

一种新的基于 Dewey 编码的 XML 路径索引

李玲娟,倪 铖,韩京宇

(南京邮电大学 计算机学院,江苏 南京 210003)

摘要:建立高效的索引来快速定位满足要求的节点是提高 XML 数据查询效率的一个必要手段。文中以降低复杂度和提高查询效率为目标,以基于路径的 XML 索引原理为基础,提出了一种新型的基于 Dewey 编码的索引结构 RTL-Index。RTL-Index 通过对文档节点编码来表示结构信息,利用前缀路径匹配操作完成结构查询,支持含通配符“*”和后代轴“//”的查询以及兄弟节点无序的模式树的查询。仿真实验结果表明 RTL-Index 索引具有较低的时间和空间复杂度,解决了 XML 文档分支路径查找问题,是一种较为有效的 XML 索引结构。

关键词:XML;查询处理;Dewey 编码;索引

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2010)10-0098-05

A Novel Dewey - Based XML Path Index for XML Data

LI Ling-juan, NI Cheng, HAN Jing-yu

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Establish an efficient index is a necessary means for meeting the requirements of positioning nodes and improving the efficiency of XML document query. In order to reduce the complexity and to improve the querying efficiency, according to the principle of XML path index, designs a novel Dewey - based XML path index for XML data, which is named RTL - Index. It expresses the structural information by encoding the document nodes, and implements the structural query process through matching prefix path. It supports wildcard “*”, descendant axis “//”, and queries of pattern trees with out-of-order brother nodes. The experimental results demonstrate that RTL - Index has low time complexity and space complexity, and can resolve the problem of branch path query for XML document. It's an effective XML index.

Key words: XML; query processing; Dewey; index

0 引言

XML 是一种文本标记语言,具有灵活、开放、跨平台的特点,近年来得到了快速的发展,已成为数据表示和数据交换的标准。随着 Internet 尤其是电子商务、Web 服务等应用的快速发展,产生了大量的 XML 数据,对 XML 数据的管理特别是查询技术的研究便成为了当前的热点。

由于传统的基于树的遍历方法不能够满足 XML 数据的复杂的处理要求,因此建立高效的索引来快速定位满足查找要求的节点是提高 XML 数据查询效率的一个必要手段^[1]。在过去近十年的时间里不少学者对之进行研究并提出了一些用于提升 XML 数据查询和管理效率的 XML 索引方案^[2~4]。这些索引可以分

成三类:基于节点记录的索引^[5]、基于路径结构的索引^[6,7]和基于序列的索引^[8,9]。

基于节点记录的索引技术对 XML 文档中的所有节点赋予一个区间编码,通过编码能快速地判断任意两节点间的祖先后裔关系,可以避免对树的大量重复遍历。但是,对于长路径表达式,这类索引技术在查询过程中会产生大量中间结果,需对这些中间结果进行连接操作才能得到最终的查询结果,时空复杂度偏高,查询效率低。

基于路径结构的索引技术以路径信息为基础,使用树结构维护不同节点的路径信息。这样可以避免对 XML 文档中标签相同的节点的重复访问,可以有效提高查询效率。但是对于含有通配符和分支路径的复杂路径查询,需要进行多次结果连接操作,查询效率低。

基于序列的索引技术将树结构作为最基本的查询单元,将 XML 文档树和查询树转换成相应序列,通过子序列匹配来应答 XML 查询。通过一种序列来表示任意树的结构,使得结构匹配和序列匹配之间等价,这

收稿日期:2010-02-11;修回日期:2010-05-18

基金项目:国家自然科学基金(60863001)

作者简介:李玲娟(1963-),女,辽宁辽阳人,教授,研究方向为数据挖掘、网络安全、数据库新技术等。

样可以整体地完成查询,不产生中间结果,避免了昂贵的中间结果连接操作。但是序列索引中的假报警和假不予考虑问题^[8]会影响其查询效率。

文中借鉴基于路径结构的索引的思想,针对其缺陷,提出一种新的基于 Dewey 编码的 XML 多文档索引 RTL-Index。该索引不仅可以执行简单的线性查询,而且可以支持复杂的分支路径查询,对于通配符“*”和后代轴“//”等特殊查询也完全支持。

