基于相容关系的 XML 索引机制

徐海渊, 吴泉源, 王怀民, 贾 焰
(国防科学技术大学计算机学院, 湖南长沙 410073)

摘 要: 随着 XML 逐渐成为 Internet 数据表示与数据交换的标准, 存储与查询 XML 数据变得日益重要. 由于传统方法无法适应 XML 数据新的需求, 使得 XML 索引成为一个挑战性的课题. 到目前为止, 已经出现了不少针对 XML 数据的索引方法. 然而, 在处理基于相对路径的查询上, 缺少有效的解决办法. 本文提出了一种基于相容关系的索引模式, 结合 XML 文档拓扑结构的自身特点, 能够有效地处理基于相对路径的查询. 这种模式的主要特点包括: (1) 利用 XML 数据的拓扑结构而不是 XML 文档的模式(DTD 或 XML Schema) 来进行相对路径到绝对路径的转换; (2) 对拓扑结构进行基于相容关系的数字方式编码, 能够快速确定对应结点的依赖关系. 实验证明这种方法在处理基于规则路径表示(尤其是相对路径) 的 XML 查询时具有更高的效率.

关键词: XML; 查询; 索引
中图分类号: TP04 文献标识码: A 文章编号: 0372-2112 (2003) 08-1155-05

Containment Based XML Indexing

XU Hai-yuan, WU Quan-yuan, WANG Huai-min, JIA Yan
(School of Computer Science, National University of Defence Technology, Changsha, Hunan 410073, China)

Abstract: With the advent of XML as a standard for data representation and exchange on the Internet, storing and querying XML data becomes more and more important. This poses a new challenge concerning indexing and searching XML data, because conventional approaches no longer apply to XML data. A variety of novel indexing techniques for efficiently retrieving the results have been proposed in the recent literature. However, relative path based query still needs to be more efficient. A containment based indexing scheme for XML data is proposed, which makes full use of the topological structure of the XML documents and can be more efficient to deal with relative paths than previous methods. The key properties of the scheme are as follows: (1) Transform relative paths to absolute paths by using the topological structure of the XML documents instead of DTD or XML Schema; (2) Apply a numbering scheme to the topological structure, which can quickly determine the ancestor-descendant relationship between elements in the hierarchy. Experimental results from the prototype system implementation shows that the scheme can be used to process XML queries with regular path expressions (especially relative paths) faster than previous approaches.

Key words: XML; query; index

1 引言

XML[1] 作为一种自定义的数据格式, 具有许多传统数据形式所不具备的特点. 尤其在 Internet 领域, 普遍认为 HTML 将逐步为 XML 所取代. 正因为如此, 近年来无论是学术界还是在商业领域, 对 XML 数据查询的研究异常活跃. 很多查询语言被相继提出, 例如 Lorel[2-4], XML-QL[5], XPath[6], XQuery[7] 等等. 这些语言规范的一个共同特征是都支持规则路径表示(regular path expressions, 简称 RPE). RPE 是一种适合 XML 数据表示的基于图模式的路径表示机制. 为了有效地处理基于 RPE 的查询, 先前的研究提出了很多新的索引技术, 例如 Lorel, XSet[8], T-2 index[9], ToXin[10], XISS[11], SphinX[12] 以及 Index Fabric[13] 等等.

其中, 斯坦福大学的 Lorel 是较早的面向半结构化数据的工程. Lorel 提出了包含多种索引方法的杂合索引体系, 并且提出了一种典型的基于路径的索引结构 DataGuides[14]. DataGuides 适于处理基于绝对路径(从根出发的简单路径) 的查询, 对于相对路径的情况则效率不高, 并且当数据库为图结构而不是简单的树结构时, DataGuides 可能出现时间和空间的消耗呈指数膨胀的情况. XSet 是伯克利大学开发的基于 XML 的搜索引擎. XSet 提出了一种基于主存的索引模式, 对于针对小规模应用的简单查询具有很高的效率. 其不足在于扩展性受到主存空间的限制, 并且索引机制相对单一, 无法适应复杂的情况. T-2 index 是一系列针对特定路径表的索引模式, 适于

收稿日期: 2002-07-08; 修回日期: 2003-03-26
基金项目: 国家 863 课题(No. 2002AA116040)

