XML 数据索引技术*

孔令波[1+], 唐世渭[1,2], 杨冬青[1], 王腾蛟[1], 高 军[1]

[1](北京大学 计算机科学技术系,北京 100871)

[2](北京大学 视觉与听觉信息处理国家重点实验室,北京 100871)

XML Indices

KONG Ling-Bo[1+], TANG Shi-Wei[1,2], YANG Dong-Qing[1], WANG Teng-Jiao[1], GAO Jun[1]

[1](Department of Computer Science and Technology, Peking University, Beijing 100871, China)

[2](National Laboratory on Machine Perception, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62755440, E-mail: lbkong@db.pku.edu.cn, http://www.pku.edu.cn

Received 2004-12-07; Accepted 2005-08-24

Kong LB, Tang SW, Yang DQ, Wang TJ, Gao J. XML indices. *Journal of Software*, 2005,16(12):2063–2079.
DOI: 10.1360/jos162063

Abstract: XML has become the *de facto* standards for data representation and exchange on Web applications, such as digital library, Web service, and electronic business, etc. Indexing technique is still significant for efficient XML data processing. This paper discusses the actualities of the recent researches on XML indexing. It classifies the techniques into two categories, node-record-style index with three subcategories, and structural-summary-style index. It analyzes the virtue and deficiency of the related schemes based on the considerations for query processing efficiency and data modification supporting. And hereby it proposes three issues for future XML indexing researches, including internal structure retrieval, multi-dimensional processing on node paths, efficient modification-validating support and the index amalgamation for satisfying both querying and IR on XML data.

Key words: XML index; interval encoding; B-E-L model; node numbering; bisimilarity; k-bisimilarity; structural summary; update XML; incremental validation; XML information retrieval

摘 要: 对 XML 数据建立有效的索引,是左右 XML 数据处理性能的重要因素.深入地讨论了目前 XML 索引技术的研究现状,将 XML 索引技术分为两大类:节点记录类索引(本身还可以分为 3 个小的类型)和结构摘要类索引.根据 XML 数据查询处理效率以及 XML 数据修改对 XML 索引的要求,讨论了相关 XML 索引方法的优点和不足,并归结出 XML 索引后续研究的 3 个方向:XML 结构信息的获取,路径信息的多维处理,数据修改合法性的有效支持,以及涉及能够同时有效满足 XML 查询和信息获取的索引.

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2002AA4Z3440, 2005AA4Z3070 (国家高技术研究发展计划(863))

作者简介: 孔令波(1974－),男,山东日照人,博士生,主要研究领域为关系数据库实现技术,XML 数据处理技术,数据挖掘;唐世渭(1939－),男,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,半结构化数据,Web 数据集成,数据挖掘;杨冬青(1945－),女,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,数据仓库,Web 数据集成,移动数据挖掘;王腾蛟(1973－),男,博士,副教授,CCF 高级会员,主要研究领域为数据库,数据仓库,Web 数据集成,数据挖掘;高军(1975－),男,博士,副教授,主要研究领域为数据库,数据仓库,半结构化数据,Web 数据集成,移动数据挖掘.

关键词: XML 索引;区间编码;B-E-L 模型;节点赋数;双似;k 阶双似;结构摘要;XML 数据修改;增量式验证; XML 信息获取

中图法分类号: TP311 文献标识码: A

XML(最新的规范为 2004 年的 XML1.1)(extensible markup language),即可扩展的标记语言,是一套定义语义标记的规范,其目标是能够定义计算机和人都能方便识别的数据类型.随着网络应用的快速发展,尤其是电子商务、Web 服务等应用理念的进一步发展,使得 XML 类型的数据成为当前主流的数据形式.对 XML 数据的管理也成为研究的热点[1].

XML 数据的基本形式是 XML 文档.对 XML 数据的处理分为两种不同的方式,一类是 XML 流处理,另一类为静态数据处理方式,即传统的数据管理形式.在后一种处理方式中,类似于索引在关系数据管理系统中的地位,XML 索引技术仍然是研究人员考虑的主要内容,也是本文关注的内容.