1 XML 结构化查询类型及应用场景定义

文中将 XPath 作为用户查询的格式,所涉及的 XPath 仅包括 {/, //, *, []} 四种轴或谓词语法。另外,将不含谓词判断的查询称为简单路径查询或线性查询,将含有谓词等分支结构的查询称为分支查询。图 1 中的 Q1 为简单线性查询,而 Q2 和 Q3 为分支路径查询。

关于 XML 文档查询的应用场景,可以细分为两种情况:第一类是在一个较大的 XML 文档(如 dblp.xml)中进行查询,返回给用户的是所有符合查询条件的节点 ID;第二类是在大量的较小的 XML 文档(如许多应用中的 RSS 数据)中进行查询,返回给用户的是所有符合条件的文档 ID。在现实应用中,以上两种情况经常混合使用。同时,这两种查询模式很容易相互转化。比如处理大文档 dblp.xml(第一类应用)时,往往会去掉文档中 <dblp> 这个根节点,从而得到大量的子文档片断,从而转化为第二类应用场景;同理,对于第二类应用,可以为所有小文档添加一个共有的虚根节点,从而转化为第一类应用。所以,两种查询模式是相通的,适用的查询方法也都是一致的。为了叙述方便,文中采用第二种查询模式,即在大量较小的 XML 文档中进行查询处理,并返回文档 ID 集。

2 Dewey 编码及分支节点位置计算

2.1 Dewey 编码原理

Dewey 编码(Dewey Decimal Classification, 点分十进制编码)已有超过一百年的历史,其最初主要用在图书馆分类系统上。基本思想是:将一个节点的双亲节点的编码作为该节点编码的前缀。例如,树 T 的一个节点的前缀编码为 $c(u)$,则节点 u 的孩子节点 v 的前缀编码为 $c(v) = c(u).n$,这里 n 是节点 v 在节点 u 的所有孩子节点中的序号。有了前缀编码,要判断一个节点 v 是否是另一个节点 u 的后裔,只需要判断字符串 $c(u)$ 是否是字符串 $c(v)$ 的前缀。前缀编码的一个重要性质是它们是字典有序的。以节点 r 为根的子树中任意一个节点 u ,它的前缀编码 $c(u)$ 大于它左兄弟子树中所有节点的前缀编码,小于右兄弟子树中所有节点的前缀编码。因此,前缀编码既能有效支持包含关系的计算,也能有效支持文档位置的计算。

2.2 分支节点位置的计算方法

分支节点定义:分支节点为 XML 文档或分支查询结构中产生分支的节点。比如图 1 中 Q2 的 S 节点和 Q3 中的 I 节点。

分支节点位置的计算方法示例如下:

图 1 中分支结构 Q2 的两个叶节点 V5 和 V4 的 Dewey 编码分别是 DEWEY1(1.1.1.1.1)和 DEWEY2(1.1.2.1.1)。分支节点的位置为:

$$\delta(\text{DEWEY1}, \text{DEWEY2}) = (1 - 1.1 - 1.1 - 2.1 - 1.1 - 1) = (0.0 - 1.0.0) = 2$$

即:分支节点的位置是分支结构中两个叶子节点的 DEWEY 编码对应位置的数值相减取正后,自左向右第一个非零位置的前一个位置。如 Q2 中的第一个非零位置是 3,则分支节点的位置是 $3 - 1 = 2$ 。即 RTL 标签路径 PSINV5 中的 S 节点。

