

一种面向重构的 XML 混合编码方法

曹 睿^{1,2}, 吴玲达^{1,2}, 邓 维¹

(1 装备学院 复杂电子系统仿真重点实验室, 北京 101416; 2 国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

摘 要: 为了在关系数据库中高效地组织和管理 XML 数据, 提出了一种混合编码方法. 适当改进了向量编码方法, 通过向量梯度标识 XML 数据各节点的位置信息, 与各节点的路径信息相结合, 能够在数据重构中支持各节点映射与定位, 有效避免了由数据更新引发的重新编码问题, 实现了高效的路径查询; 由此建立了混合编码方案的数学模型、数据模型及关系模式. 通过性能分析和实验, 表明混合编码能够较好地支持 XML 数据的查询、更新与重构, 验证了该编码方法的有效性.

关键词: 向量编码; 路径编码; 混合编码; 数据模型; 数据重构

中图分类号: TP392

文献标识码: A

文章编号: 1000-7180(2014)04-0001-05

A Reconstruction-Oriented XML Hybrid Encoding Scheme

CAO Rui¹, WU Ling-da^{1,2}, DENG Wei¹

(1 Science and Technology on Complex Electronic System Simulation Laboratory, Academy of Equipment, Beijing 101416, China;

2 The Academy of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract: A hybrid encoding scheme is proposed to organize and manage XML efficiently with relational database. Vectors encoding scheme is improved, and vectors gradients are utilized to identify XML nodes position information, combining with nodes path information; which supports nodes mapping and location, avoids re-encoding caused by data update, and achieves quick path query. The hybrid encoding mathematic model, data model and relational schema are established. Performance analysis and experiment results show that, the hybrid encoding scheme can support XML query, update and reconstruction well and validly.

Key words: vectors encoding; path encoding; hybrid encoding; data model; data reconstruction

1 引言

随着计算机和网络技术的不断发展, XML (Extensible Markup Language, 可扩展标记语言) 作为 W3C 的一项国际标准, 已经成为网络上信息表示和交换所广泛采用的数据格式^[1]. XML 的组织与管理方法一般可以分为四类: 基于文件系统的组织、基于 Native 数据库的组织、基于面向对象数据库的组织、基于关系数据库的组织. 需求与技术两种驱动导致现阶段大多数的研究主要集中于在关系数据库基础上组织 XML 数据; 同时为了提高查询的

效率, 近年来人们进一步开展了关系数据库中 XML 索引和查询技术的研究^[2], 而这些技术大部分基于对 XML 数据的编码方法, 编码技术具备处理未知数据模式 XML 数据的特点, 在 XML 数据处理中变得越来越重要^[3]. 根据编码原理, 目前 XML 数据的编码机制可以分为节点序号类编码与节点路径类编码两大类^[4], 它们具有各自的特点. 在由关系数据库重构为 XML 数据时, 若 XML 数据经过删除、插入等更新行为之后, 数据更新要求编码进行相应的调整以保证基于该编码的各种查询和重构算法正确, 而这种重新编码的代价可能是很大的, 甚至需要更新

收稿日期: 2013-07-10; 修回日期: 2013-09-03

基金项目: 国家自然科学基金项目(61202129)

XML 数据的全部编码,然而现有这些编码缺乏对数据查询、更新与重构等行为支持的能力。

本文将节点序号类与节点路径类编码的思想相结合,提出了一种面向重构的混合编码方法。

2 基于向量的混合编码方案

将力学、解析几何中向量的概念引入到 XML 数据编码领域,本文的编码方案利用向量梯度标识节点的次序位置,同时记录节点的路径信息以支持字符串模式匹配。

2.1 数学模型

本文编码所采用的向量均位于二维笛卡尔坐标系第 IV 象限,令 $v = (x, y)$, 表示向量。

定义 1 梯度(Gradient), 向量 v 的梯度定义为 $G(v) = -y/x$ (1)