本文首先概述了基于静态 XML 文档数据之上的查询处理的情况,针对查询中的不足,将当前研究文献中出现的 XML 索引分为两大类别,并简要叙述了 XML 索引设计中应该考虑的主要因素;之后对当前 XML 索引的研究,分别从结构关系表示、结构关系的获取以及如何提供数据修改支持 3 个方面进行了阐述.最后是总结和研究展望.本文讨论的问题只是数据库研究领域中的一部分,观点也可能存在偏颇之处,但我们希望通过本文的工作,能给数据库研究者,尤其是正在进入相关研究领域的人员一些启发和帮助.

1 XML 索引及其分类

1.1 XML数据及XPath查询处理

```
〈?xml version="1.0" encoding="UTF-8"?〉
〈!DOCTYPE root SYSTEM "sample2.dtd"〉
〈root〉
  〈a〉a1
    〈b〉b1
      〈c〉c1〈/c〉
    〈/b〉
  〈/a〉
  〈a〉a2
    〈b〉b2
      〈c〉c2〈/c〉
    〈/b〉
  〈/a〉
  〈a〉a3
    〈b〉b3
      〈c〉c3 c4 c5 c6〈/c〉
    〈/b〉
  〈/a〉
〈/root〉
```

(a) Sample.xml

```
〈?xml version="1.0" encoding="UTF-8"?〉
〈!ELEMENT root (a+)〉
〈!ELEMENT a (#PCDATA | b)*〉
〈!ELEMENT b (#PCDATA | c)*〉
〈!ELEMENT c (#PCDATA)〉
```

(b) Sample.dtd

Fig.1 sample.xml and its DTD file

图 1 sample.xml 和 sample.dtd 示例

XML 规范规定了 XML 数据必须满足的条件,其基本的形式是 XML 文档.从逻辑上讲,一个 XML 文档通常由 5 部分组成:声明、元素、注释、字符引用和处理指令,为叙述方便,统称为 XML 数据单元.XML 数据内容的限制通常由 XML 模式(包括 DTD 和 XML Schema)来描述.图 1 为 sample.xml 文档及其 DTD 的示意.图中 root 为 sample.xml 的根,它具有 3 个标签为 a 的区段,而每一个 a 区段都嵌套一个以 b 标签为标志的区段.当给定 XML 文档时,该文档也就隐含地确定了其中所含的标记单元的顺序,称为 XML 顺序(XML ordering),这一概念在 XML 查询处理时是需要考虑的,即查询结果中的标记单元应当具有类似的顺序特征.

为了从 XML 以及半结构化数据中获取所需要的信息,研究人员开发了许多查询语言,包括 Lorel,XML-QL,XML-GL,Quilt,XPath,XQuery.它们共同的特征为:采用了正则路径表达式[2−4]的形式,其本质是捕捉 XML 数据单元间的结构关系和内容.XPath 是实现 XML 数据周游的基本语言,是 XSLT,XQuery 的基础.它首先定义了 XML 数据的树形模型.图 2 即为图 1 的满足 XPath 数模型定义的 XML 树.图 2 中每个元素节点左侧的数字表示该节点在前序遍历 XML 树形成的节点标签序列中的序号;右侧的数字表示该节点在后序遍历 XML 树形成的节点标签序列中的序号.由 XPath 的定义可以构建很复杂的查询语句,主要分为 3 个层次[5]:线性路径表达式、分支路径表达式和路径树.其中,路径树的定义包含了研究文章中常见的小枝概念:Twig[6−10].

当在 XML 文档类数据之上处理 XPath 查询时,借助于 W3C 定义的 XML 数据处理的两种接口规

范,DOM(document object model)和 SAX(simple API for XML),处理方式可概括为:顺序读取 XML 文档中的节点;如果该节点的路径满足 XPath 中定义的条件(包括结构关系、测试条件和谓词关系),那么该节点即为 XPath 查询语句的一个输出.这种基本的线性表达式查询语句的处理方式存在如下两个缺陷:

(1) 只能采取周游的方式在 XML 文档中寻找满足查询语句结构关系的数据单元,即为了获取满足查询路径的结果,必须周游所有可能的数据单元的标签路径,效率不高.