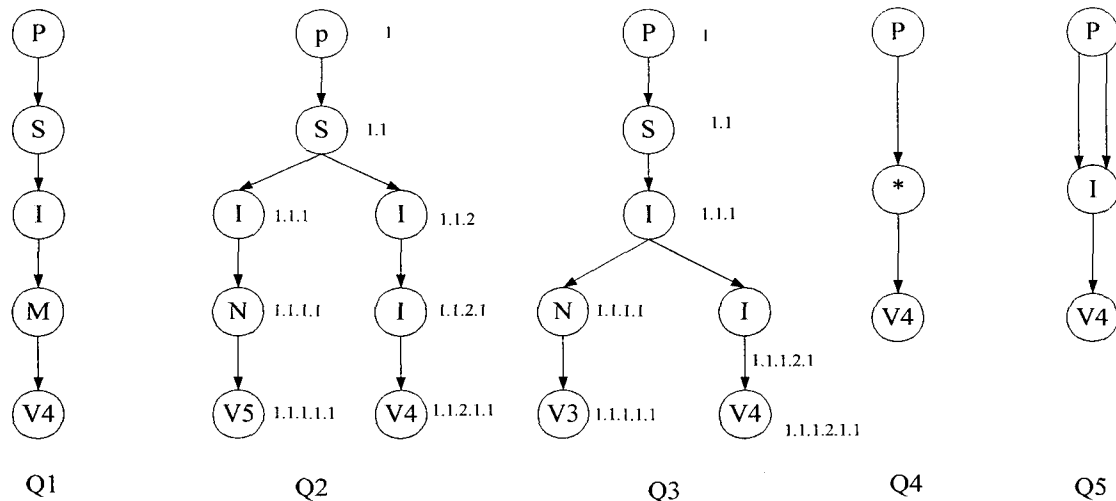


图 1 XML 文档中的查询结构

同理 Q3 的分支节点的位置为 3,即 S 节点。

若同一个文档中的两个分支结构 RTL 标签路径和分支节点相同则两个分支结构相同。

在文中设计的 RTL - Index 中,计算分支节点位置的目的是找出分支结构是在哪个节点产生分支的。比如 Q₁,Q₂ 虽然具有不同的结构,但是 RTL 标签路径完全一样,RTL 匹配查询的过程中会产生冗余结果。通过叶的 Dewey 编码计算出分支产生的节点的位置,可以消除冗余结果。

3 新的基于 Dewey 编码的 XML 路径索引 RTL - Index

3.1 RTL - Index 基本思想

文中设计的新的基于 Dewey 编码的 XML 路径索引是一种适用于多文档的结构化查询索引。其基本思想是:一个 XML 文档可以由有限条根到叶节点的 RTL(Root - To - Leaf)标签路径序列组成,如果能对各条 RTL 标签路径加载足够的结构信息就可以由 RTL 标签路径重构 XML 文档的树形模式。文中通过对 RTL 标签路径加载 Dewey 编码来保存结构信息,完成查询过程。

首先对每一个 XML 文档进行深度遍历并且对各节点进行 Dewey 编码,记录每一条 RTL 标签路径和叶节点的 Dewey 编码。

形成的 RTL 标签路径集为:

RTL₁: PrefixPath₁, (LeafNodeLable₁, DEWEY₁, DocId)

RTL₂: PrefixPath₂, (LeafNodeLable₂, DEWEY₂, DocId)

RTL_n: PrefixPath_n, (LeafNodeLable_n, DEWEY₁, DocId)

... ..

其中, n 为一个 XML 文档中叶子节点的数量。DocId 是文档的 ID,用于在多个 XML 文档中标识各个 XML 文档。

将查询结构同样处理成 RTL 标签路径集,如果查询结构是线性的,如 Q₁、Q₄、Q₅,则通过标签路径做为关键字匹配即可完成查询返回查询结果。如果查询结构为分支查询,首先通过上面的步骤形成 RTL 标签路径集,然后以 PrefixPath 为关键字在 XML 文档的 RTL 集合中进行匹配得到中间结果集,最后通过分支节点计算,完成结构匹配得到最终结果。

3.2 RTL - Index 的建立

为了更好地完成 RTL 标签路径匹配,需要对

XML 文档产生的大量的 RTL 标签路径建立有效的索引。文中通过 Tire 树管理 RTL 标签路径。具体步骤如下:

(1)深度遍历 XML 文档树,并对树中节点进行 Dewey 码标记。

(2)将 XML 文档树转换成 RTL 标签路径集。

以两文档查询为例,设文档 DocId1、DocId2 转换成的标签路径集如下:

序号	RTL
1	PSN(V1, 1.1.1.1, 1)
2	PSIM(V2, 1.1.2.1.1,1)
3	PSIIM(V3, 1.1.2.1.1.1,1)
4	PSIM(V4, 1.1.3.1.1, 1)
5	PBM(V1, 1.1.1.1, 2)
6	PBL(V2, 1.1.2.1.1, 2)