定理 1 假设 $a = (x_1, y_1)$, $b = (x_2, y_2)$ 表示两个向量, 它们的梯度分别为 $G(a)$ 和 $G(b)$, 若 $y_1/x_1 > y_2/x_2$, 则 $G(a) < G(b)$ 。

定理 2 假设向量 $c = (x_3, y_3)$, 对于定理 1 中的向量 a 与 b , 若 $c = a + b$, 如图 1 所示, $G(c) = -y_3/x_3 = -(y_1 + y_2)/(x_1 + x_2)$, 那么 $G(a) < G(c) < G(b)$ 。

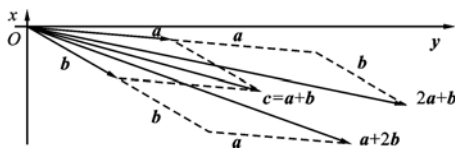


图 1 平面向量线性运算

证明: 由定理 1 知 $y_1/x_1 > y_2/x_2$, 则有

$$G(c) = -\frac{y_1 + y_2}{x_1 + x_2} = -\frac{(y_1 + y_2)/x_1}{(x_1 + x_2)/x_1} < -\frac{y_2/x_2 + y_2/x_1}{(x_1 + x_2)/x_1} = -\frac{y_2}{x_2} = G(b) \quad (2)$$

同理有 $G(c) < G(a)$, 所以 $G(a) < G(c) < G(b)$ 。

推论 1 由向量 a, b 线性运算能够得到向量 $2a + b, a + b, a + 2b$, 根据定理 2 可知, 它们的梯度都处于以 a, b 梯度 $G(a), G(b)$ 构成的梯度区间中。

推论 2 若向量 c 的梯度取值区间为 $(G(a), G(b))$, 根据推论 1 可知, 向量 $2a + b, a + b, a + 2b$ 的梯度也处于向量 c 的梯度取值区间中。

推论 3 对于推论 2 中的向量, 向量 $2a + b$ 与 $a + b$ 的梯度差为 $G(a)$, $a + b, a + 2b$ 的梯度差为

$G(b)$, 则由 $2a + b, a + b$ 构成的梯度区间较小, 且均处于向量 c 的梯度取值区间中。

2.2 数据模型

建模的基本思想是将 XML 数据转化成为一个树图, 以 XML 数据中的基本组成元素作为树图中的节点, 元素之间的关系用树图中的边表示, 建立节点数据模型。不失一般性, 考虑的节点类型有四种, 分别是元素、属性、文本、空元素, 对于其他类型的元素, 如注释、名字空间等, 本文建模原理同样适用。对于数据模型中的每个节点, 采用无模式的映射方法, 通过固定的关系模式, 一方面记录节点基本信息, 另一方面将节点间的结构信息反映在关系模式中。

应用上节建立的数学模型, 由坐标原点 O 出发, 利用一系列向量标识 XML 数据基本组成元素的次序信息, 如图 2 所示, 于是 XML 数据中的每一个基本元素都可以用数学模型中的向量唯一标识。注意到元素的属性与该元素在同一标记中, 而模型中视它们为两种类型的节点, 需要用两个向量进行标识, 如图 2 中次序为 3、10 的属性节点, 图中的虚线箭头即为该类节点的标识向量示意。