以图 2 为例,要想得到文本包含“c3 c4 c5 c6”的“c”节点,那么必须从根节点开始沿着 root→a→b→c 的路径到达.如果采取某种索引结构,可以直接定位到目标节点,将会大大提高查询的效率.

(2) 对 XML 中标签路径相同的节点,仍然需要重新遍历它们的路径.这个问题直接来自于(1).

同样以图 2 为例,如果目标节点是 c,文档中存在 3 条相同标签路径的节点路径,而根据上述基本处理方式,为了搜索所有的 c 节点,必须对同样的路径都要遍历.那么,将同样路径的节点汇集到同一节点中,就可以提高搜索的效率.

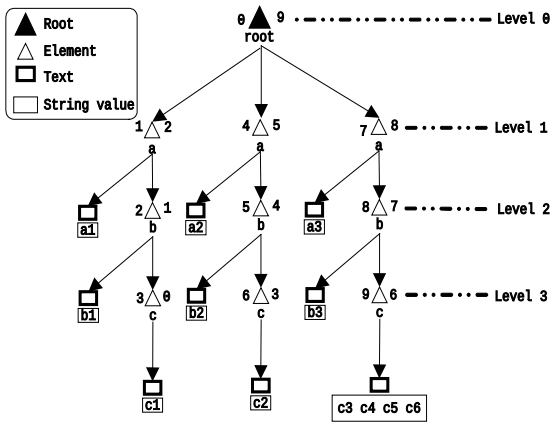

Fig.2 XML tree of sample.xml

图 2 sample.xml 对应的 XML 树

由此在实际研究中引申出两类 XML 索引形式:节点记录类索引和结构摘要类索引.

1.2 XML索引分类

1.2.1 节点记录类索引

本质上讲,第 1 类缺陷是由于查询处理必须通过标签路径查找节点这一限制造成的,即要想最终得到满足查询路径的节点,必须顺序地依次访问标签路径对应的节点路径才行.那么,如果能够改变 XML 数据管理的方式,使得可以避免必须通过路径找节点的限制,进而变为在某种辅助信息的帮助下,直接针对节点集合的划分得到最后结果的方式,这就形成了节点记录类 XML 索引的基本思想.

类似于树结构的保存,基于实际的处理要求,将 XML 数据单元(标签名称、属性名称、属性的值、文本,甚至是文本中的字符串等)作为记录保存;同时在数据单元的记录中保存有助于确定节点间结构关系的位置信息,模式为(数据单元标识,数据单元在 XML 中的位置信息).位置信息通常分为两类,一类是节点在某种遍历方法所得字符序列中的序号,另一类是节点在 XML 文档中的路径信息.

实际索引数据的保存可以采用不同的形式[11,12],如保存为文本(native storage)、关系数据形式和面向对象数据形式.其中最常见的就是将索引记录保存为关系的方式.本文主要针对这种方式展开叙述,其中涉及到的相关概念对其他方式的理解也是具有借鉴意义的.

1.2.2 结构摘要类索引

结构摘要是针对 XML 查询处理基本方式中的第 2 个缺陷而来的:既然是相同路径重复遍历的问题,那么,将 XML 数据按照路径进行约简,要求这种约简中只保存 XML 数据中不同的路径,将具有相同路径的节点集合作为约简中该路径的末端节点的内容,那么,在 XML 数据上的路径查询处理,也就能够在约简结构中得到相同的结果节点集合.