(3)建立一棵空的 Tire 树,然后依次将文档的 RTL 标签路径插入到 Tire 树中,并且对具有相同 PrefixPath 的叶节点建立 B + 树,以 LeafNodeLable 和 Dewey 为关键字。图 2 是由 Doc1 和 Doc2 建立起来的索引结构。

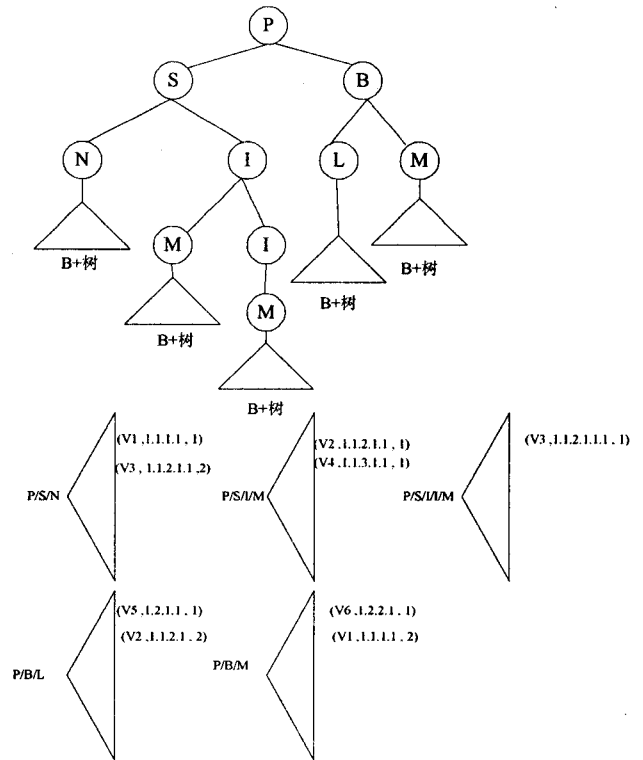


图 2 DOC1 和 DOC2 的索引结构

由于多个文档的根元素不一定相同,为每个文档都添加了一个虚根节点 R,以保证所有的 RTL 标签路径都有相同的起始元素标签。这样,所有的 RTL 标签

路径都可以顺利地依次插入 Tire 树,从而得到一棵根节点为 R 的 Tire 树。同样为查询结构也添加一个虚根节点 R,方便 RTL 标签路径的匹配。

3.3 RTL-Index 的查询

基于 RTL-Index 索引的 XML 文档查询算法包括三个步骤:一是将 XML 文档和查询请求转换成 RTL 标签路径,二是通过 Tire 树进行标签路径查询匹配,三是通过叶节点中的 DEWEY 编码进行结构验证得到最终的查询结果。

(1)RTL-Index 索引查询具体算法描述。

输入: 查询 Q 对应的 RTL 标签路径 $RTL_1, RTL_2 \cdots RTL_n$

输出: 给定文档 D 中满足查询 Q 的文档的 ID

过程: $RTL_Query(RTL_k, RTL_{k+1}, n, DocId)$

If $k \leq n$ then

/* 假设 RTL_k 对应的 B+ 树为 B_k , 通过 Tire 树的路径标签匹配可以得到对应的 $B_k + B_{k+1}$ 树根地址; 以 LeafNodeLable 为关键字分别在两棵 B+ 树中找到对应的 LeafNode_i 和 LeafNode_j */

StructureMatch(LeafNode_i, LeafNode_j)

For ($i = 0, i \leq N, i++$)

For ($j = 0, j \leq M, j++$)

/* N、M 为两棵 B+ 树中具有相同 LeafNodeLable 的节点数量 */

While(LeafNode_i.DocId = LeafNode_j.DocId)

If $\delta(\text{LeafNode}_i.\text{Dewey}, \text{LeafNode}_j.\text{Dewey}) = \delta(RTL_k.\text{Dewey}, RTL_{k+1}.\text{Dewey})$

/* 分支节点位置的判断 */

DocId \leftarrow DocId \cup LeafNode.DocId

$k++$;

End;

