周 颖,王义发
(武汉理工大学 自动化学院,湖北 武汉 430070)

摘 要:大量分散的形式及不同格式的数据给现代数据处理带来了越来越大的困难。为统一数据形式以利于数据操作和处理,讨论了将形式多样的数据格式转换成统一的XML(Extensible Markup Language)格式的问题。对数据源中不同格式文件数据,按照预先定义的XML模板,以格式说明文件结构统一描述,并提取数据或作进一步的处理,最后转换为XML格式输出。文中论述了从数据库中提取数据转换为XML格式的方法及步骤,并且方法简单实用,可以推广到对所有格式数据的提取。

关键词:XML;JDOM;数据提取
中图分类号:TP311.13 **文献标识码**:A **文章编号**:1673-629X(2006)11-0125-03

Extracting Data from Heterogeneous Database with JDOM and XML

ZHOU Ying, WANG Yi-fa
(Sch. of Automation, Wuhan Univ. of Tech., Wuhan 430070, China)

Abstract: The largely dispersive and different forms of data have resulted in more and more difficulties to the modern data processing. In order to unify the data form to help data operating and processing, the issue how to transform the various data forms into unified XML (extensible markup language) format is discussed. For the file data of different forms in the source data, according to the predefining XML cyclostyle, with unified specification in the structure of format specification file processing, and finally outputting the transforming into XML format. Also has extracted the data from the database to XML form, the method and the step of which is simple and practical, and might be promoted to all forms data's extracting.

Key words: XML; JDOM; extraction of data

0 引 言

在信息时代,特别是随着网络的发展,每天都会有海量的数据产生。在这些海量数据处理过程中,数据间的异构是数据处理过程中最大的障碍。数据可以有很多存在格式,例如txt文本格式、PDF格式、word格式、excel格式以及存储在数据库中的数据格式。其中,数据库是海量数据存在的主要形式。而数据库之间仍然存在异构,称之为异构数据库[1]。异构数据库系统是相关的多个数据库系统的集合,可以实现数据的共享和透明访问,每个数据库系统在加入异构数据库系统之前本身就已经存在,拥有自己的DMBS。异构数据库的各个组成部分具有自身的自治性,实现数据共享的同时,每个数据库系统仍保持自己的应用特性、完整性控制和安全性控制。

异构数据库系统的异构性主要体现在3个方面:计算机体系结构的异构、基础操作系统的异构、DBMS本身的异构。

异构数据库系统的目标在于实现不同数据库之间的数据信息资源的合并和共享。实现方式就是每个异构数据库间通过一个中间件进行数据的交换。一个异构数据库系统的框图如图1所示。

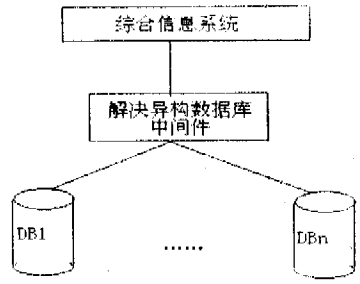

图1 异构数据库应用模型

从异构数据库应用模型中,可以看出中间件的实现是整个系统的关键。对于中间件的实现,采用XML和Java的相关技术实现其中最为重要的数据提取。

可扩展标记语言(XML, eXtensible Markup Language)[2,3]是一种简单灵活的文本格式的可扩展标记语言,起源于SGML(Standard Generalized Markup Language),是SGML的一个子集合,也就是SGML的一个简化版本,非常适合于在Web上或者其它多种数据源间进行数据的交换。它提供了一套跨平台、跨网络、跨程序语

收稿日期:2006-02-24
作者简介:周 颖(1973-),女,湖北武汉人,博士研究生,研究方向为远程监控与智能故障诊断、计算机网络与信息系统。

言的数据描述方式，使不同系统之间的数据交换更加高效。毋庸置疑，它是数据交换的一种标准格式[2]。同样跨越平台限制的 Java 利用 JDBC 接口技术实现对不同数据库的访问，利用 JDOM 进行 XML 的操作。对于 XML 和 Java，前者提供规范化的数据描述，后者提供处理数据的代码，进行不同数据库数据的提取工作。它们的完美结合不仅可以提取数据，还可以进行数据交换、传输，是 Web 发展的主流方向。