2 XML 索引应考虑的因素

2.1 XML查询的要求

不拘 XML 索引是何种形式,其实际设计与实现都必须考虑 XML 查询的基本特征——结构关系的保存以及基于结构信息快速计算节点间结构关系这两个因素.这实际上就是要求相关的技术能够满足高效处理 XML

查询的请求.实际研究中,XML 查询是一个范围很大的方面,内容很多.本文仅就 XML 索引为满足查询而应当具备的两个基本方面展开讨论,包括:

2.1.1 结构关系的获取

虽然传统的索引技术经过长期的积累已经相对成熟,但是,这类索引技术针对的主要是根据值(而不是具有某种关系的模式)定位数据记录的功能,不太关注数据记录间的逻辑关系;而 XML 数据查询的基本特征就是根据模式特征(正则路径表达式形式描述的结构关系)的输入提取符合该模式的数据,所以,XML 索引的主要内容就是设计适用于模式匹配的技术.自然,针对正则路径表达式的相关设计也就成为考虑的重点.

XML 查询中模式匹配的问题,如前面所述,主要就是对 XML 数据中结构关系的表达,以及如何利用已有的索引技术为高效获取符合结构关系[13](见前面所述的包含关系定义)的数据集提供支持的问题.

2.1.2 保 序

根据前面的叙述可知,元素在具体的 XML 文档中是有一定的次序限制的,该次序称为 XML 文档的次序.各元素在次序中出现的顺序取决于对 XML 树的先序遍历.当查询关系存储的 XML 分解数据时,由于关系模式不存在次序的概念,为此,就必须在分解 XML 的过程中考虑如何确保查询的结果仍然符合结果元素集在原 XML 数据中的次序关系,与这些元素在原 XML 文档先序遍历次序间有一一对应的关系,而且元素间的结构关系也必须一致.

2.2 XML数据修改的要求

2.2.1 修改有效性

要想真正实现对 XML 数据的管理,对 XML 数据进行修改是必不可少的功能.但是,到目前为止,关于 XML 数据修改问题还没有形成统一的标准.文献[14]对此作了探讨:给出了一些 XML 数据修改操作的定义,并就如何在 XQuery 之上进行数据修改功能的扩展做了有益的尝试.

XML 数据通常都会有相应的类型限制:当没有 XML 模式存在时,XML 必须是格式良好的;当存在一个相关联的 XML 模式时,XML 数据就必须满足模式规定的数据生成规则.所以,既然修改存在改变 XML 数据结构的可能性,那么,如何保证在修改之前确认这种修改不会触犯 XML 模式规定的规则,就成为 XML 数据管理研究的主要问题.

由于在 XML 处理中,XML 索引结构完全能够代替 XML 源数据,即 XML 索引结构完全能够满足查询操作的要求.所以,XML 索引结构同样应该支持这类限制.文献[15−17]给出了 XML 数据修改有效性的探讨,但是,文章主要是基于 XML 树以及结构摘要上的工作,并没有涉及对关系化存储的 XML 数据的修改、验证的问题.

与之相象但有本质不同的一个概念是 XML 类型检查问题(XML type checking problem).简单地说,类型检查是看一个 XML 数据是否符合 XML 模式的要求;而修改有效性则是判断一个 XML 数据生成过程是否能够保证生成满足 XML 模式的数据.

2.2.2 数据与索引结构的一致性

由 XML 数据修改操作引起的另外一个问题就是如何维护 XML 数据与 XML 索引间的一致性,即保证 XML 索引能够动态反映 XML 数据上的修改.而且,当存在 XML 的二级索引时,修改引起的连锁反应也必须在二级索引中体现.有关的概念在关系数据库研究中都有很好的阐述,在此不一一叙述.但是,由于 XML 数据保有结构信息的原因,这类一致性问题也体现出新的特点[18,19].

2.3 两点说明

2.3.1 节点记录索引的关系保存与 XML-关系映射存储的区分

在实际文献中,将 XML 数据管理与关系数据库管理系统结合,除了上述将节点记录索引保存为关系的方式以外,还有一类基于 XML 模式分解的 XML 数据关系保存方式[20−25].其基本思想就是根据 XML 模式中蕴含的限定 XML 数据结构的规则,将结对出现概率大的元素作为 RDBMS 中的关系(在有关的文献中称这种技术为内联(inlining)),从而实现了对 XML 模式的分解,并作为关系模式;之后,将 XML 数据分解保存到相应的关系表中.