基于特定模板的查询,其弱点在于通用性不强. Index Fabric 提出了一种基于两种路径(粗糙路径和精炼路径)类型的索引模式.其中,针对粗糙路径的处理方式类似 DataGuides,而精炼路径实质上是对特定查询模式的优化. Index Fabric 的问题在于处理相对路径查询的效率不高,并且由于使用层次化的 Patricia 树,因而不适合基于范围的查询. XISS 索引模式包含元素索引、属性索引、名字索引、结构索引以及值表. XISS 的主要特点在于将一种类似文[15]的数字模式作用于数据文档,从而有助于高效地获取结点间的依赖关系. ToXin 索引模式实质上是将 XML 文档变成类似 DOM 树的结构.在处理基于 RPE 的查询时,该索引结构包含双向的导航能力,使得处理焦点可以沿着路径正向或者逆向移动.这种能力可以使得结果集的推演更为高效.然而,同 XSet 一样,由于 ToXin 的实现是基于主存的,因而其扩展性同样受到限制. SphinX 提出了一种基于 XML 模式的索引模式.该方法对 XML 数据与 XML 模式分别进行处理,形成各自的关联模型,在此基础上进行查询.

上述索引模式的共同问题在于对相对路径查询的处理没有给出有效的办法.尽管 SphinX 为高效处理相对路径查询提供了有益的思路:首先利用模式信息进行相对路径到绝对路径的转化.由于大多数索引技术都能相对高效地处理绝对路径查询,因而很明显这种转化可以提高查询处理的效率.然而 SphinX 的问题在于转化本身的效率不高.因为 SphinX 在问题背景中假设 XML 模式结构非常的小,在这个前提下使用非确定性有限自动机的方法来完成相对路径到绝对路径的转化,这实质上是基于广度优先的遍历过程. SphingX 认为由于模式结构非常小,因而即使用最简单的遍历也不会严重影响系统的性能,然而在实际应用中路径转化的效率问题是不容忽略的.

基于上述考虑,本文在先前研究的基础上,提出了一种基于相容关系的索引模式.该模式有两个主要特点.一是利用 XML 数据的拓扑结构而不是 XML 文档的模式来进行相对路径到绝对路径的转换.之所以不象 SphinX 那样利用 XML 模式来进行路径转化,主要因为模式定义为实际数据所包含的模式信息的超集.也就是说,模式定义通常包含许多实际数据中不存在的可能性.例如图 1 就给出了一个递归定义的例子.对于这样的定义,在相应的文档中递归项展开的层数是不确定的,这样在路径转化时就会出现两难的境地.如果不完全展开,则可能出现路径遗漏的情况;反之,则会陷入死循环. SphinX 对此没有给出明确的回答.而在我们的模式中,由于数据的拓扑结构是有限数据集合模式信息的真实反映,因此拓扑结构显然也是有限和封闭的.并且由于该结构实质上是相应模式定义的子集,因此信息复杂度至少不高于模式定义.

```
3! ELEMENT monograph(title, author, editor)4
3! ELEMENT editor monograph4
```

图 1 DTD 中的递归定义

另一特点是对拓扑结构进行基于相容关系的数字方式编码,从而能够快速确定对应结点的依赖关系.这种编码模式主要针对形如/ / A/ / B0的相对路径的查询处理.该路径泛指所有经过 A 而后到达 B 的路径,而不管中间经过多少结点.

2 系统模型

2.1 数据模型

定义 1 假定 E 为有限标号集合,A 为有限属性名字集合,S 指代文本信息.定义树模型 T= (V, lab, Ele, att, val, r),其中,V 是节点集合. lab 是 V 到 E∪A∪{S}的函数. Ele 为 V 到 V 中节点序列的部分函数,并且对于任意 v∈V,如果 Ele(v)有定义,则 lab(v)∈E. att 为 V×A 到 V 的部分函数,并且对于任意 v∈V,l∈A,如果 att(v, l)= vc 则 lab(v)∈E,且 lab(vc)= l. val 为 V 到字符串的部分函数,并且对于任意 v∈V, val(v)是字符串当且仅当 lab(v)= S 或 lab(v)∈A. r 为 T 的根,不失一般性,假定 lab(r)= r.