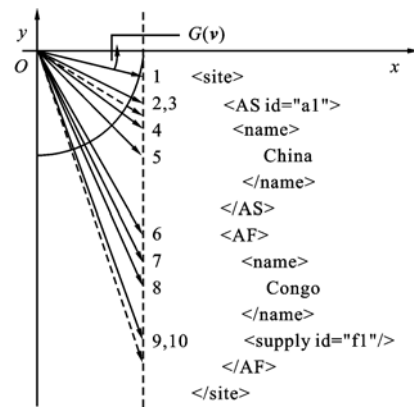
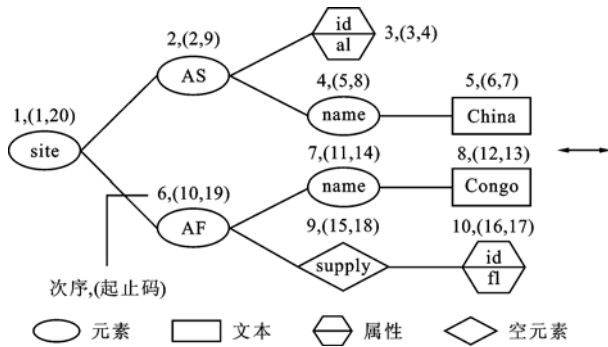


图 2 节点次序与标识向量的映射

通过前序遍历建立起 XML 数据的节点数据模型并记录每个节点的起止码^[5], 参考文献[6]的方法将向量编码应用到起止码中, 得到各节点的起止向量, 如图 3 所示。以起始向量为各节点的标识向量, 通过定义 1 计算出起止向量的梯度, 每个节点的混合编码即由其标识梯度(起始梯度)、终止梯度以及路径信息组成。对于各节点的路径信息, 如图 3 路径编码所示, 用“#”代替路径分隔符“/”作为该节点的路径编码, 以避免在直接利用 SQL 模式匹配功能进行查询时出现错误。



次序	起始向量	终止向量	标识梯度	终止梯度	路径编码
1	(1,0)	(0,1)	0	$+\infty$	# site
2	(5,1)	(3,2)	0.2	0.666 7	# /site/AS
3	(4,1)	(3,1)	0.25	0.333 3	# site#/AS
4	(5,2)	(5,3)	0.4	0.6	# site#/AS#/name
5	(2,1)	(7,4)	0.5	0.571 4	# site#/AS#/name
6	(4,3)	(1,4)	0.75	4	# site#/AF
7	(1,1)	(5,8)	1	1.6	# site#/AF#/name
8	(3,4)	(2,3)	1.333 3	1.5	# site#/AF#/name
9	(3,5)	(1,3)	1.666 7	3	# site#/AF#/supply
10	(1,2)	(2,5)	2	2.5	# site#/AF#/supply

图3 节点数据模型及其混合编码

例如,在查询拥有“AF”与“name”祖先—后裔关系节点时,正则表达式“#/AF#%/name”能够匹配正确结果;若以“/”为分隔符,正则表达式“/AF%/name”会将“/AFS/name”等结果错误匹配为查询结果。

2.3 关系模式

根据节点数据模型,设计存储节点与节点间结构信息的关系模式,映射并记录各节点的数据,关系模式构成如下:

标识梯度(Gradient)、终止梯度(EndGradient)、路径编码(Path)、节点类型(Type)、扩展名(Extend)、节点值(Value)。

其中,扩展名(Extend)主要用于记录属性、名字空间等相似类型的元素节点。如,当节点类型(Type)为属性节点时,扩展名记录该节点的属性名称,属性值则由节点值(Value)记录。

该关系模式保证了数据的完整性,在数据重构时不会丢失原始XML数据信息;支持基于内容和结构的查询,而且能够由一张基本表进行存储,在一定程度上避免了多表之间的连接操作。尤其是设计的关系模式不依赖于XML数据的DTD,具备普遍适用性。

3 编码性能分析

定义2 若 p, q 为数据模型中的两个节点,则 $\text{Grad}(p)$ 、 $\text{EGrad}(p)$ 、 $\text{Type}(p)$ 分别对应节点 p 的标识梯度、终止梯度、路径编码、节点类型、扩展名; $\text{Dpath}(p, q)$ 代表节点 p 与 q 的分隔符数量差。

3.1 结构关系判断

Xpath 作为 W3C 所推荐的查询语言,本质上要求所有的编码在结构关系判断方面都能够支持它,本文混合编码方案能够支持 Xpath 所定义的全部

13 个查询轴^[1]:

(1) child 轴、parent 轴。

$\text{child}(p): \text{Grad}(p) < \text{Grad}(q), \text{EGrad}(p) > \text{EGrad}(q)$ 且 $\text{Dpath}(p, q) = 1$, $\text{Type}(p)$ 为元素类; $\text{parent}(p)$ 类似。

(2) ancestor 轴、descendant 轴。

$\text{ancestor}(p): \text{Grad}(p) > \text{Grad}(q), \text{EGrad}(p) < \text{EGrad}(q)$, $\text{Type}(p)$ 为元素类; $\text{descendant}(p)$ 类似。

(3) following 轴、preceding 轴。

$\text{following}(p): \text{Grad}(p) < \text{Grad}(q), \text{Type}(p)$ 为元素类; $\text{preceding}(p)$ 类似。

(4) following-sibling 轴、preceding-sibling 轴。

$\text{following-sibling}(p): \text{EGrad}(p) < \text{Grad}(q)$, $\text{parent}(p) = \text{parent}(q)$; $\text{preceding-sibling}(p)$ 类似。

(5) self 轴、ancestor-or-self 轴、descendant-or-self 轴。

$\text{ancestor-or-self}(p): \text{Grad}(p) \geq \text{Grad}(q)$, $\text{EGrad}(p) \leq \text{EGrad}(q)$, $\text{Type}(p)$ 为元素类; $\text{descendant-or-self}(p)$ 类似。 $\text{self}(p): \text{Grad}(p) = \text{Grad}(q)$, $\text{EGrad}(p) = \text{EGrad}(q)$ 。

(6) attribute 轴、namespace 轴。

$\text{attribute}(p): \text{Grad}(p) < \text{Grad}(q), \text{EGrad}(p) > \text{EGrad}(q)$ 且 $\text{Dpath}(p, q) = 1$, $\text{Type}(p)$ 为属性; $\text{namespace}(p)$ 类似。

3.2 节点更新规则

不失一般性,假设需要在节点 p 与 q 之间插入节点 u ,如图4所示,通过节点 p 与 q 的起止梯度能够得到它们在单位圆中的起止向量,因此节点 u 的起止向量,能够根据推论3通过线性运算得到:即认为节点 p 的终止向量 $((1 + m^2)^{-1/2}, -m(1 + m^2)^{-1/2})$ 为推论3中的向量 $\mathbf{a}$,节点 q 的标识向量

$((1+n^2)^{-1/2}, -n(1+n^2)^{-1/2})$ 为推论 3 中的向量 $\mathbf{b}$, 因此插入节点的起止向量为 $2\mathbf{a}+\mathbf{b}, \mathbf{a}+\mathbf{b}$. 根据这种更新规则, 能够在两个节点之间插入无限多个节点, 而且既保证了插入节点的次序正确、占用梯度区间少, 又不会对其他节点编码造成影响, 为数据重构提供了较好的支持.

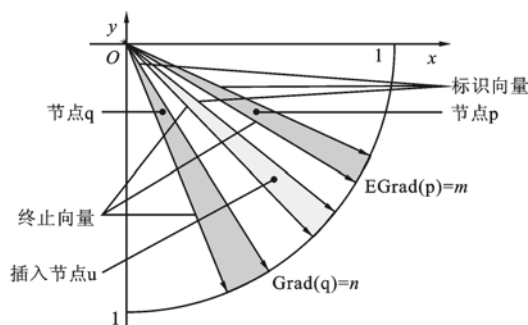


图4 节点插入示意

3.3 数据重构支持

编码方案与节点更新规则的设计, 使得关系数据能够方便的重构为 XML 数据.

