第33卷 第15期 计 算 机 工 程 2007年8月
Vol.33 No.15 Computer Engineering August 2007
·软件技术与数据库· 文章编号：1000—3428(2007)15—0058—03 文献标识码：A 中图分类号：TP 311.13

基于面向对象 XML 的集中式和分布式存储模型

张晓琳[1]，丁 红[2]，谭跃生[1]，王国仁[3]
（1. 内蒙古科技大学网络中心，包头 014010；2. 鲁东大学物理与电子工程学院，烟台 264025；
3. 东北大学信息科学与工程学院，沈阳 110004）

摘 要：针对面向对象的 XML 数据，提出了 2 种存储模型。在分布式存储中，对象的一个属性或元素，如果是从超类继承的，其属性值或元素值存放在超类对应的对象中，如果是该类新定义的属性或元素，则放在该类对应的对象中。在该集中式存储中，对象的所有属性值都存放在所属类的对应对象中，超类中没有子类实例的数据，只有直接实例数据。通过典型查询语句，测试、分析了 2 种存储模型的性能，实验结果表明两种存储模型的可行性和高效性。

关键词：面向对象的 XML；分布式存储模型；集中式存储模型

Centralized and Distributed Storage Model Based on Object-oriented XML

ZHANG Xiao-lin[1], DING Hong[2], TAN Yue-sheng[1], WANG Guo-ren[3]
(1. Network Center, Inner Mongolia University of Science and Technology, Baotou 014010; 2. School of Physics & Electronic Engineering, Ludong University, Yantai 264025; 3. School of Information Science & Engineering, Northeastern University, Shenyang 110004)

【Abstract】 This paper proposes centralized and distributed storage models for object-oriented XML. As for distributed storage model, an attribute or element inherited from superclass resides in object of superclass, an attribute or element defined oneself resides in its own object. As for centralized storage model, all attributes and elements reside in their own object, there is not any subclass instance but direct instance in superclass object. The storability of the two models is analyzed by some typical queries. The experimental results show that they are effective and feasible.

【Key words】 object-oriented XML; distributed storage model; centralized storage model

1 概述

XML正成为网络数据描述和交换的标准。面向对象的方法具有很强的建模能力，例如继承、非单调继承、多态性、复杂数据结构等。将面向对象的特征引入到XML中，可以提高XML语言的建模能力。可用继承扩展XML的模式语言DTD支持元素继承、非单调多重继承、重载、阻断、多态和冲突处理机制[1]。XML-RL是基于高级数据模型的、以规则为基础的XML查询语言，可用面向对象的特征扩展XML-RL支持多态元素、多态引用、包含元素、包含引用[2]等。

根据XML文档本身的特点，XML存储方法有 2 种：关系数据库方式[3]和Native XML数据库方式[4,5]。

Native XML 数据库在物理上存储 XML 数据有 3 种方式：

(1)平坦的流方式(flat streams)，又称为基于文本的方式(text-based)，将 XML 数据转换为字节流。这种方式将 XML 文档转换为字节流，将其存储在文件系统的文本文件中或存储为数据库的 BLOB 字段中。

(2)建立模型方式，又称为基于模型的方式(model-based)，按照某种(物理)模型存储 XML 文档，如 DOM 或它的变体。这种方案能够以一种比较自然的方式来存储 XML 数据。

(3)混合型方式(mixed)，综合了方式(1)、方式(2)的特点。

在物理存储中，存取的最小单位是记录，每个记录都有自己的ID，在决定存储方案时，就要考虑记录与节点的对应关系，即记录的粒度。记录的粒度[5]有以下 3 种：

1)节点级：每一个节点对应一个记录。

2)子树级：一个子树对应一个记录。这种方法的关键在于如何划分子树，可根据物理块大小、逻辑意义进行划分。

3)文档级：将整个文档作为一个记录。

粒度的记录组织方式各有特点。记录的粒度越小，记录数目就越多，从而降低了记录的存储效率；但是粒度小的记录在重组文档时可以避免不必要的转换和解析。记录的粒度越大，记录表示的节点就越多，在构造记录、向记录中插入节点和分裂记录时就越麻烦。本文针对面向对象的XML数据，采用了基于模型的Native XML数据库存储策略。节点的粒度是子树级，根据逻辑意义划分子树。根据用继承扩展的DTD[1]信息，设计并实现了 2 种存储模型：分布式存储模型和集中式存储模型。

2 面向对象 XML 的存储模型

用一棵有序的标识树模拟 XML 文档，节点对应元素，边表示元素间的包含关系。

用三元组(id，label，[value])表示一个节点，id 表示节点标识符，id 采用前序遍历递增的生成方法；label 表示标识名；value 是可选的，表示节点的值。图 1 显示一个 XML 数据树实例。