2.2 路径表示

定义 2 定义路径 Q::= ε| l. Q,其中 ε为空路径,节点标号 l∈E∪A∪{S},/ .0为连结操作符.定义 T|= Q(v1, v2)表示存在一条从 v1 到 v2 的路径.本文中连结操作符采用 XPath 的语法表述,即用/ /0表示绝对连接,用/ / /0表示相对连接.

3 拓扑结构索引

3.1 XML 数据拓扑结构

如定义 1、定义 2 所示,XML 数据可以表示为结点与边组成的树形结构(如果考虑 ID 属性的引用关系,则为图模式).这样,每一个结点都对应一个路径(这里指绝对路径),而路径相同的结点的集合成为路径等价类.显然,不同的等价类对应的路径是不同的,而所有不同的路径重叠的结果就形成数据的拓扑结构.

3.2 基于相容关系的数字模式

我们的数字模式采用深度优先算法进行编码.每一个结点的编码包括序偶3 start, end4,编码方式遵循以下规则:(1)对任意结点 x,有 start(x)< end(x),我们称[start(x), end(x)]为结点 x 的编码域;(2)如果结点 y 是结点 x 的祖先结点,则 start(y)< start(x),并且 end(y)> end(x);(3)如果 x、y 是兄弟结点,且 x 先于 y,则 end(x)< start(y);(4)假设树模型最大深度为 MAXDEPTH, depth( )为结点当前深度函数,x、y 为先序遍历中相邻的两个点(x 先于 y),则 start(y)- start(x)= f(MAXDEPTH2depth(y)),f 为空间预留函数.空间预留函数的作用在于预留空间以便当有新结点插入时,可以减少编码系统更新的开销.为了说明该数字模式能够用于确定结点间的相容关系,我们给出下面的定理.

定理 1 给定树模型 T 中的任意两个结点x、y,不妨假设 start(x)< start(y),如果 start(y)< end(x),则必有 end(y)< end(x).证明从略.定理 1 说明任意两个结点的编码域不存在交叉的情况.

定理 2 给定树模型 T 中的任意两个结点x、y,x 是 y 的祖先结点(记为 x = y)当且仅当 start(x)< start(y)< end(x).证明从略.定理 2 说明给定任意两个结点,可以在常量时间内确定其相容关系.

定理 3 给定树模型 T 中的任意不同结点x、y、z,如果 x

= y, 且 y = z, 则 x = z. 证明从略. 定理 3 说明了相容关系的传递性质.

3.1.3　拓扑结构索引

拓扑结构索引实质上是用数字模式对数据拓扑结构进行编码. 同时, 为了充分利用数字模式的优势, 我们为拓扑结构建立了基于标号的索引. 这样, 通过标号就可以容易地得到拓扑结构中该标号对应的所有结点. 需要说明的是, 拓扑结构中的结点是虚拟的结点, 不同于真实的数据结点.

4　路径转化算法

```
输　入: 标号 A, 标号 B　　/* 处理形如// A//B0的路径转换* /
输　出: 绝对路径集合 P, 匹配结点对的集合 M
初始化: TempSet1、TempSet2、P、M 初始化为空
第一步　利用针对拓扑结构的标号索引获取所有标号为 A 的结点, 将该集合赋给 TempSet1
    /* 标号索引中每个标号对应的所有结点预先按照 start( ) 的顺序排列* /
  利用针对拓扑结构的标号索引获取所有标号为 B 的结点, 将该集合赋给 TempSet2
第二步　取 TempSet1、TempSet2 的第一个元素分别赋给 K1、K2,
  if　TempSet1 中任意两个结点之间不存在相容关系
    loop
      if　　K1 = K2
            将 K2 对应的绝对路径加入 P, 将3 K1, K24加入 M
            取 TempSet2 的下一个元素赋给 K2
                while(K1 = K2)
                    将 K2 对应的绝对路径加入 P, 将3 K1, K24加入 M
                    取 TempSet2 的下一个元素赋给 K2
                end while
            取 TempSet1 的下一个元素赋给 K1
      else if start(K2) < start(K1) 取 TempSet2 的下一个元素赋给 K2
            else 取 TempSet1 的下一个元素赋给 K1
            end if
      end if
    until TempSet1 或 TempSet2 中所有元素处理完毕
  else
    loop
      if K1 = K2
            将 K2 对应的绝对路径加入 P, 将3 K1, K24加入 M
            取 TempSet2 的下一个元素赋给 K2
                while(K1 = K2)
                    将 K2 对应的绝对路径加入 P, 将3 K1, K24加入 M
                    取 TempSet2 的下一个元素赋给 K2
                end while
            取 TempSet1 的下一个元素赋给 K1
      else
          if start(K2) < start(K1) 取 TempSet2 的下一个元素赋给 K2
          else 取 TempSet1 的下一个元素赋给 Kc
            while(end(Kc) < end(K1))
                取 TempSet1 的下一个元素赋给 Kc
            end while
            K1 = Kc
          end if
        end if
      until TempSet1 或 TempSet2 中所有元素处理完毕
    end if
  return P
```