虽然都是借助于关系数据库管理系统,但是二者的本质区别在于:节点记录索引的关系存储是针对结构查

询操作而来,每条索引记录都包含有便于结构查询的辅助信息——位置信息;而模式映射的关系存储方式是尽可能地将 XML 数据转换为符合关系规则的形式,不便于结构查询的处理.

2.3.2 XML 索引的结构层次

在分析 XML 索引研究内容时,比较混乱的是 XML 索引的界定问题.尤其是将 XML 数据交由关系数据库管理系统(RDBMS)管理时,原有 RDBMS 中开发的索引与 XML 索引间的关系,现有研究文献都没有进行讲解.本文建议将前面叙述的索引看作是 XML 索引结构中的第 1 层次,而原有数据管理系统的索引(甚至一些针对 XML 数据处理特点新研发的部分结构)都可看作是 XML 索引结构中的二级索引(secondary index),这样一来,有关 XML 索引的整体研究就有了清晰的把握.例如,文献[26]中的 Xp-tree 索引就是在区间编码的基础上引入空间数据访问的索引结构,即可看作 XML 二级索引的一种形式.比较常用的有经典的 B-tree, B+-tree 索引、哈希索引、R-tree 索引,以及延自于它们的各种演化形式索引.

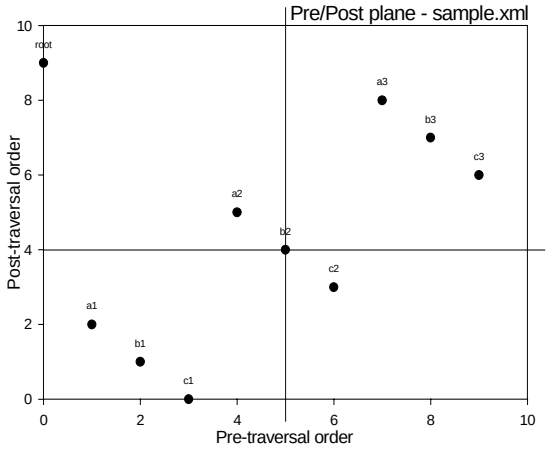

Fig.3 The node distribution of sample.xml on pre-post plane

图 3 sample.xml 中元素节点在 pre-post 平面上的分布

3 节点记录类 XML 索引

节点记录类索引本质上即是将 XML 数据分解为数据单元的记录集合,同时在记录中保存该单元在 XML 数据中的位置信息.主要有两种获取位置信息的方法,一种是节点序号方法(node numbering method,有时也称为节点标签方法,node labeling method),另一种是节点路径方法(node path method).对它们的研究占了 XML 数据处理研究文献中相当大的部分.根据路径信息表现形式的不同,节点记录类索引分为 3 大类:基于节点序号的索引、基于节点路径信息的索引和二者相结合的混合索引.

3.1 节点序号类索引

3.1.1 基本思想

这类索引中的位置信息是节点在某种遍历序列中的序号.对于一个 XML 文档,设计遍历 XML 文档的策略,遍历的最终结果表现为一个由节点组成的序列;相应地,节点的标签在序列中就有一个序号,该序号表明了节点间的次序关系,也能够反映节点间的结构关系,进而,就可以基于该序号信息捕获节点间的结构关系.

节点序号类 XML 索引的基本思想是:基于 XML 文档设计某种遍历策略,得到由元素组成的序列,节点的标签在序列中就具有唯一的次序,将序列与某指标集(最常用的就是自然数)建立一一映射的关系,对应序列中某个标签就有唯一的序号;对任意两个具有节点序号信息的节点,可以构建某种运算,该运算的结果可以表征节点间的结构关系,即{(独立单元属性,位置信息)}→{结构关系}映射.