基金项目：内蒙古自然科学基金资助项目(200508010808)
作者简介：张晓琳(1966－)，女，教授、博士，主研方向：对象数据库，XML 数据库；丁 红，副教授、硕士；谭跃生，教授；王国仁，教授、博士生导师
收稿日期：2006-09-05 **E-mail：**zhangxl@imust.cn

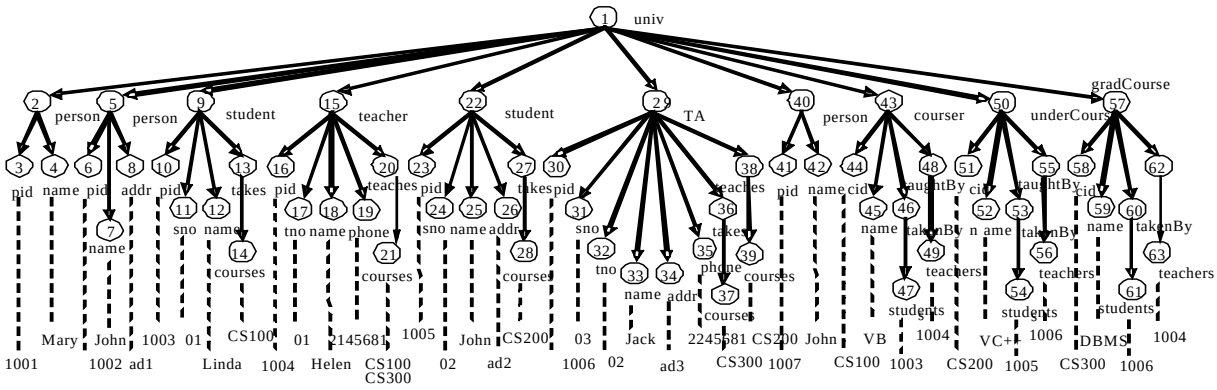

图 1　一棵 XML 数据树实例 T1

2.1　分布式存储

2.1.1　路径仓

路径仓是一棵能够紧凑、准确地表示面向对象的 XML 数据树，是两级双向树：组级和元素级。组级提供路径信息、类层次信息，元素级保存从孩子元素指向父亲元素的信息。类层次即该类及其所有直接和间接子类。类层次索引的定义为：在以索引类为根的类层次子树上存放的该类特有的元素和属性对象 id。而继承的元素和属性的对象 id 存放在较高的层次中。图 2 是图 1 XML 数据树的分布式存储的路径仓。

表 1　分布式数据仓中类节点的存储结构

内容	18B	8B	4B	ID	4B	#PCDATA	4B	#PCDATA
结构	H	SupC	H	Type	H	Type	H	Type
			pid		name		addr	
	person							

表 2　对象节点的存储结构

内容	18B		4B	4B	4B	4B	2B	1002
结构	H		P	L	H	P	L	C
					pid			
	person							
内容	4B	4B	2B	John	4B	4B	2B	ad1
	H	P	L	C	H	P	L	C
	name				addr			
	person							

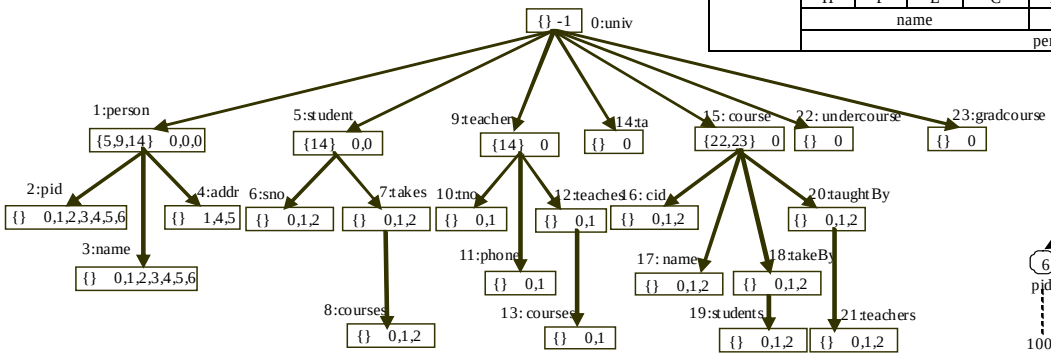

图 2　XML 数据树 T1 的分布式路径仓

1 个 XML 子树片断结构如图 4 所示。

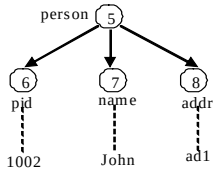

图 4　1 个 XML 子树片断

2.1.2　数据仓

图 3 显示了图 1 XML 数据树 T1 的分布式存储数据仓的二叉树结构。数据仓中的节点能存储类或实例对象，存储类的节点被连接到根节点的左指针，类节点的右指针指向每个相应类的实例对象，对象节点的右指针指向下一个对象节点。每个对象节点的左指针指向嵌套层次的父亲节点。在类层次中，一个子类的实例对象只存储子类中定义的属性和元素，从超类继承的属性和元素存储在相应的超类实例对象中。