图 2　基于相容关系的路径转换算法

相对路径到绝对路径的转换中要面对两种路径情形. 一种是形如// /A. . .0 的路径, 这种形式的问题可以通过标号索引来解决.

另一种是形如/ . . . A// B0 的情况, 也就是说,/ / /0 存在于两个具体的标号之间. 这种情况下, 对于复杂的拓扑结构而言, 特别是当拓扑结构层次很深时, 像传统方法那样进行逆向遍历, 则效率上明显存在问题. 为此对这类路径的转换采用依靠数字模式的办法来解决. 也就是说, 通过利用数字模式直接判定结点间相容关系来形成高效的匹配. 限于篇幅, 本节只讨论这类路径的转换算法.

如图 2 所示, 路径转换算法包含两个步骤. 第一步根据查询中涉及的标号利用标号索引获得相应的结点集合. 第二步在前面得到的两个集合之间找出所有基于相容关系的匹配. 我们看到, 算法根据 TempSet1 的自相容状态采取了不同的处理方式. 这里的自相容是指 TempSet1 中的结点之间是否存在相容关系. 正是由于这种自相容关系的存在, 使得先前的匹配算法具有平方时间的算法复杂度. 为了提高匹配的效率, 我们在定理 1、2 和 3 的基础上得出如下的规律性结论: (1) 子孙结点相邻原则. 给定先序模式下的结点序列中的任意结点 x, x 的所有子孙结点均位于 x 结点与下一个非 x 子孙结点之间; (2) 绝对路径唯一性原则. 根据拓扑结构定义, 对于 $TempSet_2$ 的任意结点 x, 由根到 x 的绝对路径只有一个. 由上述结论同时根据定理 3, 我们知道对于 $TempSet_1$、$TempSet_2$ 的任意结点 x、y, 如果 x = y, 则所有 x 的子孙结点与 y 的匹配不会增加新的绝对路径; 否则, 所有 x 的子孙结点必定都不包含 y. 由此, 可以越过自相容关系的障碍, 使得匹配过程在一次扫描中完成, 从而使算法具有线性时间复杂度.

5 实验结果

我们已经用 C+ + 实现了本文描述的索引模式, 同时用 C 实现了一个基于 SAX 接口的 XML 解析器. 解析器用于快速解析 XML 文档, 以便生成文档树. 测试环境包括: P Ó 850 CPU, 512 兆内存, Linux 平台. 为了测试复杂 XML 模式下的实验效果, 我们综合当前已有的模式定义, 生成了一个较为复杂的 DTD 模式, 该模式与先前研究中使用的模式相比, 规模扩大了近 10 倍, 我们用 IBM 的 XML 文档生成器[16] 产生约 271M 的数据文档, 具体参数见表 1.

表 1 实验数据

数据文档	DTD	拓扑结构结点	不同元素结点	不同属性结点	总结点	最大深度
271M	320K	2108	584	151	1509673	16

如图 3 所示, 方法 1 表示传统索引技术中自底向上的索引方法, 也即通过全局标号索引获得标号 B 对应得结点集合, 然后对每一个结点沿绝对路径逆向回朔, 如果路径中存在标号 A 则为所求得匹配. 方法 2 采用本文描述的数字模式对所有的数据文档分别进行编码, 查询时, 首先通过全局标号索引获得 A、B 对应的结点集合, 然后对这两个集合进行基于相容关系的结点匹配. 方法 3 采用本文描述的查询方法, 先通过拓扑结构将相对路径转化为绝对路径, 然后通过路径索引获得与 A、B 的绝对路径相对应的结点集合, 然后对这两个集合进行基于相容关系的匹配. 从结果可以看出, 方法 3 好于方法 2, 方法 2 好于方法 1, 并且对于形如/ A// B0 的查询, A、B 之间的平均路径深度距离越大, 方法 3 的优势越为明显.