根据标签序列生成方式的不同,目前研究文献中序号类索引可分为两种:基于标签有向树的遍历(如先序遍历和后序遍历)和基于字符流模型的顺序处理.根据映射指标集的不同,可分为赋以自然数[27−30]、赋以局部编码[31−35]和赋以素数[36]的方式.在序列生成和节点序号赋值的基础上,可以构建不同形式的位置信息,进而形成不同的节点记录索引形式.

3.1.2 相关概念:位置信息的不同形式

3.1.2.1 序号对形式

位置信息中最为常见的是所谓的区间编码[27,29](interval encoding)为序号对的形式.其基本思想是:将节点

在先序遍历序列和后序遍历序列中的序号对组成的区间(pre,post)作为该节点的位置信息.图 3 即为 sample.xml 中元素节点对应的序号对在 pre/post 平面上的分布.为了能够区分相同标签的节点,图中将 a,b 和 c 标签包含的文本信息作为该节点在图中的标签示意.图中以 b2 点为中心,作平行于 X 轴和 Y 轴的直线,就会将 Pre/post 平面分成 4 个区域,各区域的节点恰好是以 b2 为上下文节点满足如下 4 个轴关系节点的集合:b2 的祖先节点、b2 之后的节点、b2 之前的节点和 b2 的子孙.基于区间编码的基本索引记录形式为(s,pre,post),其中 s 为 XML 树中节点的标签属性.有如下的结论:

命题 **1**. 给定一棵 XML 树,XT,能够建立一组满足三元关系 $R\subseteq$String×Nat×Nat(Nat 为自然数)的实例 R_n,R_n 中的每一个元组(s,start,end)都满足如下条件:

1) s 是 XT 中节点的标签;

2) 对所有(s,start,end)$\in R_n$,start＜end;

3) 如果节点 s_1 和 s_2 在 XT 中呈祖先-子孙关系(s_1 是 s_2 的祖先节点),那么相应的元组(s_1,start$_1$,end$_1$)$\in R_n$ 和(s_2,start$_2$,end$_2$)$\in R_n$ 必然满足 start$_1$<start$_2$,end$_2$<end$_1$;

4) 如果 s_1 是 s_2 的兄弟节点,那么相应的元组(s_1,start$_1$,end$_1$)$\in R_n$ 和(s_2,start$_2$,end$_2$)$\in R_n$ 必然满足 end$_1$<start$_2$.

根据上面的命题,如果两个节点满足祖先-子孙关系或者是兄弟关系,那么,必有如上的数字不等式关系,这为查询路径处理中的最基本的二元结构关系比较提供了便利.但是,注意到结论的逆并不总是成立:仅可以由 start$_1$<start$_2$,end$_2$<end$_1$ 不等式确定对应的节点 s_1 和 s_2 是祖先-子孙的关系;s_1 和 s_2 之间是否属于父亲-孩子关系却无法确定,而且,仅由 end$_1$＜start$_2$ 不能确定 s_1 和 s_2 是否就是兄弟关系.为解决无法确定父亲-孩子关系的问题,许多文章引入节点在 XML 树中层的数字信息,记作 level.相应的四元组索引记录形式为(s,pre,post,level),通常称其为扩展区间编码;另一方面,为了解决兄弟关系的处理,在上述索引记录形式中往往还会再吸收节点的父亲节点的信息,扩展为五元组形式(s,pre,post,parent,level).

另外一种常用的区间形式为信息获取领域的 B-E-L 模式[37−42],其基本思路是:将标记的起、止标签在 XML 字符流中出现的序号组成数对,以之作为该标记的位置信息.其中的 B 表示标记的起始标签出现的次序,E 表示同一标记的终止标签出现的次序,而 L 表示标记距离根标记的嵌套层数.B-E-L 模式与区间编码有类似的性质.

3.1.2.2 素数形式

素数位置信息[36]的思想来自于判定两个整数是否具有整除关系的如下规则:

整数间非整除关系的判定规则(**indivisibility**). 如果整数 A 的素数因子中不存在整数 B 的素数因子,那么,称 B 对 A 是不能整除的.