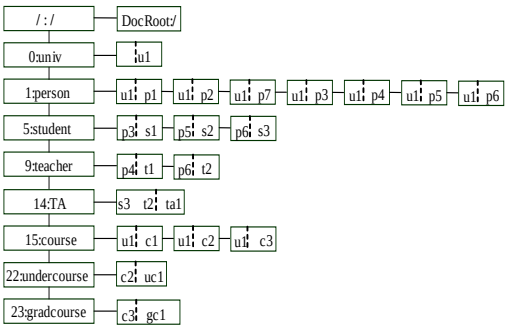

图 3　XML 数据树 T1 的分布式数据仓

分布式数据仓的类节点“ person ”的存储结构、对象节点的存储结构见表 1、表 2。

2.2　集中式存储

2.2.1　路径仓

在集中式存储中，类层次信息的定义位：在以索引类为根的类层次子树上存放该类全部的元素和属性的对象 id。超类中没有子类实例的数据，只有其直接实例的数据。图 5 是图 1 中 XML 数据树集中式存储的路径仓。

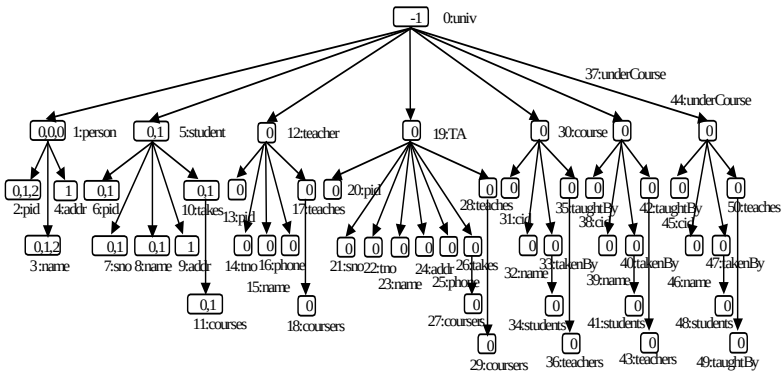

图 5　XML 数据树 T1 的集中式路径仓

2.2.2　数据仓

图 6 显示了图 1 XML 数据树的集中式存储数据仓的二叉树结构。数据仓中的节点存储类或实例对象，存储类的节点

被连接到根节点的左指针，类节点的右指针指向每个相应类的实例对象，对象节点的右指针指向下一个对象节点。每个对象节点的左指针指向嵌套层次的父亲节点。

和分布式存储数据仓的区别是：在类层次中，一个类的实例对象存储类中全部的属性和元素，超类中没有子类实例对象的数据，只有直接实例对象数据。

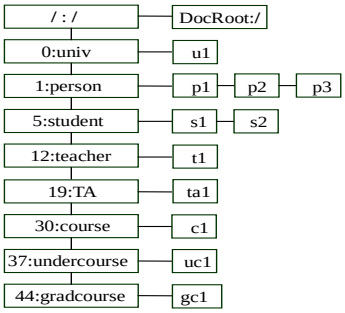

图 6　XML 数据树 T1 的集中式数据仓

3　性能测试

本系统的测试环境如下：(1)硬件环境：CPU P42.0GHz，1GB 内存，120GB 硬盘。(2)操作系统平台：Windows 2000 Professional 操作系统。(3)编程环境：JDK1.4.2_04，DOM API 采用的是 Apache XML4J-bin.4.3.0。

笔者设计了一个XML文档生成器，自动生成 3 个不同规模的、符合面向对象的DTD[2]的XML文档，文件名分别为oox1.xml、oox2.xml和oox3.xml。表 2 列出了 3 个XML文档的原文档大小、集中式存储模式和分布式存储模式大小。

从表 3 可以看出，分布式存储模式占用的存储空间比集中式略大。虽然分布式路径仓存储空间比集中式路径仓略小，但是分布式数据仓存储空间要大于集中式数据仓，主要是由于分布式数据仓的一个对象分 2 个部分进行存储；2 种模式的值索引占用空间大小相近。

表 3　XML 文档的存储空间

文档	文档大小/MB	集中式存储大小/MB	集中式装入时间	分布式存储大小/MB	分布式装入时间/s
oox1.xml	21.55	75.42	232s	78.56	315
oox2.xml	72.20	223.82	1 175s	231.53	1 536
oox3.xml	122.00	353.80	2 102s	365.65	2 923