(2)算法性能分析。

以上匹配算法的最大时间复杂度为 $O(M * N * n)$, 由于文中的索引方法以 RTL 标签路径为最小的查询单位, 查询执行效率高。其中 Tire 树匹配操作在内存中进行, 所以查询过程中 I/O 次数少。与传统的标签分解、中间结果连接的算法相比, 时间和空间效率都得到了大的提升。

由于 RTL-Index 索引支持兄弟节点无序的查询, 添加、删除文档节点的操作不会对索引结构造成影响。

对于含有“*”、“/”的特殊查询请求同样可以通过 RTL 标签路径匹配和分支节点计算来完成查询过程。

4 实验及结果分析

用 C++ 实现了 RTL-Index 的算法, 并且使用

BerkeleyDB Library 提供的 B+ 树 API^[10]。为了比较性能, 也使用相同的技术实现了 ViST+^[8]和 XISS。试验环境为 1.83GHz Pentium IV 处理器、512MB 内存、80G 硬盘、装有 Windows XP 操作系统的微机。实验数据集是标准数据集 DBLP^[11]、XMARK^[12]。

DBLP 数据集是一个即时维护更新的用来记录科学文献的数据集, 结构非常“扁平”, 其对应的文档树的深度不超过 6。在处理 DBLP 数据集时, 去除其根节点, 把它的二层节点对应的子树片断作为独立的 XML 文档进行处理。

XMark 数据集是一个人工生成的拍卖数据集, 具有非常深的嵌套结构。取该数据集的部分数据作为实验数据, 包括所有以 item、person、close-auction 以及 operauction 为根节点的子树, 实验数据的基本情况见表 1。

表 1 实验数据基本统计信息

数据集	数据集大小(MB)	XML 文档数	不同内容节点数	文档最大深度
DBLP	297	700976	12787102	6
XMark	103	93730	1790718	11

针对数据集, 实验中选择了具有代表性的六个查询(见表 2)用于检验 RTL-Index 索引的性能。

表 2 查询集

序号	数据集	Xpath
Q1	DBLP	/R/inproceeding/title (R 为增加的虚根节点, 下同)
Q2	DBLP	/R/article/author[text()='Xiaoping Li']
Q3	DBLP	/R/article/[key='journals/pami/Lee98']/author
Q4	XMARK	/R/site/item[location='US']/mail/data[text()='12/15/1999']
Q5	XMARK	/R/site/person/* /city[text()='Pocatello']
Q6	XMARK	/R/closed-auction[person='person1']/data[text()='12/15/1999']

实验结果如图 3 所示。

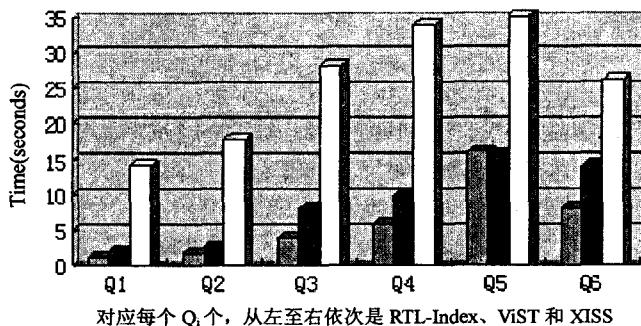


图 3 索引性能比较

Q1 是简单路径查询,查询路径中不含属性值,对于这种查询,RTL-Index 直接从相应的 B+ 树中检索结果,而序列索引 ViST 则需要不同的 B+ 树种逐一进行匹配。Q2 是含有属性值的简单线性查询请求,RTL-Index 在相应的 B+ 树中根据属性值进行检索,这个过程一般只需一次磁盘交换。Q3、Q6 是两个简单分支查询请求,RTL-Index 索引结构具有很好的查询效率。Q5 是带有通配符的查询请求,多次的结构匹配操作影响了 RTL-Index 的效率,但是仍然可以接受。

RTL-Index 索引在响应线性查询和简单分支查询请求时具有非常好的效果,在处理复杂分支查询请求(带“*”、“\”)时由于多次的结构匹配和磁盘页面交换操作影响了执行效率,但仍与序列索引结构 ViST 相当。