编码方案是通过向量记录节点在 XML 数据的次序位置, 由定义 1 可以得到节点的梯度, 梯度与节点的次序存在一一映射关系, 如图 3 所示. 在关系数据库中, 若以关系模式中的 Gradient 为主码, 记录节点数据的元组将按照标识梯度升序排列, 更新的节点也会按照标识梯度自动排序(也可以通过 SQL 中的 ORDER BY 排序), 与该节点在 XML 数据中的次序位置正确对应. 这样, 数据库中的元组顺序即为重构 XML 数据节点时的次序, 实现重构节点的映射与定位.

采用起止向量的梯度作为起止码主要有以下好处: 第一, 面向于数据重构, 简单的通过升序排列标识梯度, 就能够按照 XML 数据重构的节点次序获得元组; 第二, 由于在向量间可以插入无限个向量, 这使本文编码具备很好的动态性能, 支持节点的各种更新操作; 第三, 从理论上能够更加合理地分配各节点编码: 一方面, 起止梯度实质上为不同深度的节点赋予了不同的梯度区间, 另一方面, 根据推论 3, 插入节点更新占用梯度区间较少, 可以有效的避免起止码区间增长过快的问题, 如图 3、图 4 所示, 次序为 6 的节点终止梯度仅为 4, 梯度区间利用率较高. 总之, 本文编码结合了标识向量与路径信息, 满足了保持节点结构关系与次序这两个 XML 编码的基本要求, 同时能够较好地支持数据查询、更新与

重构.

4 实验结果

混合编码原型系统以 C++ QT 4.0 实现, 实验环境为: CPU Intel Pentium IV 3.0 GHz; 内存 1 GB; 操作系统 Windows XP; 关系数据库 Microsoft SQL Server 2005; 采用 QtXml 模块作为 XML 解析接口, 进行了两类实验, 所有的实验都重复 10 次, 结果为排除最大值和最小值之后的平均值.

4.1 数据重构

实验数据采用两种标准测试集 Shakespeare's play^[7] 和 Xmark^[8]. Shakespeare 测试集共有 37 个 XML 数据, 最大深度为 6, 总大小 7.29 MB, 取该测试集中 1 个、7 个、13 个、19 个、25 个、31 个、37 个 XML 进行数据重构实验; Xmark 的参数设置为 0.001、0.002、0.004、0.008、0.016、0.032、0.064, 分别生成 XML 并进行数据重构实验.

实验结果如图 5 所示, 两种测试集各自数据重构时间与节点数量的比值基本呈现两条斜率不同的直线; 与 Shakespeare 测试集相比, Xmark 测试集数据重构所需要的时间相对较长, 实验结果说明: 对于同一测试集, 本文编码方法具有很好的扩展性, 具有线性时间复杂度; 不同的数据重构速度, 能够反映出两种测试集节点的深度、类型、结构等方面异同.

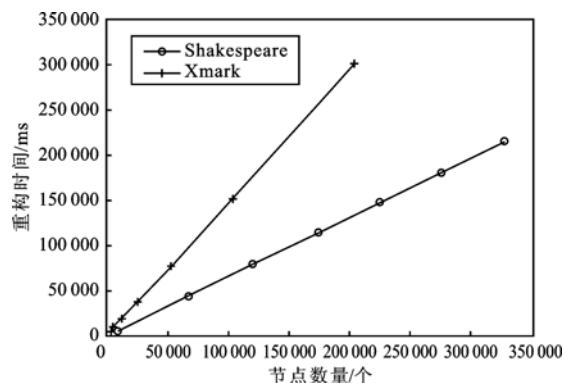


图5 数据重构

4.2 数据查询

实验数据采用 Xmark 测试集, 参数设置为 0.005 生成大小为 567 kB、最大深度为 12 的测试数据. 分别采用改进前的向量编码^[6]与本文混合编码进行 6 组路径查询实验, 结果如图 6 所示.