为了对集中式存储模式和分布式存储模式进行“查询性能”的比较测试，针对前述的 3 个 XML 文档，对查询 1~查询 8 这 8 个语句的执行性能进行了测试，并对集中式和分布式存储模式进行性能比较分析。表 4 列出了 8 个测试查询的执行时间(取连续 5 次执行的平均值)，单位为 ms。

查询 1　查询 name=“John”并且 pid>1 000 的 person 的 addr。

查询 2　查询全部 person 的 pid 和 name。

查询 3　查询全部人的名字，包括 person、students、teachers 和 TAs。

查询 4　查询全部 addr 的信息。addr 不是某个类对应的实例对象。

查询 5　查询全部被 TA“Jack”教授的 underCourse 名称。

查询 6　查询教师“Helen”教授的全部 course，包括 underCourse 和 gradeCourse。

查询 7　查询全部 TA。

查询 8　查询全部 student 包括其所有子元素。

表 4　比较查询执行时间

查询编号	集中式存储模式			分布式存储模式		
	oox1.xml	oox2.xml	oox3.xml	oox1.xml	oox2.xml	oox3.xml
查询 1	29.5	32.3	38.2	35.6	42.9	56.3
查询 2	50.3	128.8	153.4	56.3	133.9	160.0
查询 3	230.3	438.1	653.2	123.6	395.2	576.6
查询 4	65.4	138.1	1 57.2	58.5	120.1	1 32.1
查询 5	175.4	823.4	1 235.2	200.3	987.0	1 675.3
查询 6	321.7	2 187.8	3 035.3	254.9	876.4	1 827.5
查询 7	42.1	89.5	1 09.7	123.2	258.3	306.5
查询 8	150.8	338.8	561.9	266.8	436.9	695.7

实验结果分析如下：

(1)查询执行时间与查询所涉及到的路径仓、数据仓的大小以及查询结果的大小有关。XML 文档越大，数据仓和路径仓越大，查询耗时越大，如查询 1。查询结果越多，查询耗时越大，如查询 2~查询 6。

(2)在分布式存储模式中，一个子类的对象分为 2 个部分(或多个部分)进行存储：超类继承的属性和元素，该子类定义的属性和元素。在存取一个对象时，要进行连接操作，因此，查询结果如果分布在 2 个部分(或多个部分)时，分布式存储模式效率较低；例如查询 7、查询 8。集中式存储模式在对类进行查询时要涉及整个类层次，在查询多态元素、多态引用，并且查询结果对应在一个类的对象实例时，集中式存储模式效率较低，例如查询 3、查询 6。对于分布式和集中式查询 1~查询 3，从数据仓取数据所耗时间相似，差异是路径仓的索引时间。

(3)使用通配符“//”的查询，结构索引法、节点索引法都进行了大量连接操作，效率较低；而路径仓方法使用组匹配，效率较高。例如查询 4。由于此查询查询的不是一个类对应的实例对象，而是所有子元素 addr，在集中式数据仓涉及到整个类层次，因此效率较低。

(4)由于引用索引相比结构索引和值索引耗时多，因此查询引用耗时较多，例如查询 5 和查询 6。

4　结束语

针对面向对象 XML 数据提出了分布式和集中式存储技术，路径仓提供了简明的组级路径概要和类层次信息，数据仓存储类和实例对象。分布式存储模式占用的存储空间比集中式略大。虽然分布式路径仓存储空间比集中式路径仓略小，但分布式数据仓存储空间要大于集中式数据仓，主要是由于分布式数据仓的一个对象分 2 个部分进行存储；2 种模式的值索引占用空间大小相近。分布式存储存取一个对象时需要频繁的连接操作，集中式存储不会因为属性和元素的继承而进行连接操作，这可以提高系统效率，但在对类进行查询时，就可能涉及整个类层次。

参考文献

1 Wang G. Extending XML Schema with Nonmonotonic Inheritance[C]//Proc. of the 1st International Workshop on XML Schema and Data Management. 2003: 402-407.

2 张晓琳, 王国仁. 用继承扩展 XML-RL[J]. 小型微型计算机系统, 2005, 26(2): 243-247.

3 Chaudhuri S, Shim K. Storage and Retrieval of XML Data Using Relational Databases[C]//Proc. of VLDB'01, San Francisco. 2001.

4 Goldman R, McHugh J, Widom J. Lore: A Database Management System for XML[J]. Dr. Dobb's Journal, 2000, 25(4): 76-80.

5 Meng X. OrientStore: A Schema Based Native XML Storage System[C]//Proc. of VLDB'03, San Francisco. 2003: 1057-1060.