数据的提取就是把异构数据库当中不同表现格式的数据抽取出来并转换为统一格式，即 XML 文档格式。数据提取过程可简单描述如图 2 所示。

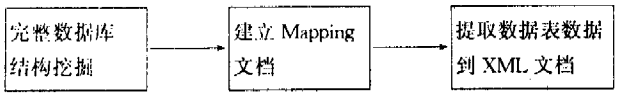

图 2 数据提取流程图

1 通过 JDBC 访问数据库

JDBC 是 JavaSoft 的一个用于连接数据库、操纵数据库和编写 JDBC 数据库驱动程序的规范。JDBC 允许 Java 应用程序和不同数据库连接。JDBC 驱动程序一般有 4 种类型，这里采用第二种类型：部分 Java、部分本机驱动程序。这种驱动程序使用 Java 实现与数据库厂商专有 API 的混合形式来提供数据访问。JDBC 驱动将标准的 JDBC 调用转变为对数据库 API 的本地调用。框图如图 3 所示。

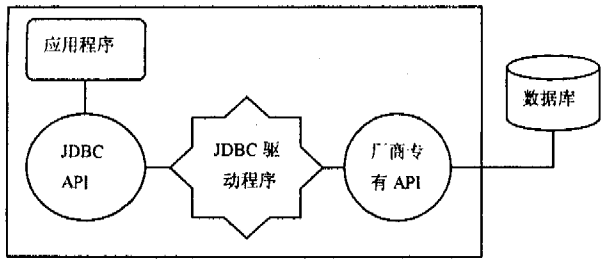

图 3 JDBC 与数据库的连接方式

其操作过程如下：

1) 装入驱动程序。

2) 建立程序到数据库的连接。

3) 创建一个 Statement 对象，该对象实际执行 SQL 语句或存储过程。

4) 创建一个 ResultSet，然后用执行查询的结果填充。

5) 从 ResultSet 检索或更新数据。

2 利用 JDOM 技术生成 XML 文档

2.1 DOM 和 SAX

DOM 是处理 XML 文档结构的一种接口[4]。作为一个 W3C 项目，DOM 的设计目标是提供一组对象和方法，使程序员的工作更轻松。理想情况下，应该能够编写一个应用程序使用一种 DOM 兼容的解析器处理 XML 文档，然后切换到另外一种 DOM 兼容的解析器而无需改变代码。DOM 模型提供的树形结构正好能体现 XML 文档中各元素的层次关系，其接口函数也能方便地对各个元素的内容进行添加、删除和修改等操作。但由于 DOM 在分析 XML 文档时，必须将整个文档都读入内存，并且 DOM 的树结构占用内存较多，致使 DOM 处理大型文档时其性能下降得非常厉害，也要花费较长的时间。

SAX 最初是由 David Megginson 采用 Java 语言开发的，之后 SAX 很快在 Java 开发者中流行起来。所有的 SAX 处理都在一次遍历中完成，采用事件触发的方式，编程人员只需要编写事件处理程序。解析同等大小的文档时 SAX 通常会比 DOM 提供更好的性能。此外，与 DOM 相比，因为在给定的时间之内只需要 XML 文档的一部分装入内存，所以 SAX 通常在处理更大文档时内存的利用效率更高。

2.2 运用了 80－20 法则的 JDOM

针对文档对象模型的复杂性，人们提出了另外一种解决方案，即 JDOM。它由 Brett McLaughlin 和 Jason 所创建，是一个开源项目，它基于树型结构，利用纯 Java 的技术对 XML 文档实现解析、生成、序列化以及多种操作。

JDOM 使用 80－20 法则为最常用的 80% 的 XML 处理功能提供一种简单的 API，即“使用 20%（或更少）的精力解决 80%（或更多）Java/XML 问题”（根据学习曲线假定为 20%）[4]。

JDOM 直接为 Java 编程服务。它利用 Java 语言的诸多特性（方法重载、集合概率以及映射），把 SAX 和 DOM 的功能有效结合起来，以弥补 DOM 和 SAX 在实际应用中的不足。DOM 的缺点主要由于 Dom 是一个接口定义语言（IDL）[5]，它的任务是在不同语言实现中的一个最低的通用标准，并不是为 Java 特别设计的。SAX 的不足主要在于 SAX 没有文档修改、随即访问以及输出的功能，相对 DOM 来说，在使用时总觉得不太方便。