例如,6 不可能是 10 的因子,因为 6 的素数因子“3”不是 10 的素数因子.

在 XML 树遍历过程中,为序列中每个标签节点序号对应的数字等于父亲节点分得的数与未用过的素数的乘积.根据上述判定规则,显然,只有同一路径上的节点序号才可能具有整除的关系.将这种关系与祖先-子孙的结构关系建立映射,这样,当给定两个具有上位置信息的节点记录时,根据它们的位置信息是否能够整除即可判定两个节点是否为祖先-子孙结构关系.这类索引记录的基本模式为(数据单元标识,素数序号).类似于区间编码的讨论,为支持其他结构关系,必须扩展上述基本模式.实际应用中多引入 level 或父节点的信息.

3.1.2.3 局部编码形式

在节点位置信息生成过程中,如果仅考虑该节点在兄弟节点中的次序,并显式地保留父节点的序号信息,则称其为局部编码方式,相应的位置信息形式为“父节点序号信息.本节点在兄弟节点集合中的序号”.

文献[32]给出了 Dewey 编码;文献[31]的 ORDPATH 编码采用的也是类似的方法,并给出了压缩 ORDPATH 的方法.文献[33]给出了称为结构持久编码的方法(persistent structural labeling scheme).该方法与前述的 Dewey 和 ORDPATH 相似之处在于都采用继承父节点编码的策略,只是采用的数为二进制形式,而不是自然数.其方法为:对每一个子节点,根据它在兄弟节点中的位置,赋给(相应个数−1)个 1,最后一位数为 0.例如,某节点在兄弟节点中处于第 4 个位置,那么,它的编码就应该是“L(u).1110”,其中,L(u)为父节点的编码.规定,根节点的编码为 0.

3.1.3 节点序号类索引概述

由于局部编码和素数编码的研究文献较少,相关的内容已在前面的概念讲述中有所涉及,本节仅就序号对

编码的索引展开叙述.

较早尝试将 XML 数据纳入 RDBMS 管理的方法是文献[41]中提出的 Edge 方法.该方法采用的数据模型近似 OEM,这一点与后续的基于 XML 树的相关方法不同.它的设计介于面向 XML 数据的映射和模式指导下关系映射之间:既延续了第 2.3.1 节中的模式映射的特点——需要构造复杂的关系模式来支持 XML 数据的查询;又吸收了树结构分解的方式.虽然文中讨论了对数据修改的支持,但是却没有深入探讨数据修改合法性的控制问题.文献[20]也具有类似的特点.文献[43]延续了 Edge 的形式,通过引入无损和有效两个概念从理论上系统地讨论了 XML 数据保存为关系表的有效性问题.可惜的是没有深入探讨基于 $Edge^{++}$之上的查询、修改等问题.

文献[22]给出了基于区间编码的 XML 索引方法.该方法基于 XPath 中给出的 XML 树模型.相关概念已在第 3.1.2 节给出.需要指出的是,基于这种模式实现路径模式的查询的基本方式只能是将路径分解为基于区间编码的比较.这一特点造成基于该方法的 XPath 处理性能不理想,尽管文中点明基于区间编码的 XPath 处理性能较 Edge 提高了 5 倍.为了解决这个问题,文献[31]给出了采用时空数据管理中开发的多维索引区间编码的组织索引记录,如 R-tree 和 RI-tree,但还是没有摆脱分段求解带来的问题.

与之相似的另一个广为应用的序号对编码方式,B-E-L,同样具有类似区间编码的缺陷.为了解决该问题,研究人员除了采用类似于区间编码中的解决办法——设计新的多维索引以外,如文献[44]的 RI-tree,文献[45]的 XR-tree,主要采取了优化连接操作的手段;文献[46]给出了多谓词合并连接(multi-predicate merge join,简称 MPMGJN)的算法,文献[40]给出了位置统计图连接(pH-join:position histogram join)算法,文献[47]给出了基于 B+-tree 的结构连接方法,文献[48]则分别讨论了区间模型(interval model)和位置模型(position model)上的结果集合估计算法以帮助选择较优的连接次序,以及文献[49]的基于划分的连接算法.但是,这些改进都没有摆脱结构操作是最小二元关系的本质,所以,其 XPath 处理的实际性能不能达到最优.鉴于 RDBMS 中 LIKE 语句的潜力,许多研究尝试了将二者混合的方式.