5 结束语

RTL-Index 索引实际上是直接对 XML 文档树中的叶子节点建立索引,因为叶子节点具有最大的选择度。文中通过 Trie 树存储叶子节点的前缀路径,实现前缀路径共享,有效地提高了索引的空间效率。文中提出的结构匹配操作解决了 XML 文档分支路径查找的问题,比起传统索引结构中的连接操作有了很大改进。但是在最坏情况下多次的结构匹配操作会增加 I/O 操作,在一定程度上影响查询效率,同时在处理通配符和祖先轴的查询路径时,效率提高不够明显,这都是将要进一步研究的内容。

参考文献:

- [1] 孟小峰,王宇,王小峰. XML 查询优化研究[J]. 软件学

(上接第 97 页)

参考文献:

- [1] 姜太平. 立体显示技术成熟,短期有望进入家庭市场[J]. 实用影音技术,2009(5):40-42.
- [2] Liang Zhang, Tam W J. Stereoscopic Image Generation Based on Depth Images for 3D TV[J]. IEEE Transactions on Broadcasting, 2005, 51(2):191-199.
- [3] 王惠明,董文辉. “二维+深度信息”的立体电视应用[J]. 广播与电视技术,2009(10):62-65.
- [4] 戴琼海,李涛. 基于人机交互的平面视频转立体视频的方法[P]. 中国专利: ZL200810102033.1, 2008-09-03.
- [5] 戴琼海,尤志翔,刘继明. 基于实时人机对话的平面视频转立体视频方法[P]. 中国专利: ZL200810111774.6, 2008-10-15.

报,2006,17(10):2069-2086.

- [2] 王国仁,于戈,杨晓春,等. XML 数据管理技术[M]. 北京:电子工业出版社,2007.
- [3] 王晓峰,于江. XML 文档索引研究[J]. 科技信息(学术研究),2008(28):351-375.
- [4] 白治国,徐慧,张霞萍,等. 基于素数编码 Schema 的 XML 索引结构的研究[J]. 计算机工程与设计,2009,30(8):1043-1046.
- [5] Li Q Z, Moon B. Indexing and querying XML data for regular path expressions[C]//In: Apers P M G, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases(VLDB). [s. l.]:Morgan Kaufmann,2001:361-370.
- [6] Goldman R, Widom J. DataGuides: Enable query formulation and optimization in semistructured databases[C]//In VLDB. Massachusetts, USA: Morgan Kaufmann, 1997:436-445.
- [7] Min Jun-Ki, Chung Chin-Wan, Shim K. An adaptive path index for XML data using the query workload[M]//Information Systems. Oxford, UK: Elsevier Science Ltd, 2005:467-487.
- [8] Wang Haixun, Meng Xiaofeng. On the Sequencing of Tree Structures for XML Indexing[C]//ICDE. Washington, DC, USA: IEEE Computer Society, 2005:372-383.
- [9] Rao P, Moon B. PRIX: Indexing and Querying XML Using Pruffer Sequence[C]//ICDE. Washington, DC, USA: [s. n.] 2004:288-300.
- [10] Sleepcat Software. The Berkeley database(berkeleydb)[EB/OL]. 2006-02. <http://www.sleepyeat.com>.
- [11] DBLP xmlrecords[EB/OL]. 2008-09. <http://dblp.unitrier.de/xml/>.
- [12] XMARK: The XML-benchmark project[EB/OL]. 2002-10. <http://monetdb.cwi.nl/xml/2002>.

- [6] 孙阳. 二维视频转换为三维视频的关键技术研究[D]. 上海:上海交通大学,2008.
- [7] 禹晶,苏开娜. 块运动估计的研究进展[J]. 中国图象图形学报,2007,12(12):31-40.
- [8] ISO/IEC 13818-2, MPEG-2. “Generic Coding of Moving Pictures and Associated Audio”, Video[S]. 1994.
- [9] JVT-Q042, Revised H. 264 /MPEG-4 AVC Reference Software Manual[S]. 1995.
- [10] 李淳,苏开娜. 基于菱形搜索的改进的运动估计算法研究[J]. 计算机技术与发展,2008,18(11):117-122.
- [11] 马力妮,郑志辉,潘峰. H. 264/AV 视频编码技术研究[J]. 计算机技术与发展,2008,18(7):163-166.
- [12] 邢恺. 运动估计算法研究[J]. 哈尔滨师范大学自然科学学报,2008,24(4):55-57.