JDOM 是用 Java 语言进行读、写、操作 XML 的 API 函数，是 Java 平台专用的，其对象就是像 Document、Element 和 Attribute 这些类的直接实例，因此创建一个 JDOM 对象就如在 Java 语言中使用 new() 操作符一样容易。它还意味着不需要进行工厂化接口配置——JDOM 的使用是直截了当的。使用 JDOM 生成 XML 文档的一般过程是这样的：

1) 解析 Mapping 文档，得到需要的各个 Element 元素和 Attribute 元素。

2) 使用 new() 生成相应的 Element，Attribute 对象。

3) addAttribute() 添加 Attribute 子元素；addContent() 添加 Element 子元素。

4) setText() 设置每个 Element 的 Text 内容。

5) 用根元素作为参数，生成 Document 对象。

6) 输出 XML 文档。

生成的 XML 文档结构如图 4 所示。

在图 4 树结构中，椭圆表示各个 Element 的属性，通过 addAttribute() 方法加载进 Element 中，下级子元素通过

addContent()方法加载到父 Element 当中,这样一层一层地加载,最后用 root Element 作为参数生成最顶层的 Document 对象,完成整个 XML 文档。

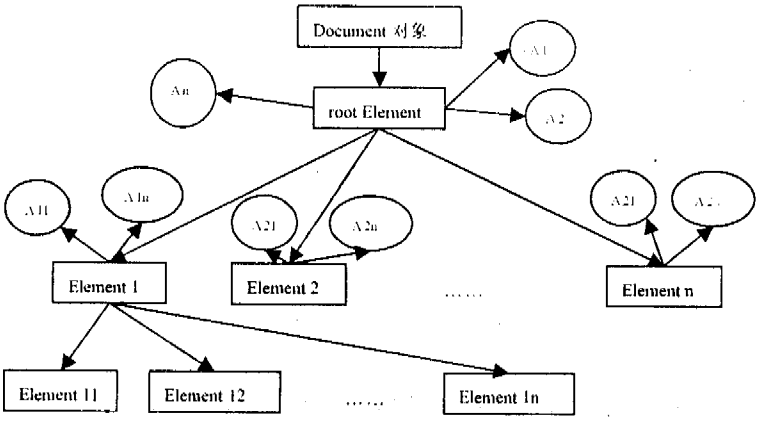

图 4　XML 文档树型结构图

3　用 XML 描述数据库结构

在数据库中,关系表是数据的存储区。表内结构及表与表之间的各种关系和约束,称之为数据库的结构。提取数据库的数据是为了以后的传输及再存储,首先得描述清楚数据库的结构,以备再存储时保证数据的完整性和一致性。

数据库结构描述的内容分为库描述、表内描述和表间描述。库描述主要是这个数据库的安全权限设置的描述;表内描述就是描述表信息,比如表名、字段类型、字段长、主键、字段是否可为空等;表间描述就是表间关系描述和表间约束描述等。而数据库结构描述的方式可以有以数据为中心的描述和以文档为中心的描述两种。以数据为中心的描述就是用 XML 的 Element 的 Text 或 Attribute 描述数据库;以文档为中心的描述就是输出整个数据库创建时的 SQL 语句。以数据为中心的描述适合异构性较大的数据库间的数据交流;而以文档为中心的描述对于异构性不大或是同构数据库来说是很方便的。可以如下简述数据库结构描述:

数据库结构描述的内容{库描述;表内描述;表间描述}

数据库结构描述的方式{以数据为中心的描述;以文档为中心的描述}

4　建立映射 Mapping 文档

Mapping 文档是 XML 文档,决定从数据库中提取哪些数据,以及这些数据在 XML 文档中的表现形式。Mapping 文档是生成数据文档的重要依据,它可以是个临时 XML 文档,根据这个临时 XML 文档建立最终想提取出来的 XML 文档,即表格数据。