可见除了第二组实验查询速度相同外, 本文编码查询速度均要优于向量编码. 这是因为混合编码

记录了节点的路径信息,有效避免了连接查询、生成中间结果集等耗时操作的影响,查询路径的长度对混合编码查询速度的影响程度很小.第二组实验属于 ancestor 轴、descendant 轴的结构关系判断,向量编码利用起止码原理进行查询,耗时为 175 ms;混合编码若采用路径信息进行查询,耗时为 183 ms,比向量编码耗时长;由于起止码适合此类路径查询,所以混合编码不采用路径信息进行查询,而选用原理相同的起止梯度进行查询,耗时与向量编码相同.实验结果说明:混合编码吸收了向量编码与路径编码的优点,可以为同一种查询路径提供多种查询手段,根据查询路径选择耗时较少方式进行查询,查询效率较高.

查询 ID	路径查询
Q1	/site//mailbox//mail
Q2	//item//mail
Q3	//item//mailbox/mail
Q4	//item/mailbox/mail/text
Q5	//item[. /mailbox]//mail
Q6	//item[. /mailbox]//mail/text

(a)

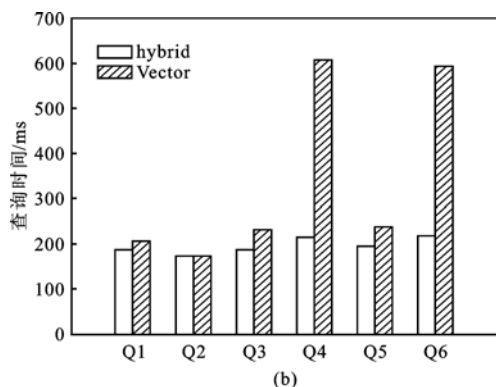


图6 查询时间

5 结束语

本文提出了一种面向于数据重构的混合编码方法,通过性能分析与实验验证,与传统方法相比,本文编码不依赖于 DTD,在支持数据重构的同时,有

效避免了数据更新造成的节点重新编码问题,兼具较高的查询速度.为高效组织与管理 XML 数据,提供了一种可行的方式,下一步将在混合编码基础上开展查询优化技术的研究.

参考文献:

- [1] W3C. XML Technology[EB/OL]. [2013-06-07]. <http://www.w3.org/standards/xml>.
- [2] 张晓琳, 吕庆, 刘立新, 等. 支持更新的连续不确定 XML 编码方案[J]. 微电子学与计算机, 2013, 30(2):157-160.
- [3] 孟小峰. XML 数据管理概念与技术[M]. 北京:清华大学出版社, 2009.
- [4] 孔令波, 唐世渭, 杨冬青, 等. XML 数据索引技术[J]. 软件学报, 2005, 6(12):2063-2079.
- [5] Zhang C, Naughton J F, DeWitt D, et al. On supporting containment queries in relational database management systems [C]//Proceedings of the 2001 ACM SIGMOD international Conference on Management of Data. Santa Barbara:ACM press, 2001:425-436.
- [6] Xu L, Bao Z F, Ling T W. A dynamic labeling scheme using vectors [C]//Proceedings of the international Conference on Database and Expert Systems Applications. Regensburg:IEEE press, 2007:130-140.
- [7] Niagara group. NIAGARA Experimental Data[EB/OL]. [2013-06-07]. <http://research.cs.wisc.edu/niagara/data/html>.
- [8] CWI. Xmark-An XML Benchmark Project [EB/OL]. [2013-06-07]. <http://www.xml-benchmark.org>.

作者简介:

曹 骥 男,(1985—),博士研究生.研究方向为空间信息获取与处理技术、多媒体信息系统与虚拟现实技术.

吴玲达 女,(1962—),博士,教授.研究方向为空间信息获取与处理技术、多媒体信息系统与虚拟现实技术.

邓 维 男,(1986—),博士研究生.研究方向为空间信息获取与处理技术、多媒体信息系统与虚拟现实技术.