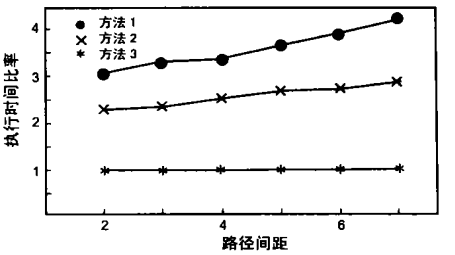

图 3 查询时间与参考点深度距离的关系

参考文献:

[1] T Bray, J Paoli, C Sperberg2McQueen. Extensible Markup Language (XML) 1. 0[S]. http: / / www. w3. org/TR/ REC- xml, 2000.

[2] J McHugh, S Abiteboul, R Goldman, D Quass, J Widom. Lore: A Database Management System for Semistructured Data[C]. SIGMOD Record, September 1997.

[3] J McHugh, J Widom, S Abiteboul, Q Luo, A Rajaraman. Indexing semistructured data[Z]. ftp: / / db. stanford. edu/ pub/ papers/ semiin2 dexing98. ps.

[4] S Abiteboul, D Quass, J McHugh, J Widom, J Wiener. The Lorel query language for semistructured data[J]. Intl. Journal on Digital Libraries, April 1997. 68- 88.

[5] A Deutsch, M Fernandez, D Florescu, A Levy, D Suciu. A query lan2 guage for XML[A]. Proc. of 8th Intl. World Wide Web Conf[C]. May 1999. 77- 91.

[6] J Clark, S DeRose. XML Path Language(XPath) Version 1. 0[S]. W3C Recommendation 16 November 1999. http: / / www. w3. org/ TR/ 1999/ REC2xpath219991116.

[7] D Chamberlin, D Florescu, J Robie, J Simon, M Stefanescu. XQuery: A Query Language for XML W3C[R]. Tech. Rep. WD2xquery220010215, WorldWide Web Consortium, February 2001.

[8] B Zhao, A Joseph. XSet: A lightweight XML search engine for internet applications[Z]. http: / / www. cs. berkeley. edu/ % 7Eravenben/ xset/ .

[9] T Milo, D Suciu. Index structures for path expressions[A]. Intl. Conf. on Database Theory[C]. 1997. 277- 295.

[10] F Rizzolo. ToXin: An indexing scheme for XML data[D]. M. Sc. The2 sis, Canada: Dept. of Computer Science, University of Toronto, January 2001.

[11] Q Li, B Moon. Indexing and querying XML data for regular path expres2 sions[A]. Proc of 27th Intl. Conf. on Very Large Data Bases[C]. 2001. 361- 370.

[12] L K Poola, J R Haritsa. SphinX: Schema2 conscious XML indexing[Z]. Database Systems Laboratory Dept. of Computer Science & Automation Indian Institute of Science.

[13] B Cooper, N Sample, M Franklin, G Hjaltason, M Shadmon. A Fast In2 dex for Semistructured Data[A]. Proc. of 27th Intl. Conf. on Very Large

Data Bases[C]. August 2001. 341- 350.

[14] R Goldman, J Widom. DataGuides: Enabling query formulation and op2 timization in semistructured databases[A]. Proc. of 23rd Intl. Conf. on Very Large Data Bases[C]. August 1997. 436- 445.

[15] Paul F Dietz. Maintaining order in a linked list[A]. In Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing[C]. San Francisco, California, May 1982. 122- 127.

[16] A Schmidt, F Waas, M Kersten, D Florescu, I Manolescu, M J Carey, R Busse. The XML Benchmark Project[R]. April 2001, http://monetdb.cwi.nl/xml/Benchmark/benchmark.html.

作者简介:

徐海渊　男, 1971 年 3 月生于太原, 国防科技大学计算机学院博士生, 主要从事分布计算、数据库等领域的研究. Email: hyxu@nudt.edu.cn.

吴泉源　男, 1940 年 2 月生于上海, 国防科技大学计算机学院教授, 博士生导师, 主要从事分布计算、人工智能等领域的研究.