Mapping 文档中,有<table>元素指示的是针对哪个表;<data sql>元素即一个 SQL 语句表示要提取的数据;<root>表示将要生成的数据文档中<table>节点的一级子节点;各个<element>的值表示二级子节点了,也是所要提取的字段名。

5　将从数据库中提取的数据写到数据 XML 文档

解析 Mapping 文档,得到<data sql>元素内容,使用 JDBC 从数据库中得到需要提取的数据表数据。根据 Mapping 文档还可以得到数据 XML 文档中各个 Element 的名称。具体生成数据 XML 文档的过程可见 2.2 节。

XML 文档对于人来说是很直观也很容易理解的,但机器是如何理解并解析文档的呢?在正式的 XML 文档中必须引入对 DTD (Documents Type Definition)或 XML Schema 的声明。DTD 和 XML Schema 都是对 XML 文档中的元素(ELEMENT)和属性(ATTRIBUTE)的定义。但由于 DTD 不支持多种多样的数据类型,扩展性较差,所以在描述数据库结构方面存在很多缺陷,而 XML Schema 是定义严格的 XML 文档,方便 XML 解析器的处理,也符合今后的发展趋势,所以使用 XML Schema 进行 XML 文档类型定义是更好的选择。

解析 XML 文档的过程如图 5 所示。

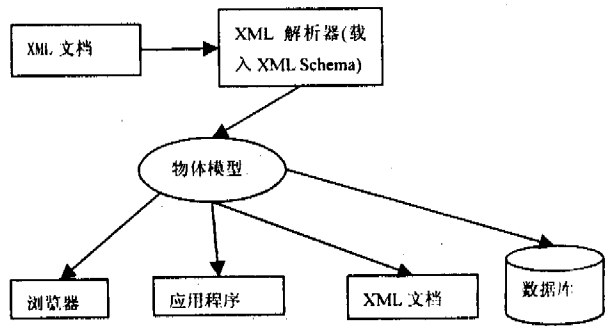

图 5　XML 文档解析过程

6　结束语

文中讨论了从数据库中提取数据的实现过程及相关技术,实际上,在数据库之外也存在大量有用并需要提取出来的数据,在那里 XML 同样可以发挥巨大的作用。文中就数据库数据的提取作了相关的研究,对于数据库以外数据格式的提取,也可以参考相关步骤方法,把各种格式的数据转换成 XML 文档,这样便于在网络上传输,最终完成整个 Internet 或 Intranet 大环境下异构数据间的转换。

参考文献:

[1]　杨　红,田富鹏,王礼刚. Java 和 XML 实现异构数据库环境下的数据抽取[J]. 西北民族大学学报,2004,25(53):52-56.

(下转第 130 页)

⑦if((c_λ(T) = c_λ(T_delay))or(c_λ(T) = c_λ(T_cost)))then return T_delay;

⑧else if(delay(T) ≤ Δ)then T_delay = T;

⑨else T_cost = T;

⑩ 转向 ⑤。

● 定理3 算法 LR－DLMR 在最坏情况下的复杂度为 $O(p(m+3/2)n^2-pm^2n)$，n 为网络中的节点数，m 为目的节点数，p 为 MCTH 算法的执行次数(松弛次数)。

证明：由于求得最小时延树算法的复杂度为 $O(n^2)$，MCTH 算法的复杂度为 $O((m+3/2)n^2-m^2n)$，所以 LR 的复杂度为 $O(p(m+3/2)n^2-pm^2n)$。因此，算法 LR－DLMR 在最坏条件下的复杂度为 $O(p(m+3/2)n^2-pm^2n)$。

3 实验仿真与分析

网络模型采用 Waxman[6] 提出的随机图模型，该模型产生的随机图与真实网络比较接近。随机图的节点随机分布在矩形区域内，随机图中边存在的概率为 $P_e(u,v)=\beta\exp[-\frac{d(u,v)}{\alpha L}]$，其中 $d(u,v)$ 为节点 u 到节点 v 的欧氏距离，L 为节点间的最大欧氏距离值，较小的 α 值将增大短链路的密度，较大的 β 值将导致较高的链路密度。α，β 为区间(0,1] 的实数，在实验中 $\alpha=0.2$，$\beta=0.4$，源节点和目的节点在图中的节点集中随机选择。链路代价值正比于链路距离，链路时延值取(0,1) 内的随机数的 10 倍。仿真情况定为时延限制为 300，组播节点数为 20。由于 KMB 算法生成组播树时只考虑到代价，LD 算法只考虑到时延，分别以这两种算法作为参照系的超出的代价百分比和时延百分比，得到结果如图 1、图 2 所示。

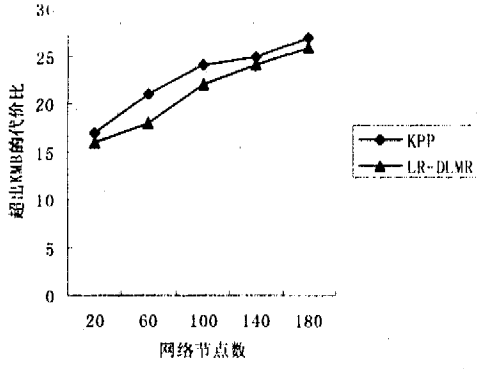

图 1 网络节点数和超出 KMB 代价比关系

由图 1、图 2 可以看出，文中算法在代价和时延性能方面都优于 KPP 算法，而且性能比较稳定，主要在于该算法并没有像 KPP 算法那样构建原网络的封闭图，从而有效地利用了链路中间节点的网络信息，以及拉格朗日松弛算法解决 NP 问题的良好性能。

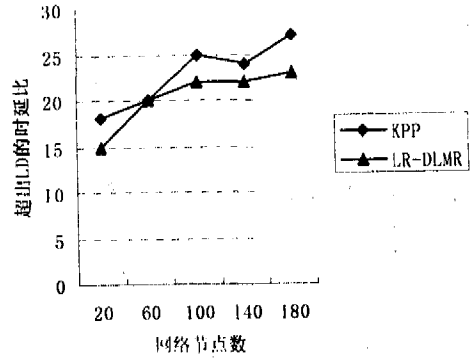

图 2 网络节点数和超出 LD 时延比关系

4 结 论

文中针对受限组播路由问题，给出了时延受限的组播路由模型，在利用拉格朗日松弛法的基础上，对其进行松弛，提出了一种基于拉格朗日松弛法的时延受限的低代价组播路由算法 LR－DLMR。该算法很好地利用了链路中间节点的网络信息，且其复杂度仅为 $O(p(m+3/2)n^2-pm^2n)$，通过实验仿真得出，该算法代价性能良好、稳定，算法时延低、速度快。

参考文献：

[1] Kompella V P, Pasquale J C, Polyzos G C. Multicasting routing for multimedia communication[J]. IEEE/ACM Tans on Networking, 1993, 1(3): 286－292.

[2] Kou L, Markowsky G, Berman L. A fast algorithm for Steiner trees in graphs[J]. Acta Informatica, 1981, 15(2): 141－145.

[3] 王明中，谢剑英. 时延抖动限制的最小代价多播路由策略[J]. 计算机学报，2002，25(5)：534－541.

[4] 杨 明，谢希仁. 一种快速的近似最小代价多播路由算法 MCTH[J]. 东南大学学报，1999，29(3)：95－99.

[5] 邢文训，谢金星. 现代最优化计算方法[M]. 北京：清华大学出版社，1999：247－290.

[6] Waxman B M. Routing of multipoint connections[J]. IEEE J on Selected Areas in Communications, 1988, 6(9): 1617－1622.

(上接第 127 页)

[2] Bray T, Paoli J, Sperberg－McQueen C M. Extensible Markup Language (XML) 1.0. W3C Recommendation [EB/OL]. 1998－02－10. http://www.w3.org/TR/1998/REC－xml－19980210.html.

[3] Box D, Skonnard A, Lam J. Essential XML[M]. 卓栋涛，译. 北京：中国电力出版社，2000.

[4] 瞿裕忠，张剑峰，陈 峥，等. XML 语言及相关技术综述[J]. 计算机工程，2000，26(12)：4－6.

[5] van der Aalst W, van Hee K. WorkFlow Management, Models, Methods and Systems[M]. Cambridge: The MIT Press, 2002.