3.1.4 节点序号类索引分

3.1.4.1 XPath 查询处理

与结构摘要类索引不同,节点记录类索引打破了必须通过标签路径查找节点这一限制,将 XML 数据分解成规范形式的节点记录.由于保存了节点的位置信息,而且能够很好地结合到成熟的关系数据库管理系统中.基于节点序号类索引的查询路径处理,根据编码的不同分为 3 类.

序号对编码的查询处理通常分成两个阶段:第 1 个阶段是将查询语句分解为基本的二元结构关系运算形式,即只利用索引记录中的位置信息处理两节点间的结构关系判断;第 2 个阶段是将第 1 阶段的中间结果进行合并.实际处理中多采用树形结构组织查询的处理.虽然任何查询语句都可以基于这类索引记录得到准确的结果,但是,正如前面的概述所述,基于这类索引的查询处理往往导致较多的连接操作,无法达到较高的性能.这在进行复杂语句的处理时更为突出.

基于局部编码的查询处理,由于无法摆脱耗时的数字序列模式匹配操作,基于它们处理 XPath 的研究相对要少得多,但是,由于这类编码隐含地保存了路径与内部节点间的关系,反而在新发展的 XML 信息获取处理中找到了进一步研究的价值(参见第 5.2.1 节).文献[50]在引入 FST(finite state transducer)模块后探讨了 Dewey 编码在 Twig 处理上的优势,并进一步将序号对编码引入到处理机制中,以减少结构关系计算的代价.

至于素数编码,相关的文献只有文献[36].虽然这种编码在结构关系的表达上具有简洁的数学上的映射,但是,当数据规模较大时,就需要系统提供很大的素数,这在实际使用时是不方便的:一是需要较大的保存空间,二是素数的关系运算也会有较大的开销.

这类索引都能够支持 XML 查询对保序的要求.

3.1.4.2 数据修改的支持

针对第 2.2 节中给出的 XML 数据修改支持的要求,在现有节点序号类索引中,文献[32]给出了将实数作为序号的指标集的思路:利用实数可表示无穷精度的特点满足插入数据带来的节点序号的扩张.但是,由实数计算带来的结构关系计算的代价是提高查询处理性能所不容忽视的.文献[33]的扩展方式类似于哈希桶的成倍分裂方式,但是,当 XML 树的扇出很大时,标签的标示会变得很大,带来了冗余.文献[3,43,51]给出了预留空间的处理

方法.文献[36]通过维护同余表(SC table:simultaneous congruence table)实现了对保序和数据更新的支持,但是,当新节点插入时,重新计算同余表的代价也是不容忽视的.文献[52]采取了动态编码的技术,引入了 W-BOX 和 B-BOX,具有较好的理论分析和实验的结果.

至于对修改合法性检验的支持,在这类索引中还没有开展相关的研究.但是,从数据管理的角度对这类形式的 XML 数据的修改的支持是非常必要的.

3.2 节点路径类索引

3.2.1 基本思想

节点的路径信息同样蕴含节点在XML数据中结构的信息:如果给定两节点的路径信息,同时预知两节点存在结构关系的情况下,就必然可以获知它们之间的结构关系.即节点A的标签路径包含节点B的标签路径,那么,在 XML 树中 A 和 B 之间一定具有祖先-子孙的结构关系,且 B 是 A 的祖先.所以,基于路径信息来获取节点的结构关系就成为另一组 XML 数据处理技术的思路来源,并演化出基于节点路径的 XML 索引[53−58].这类索引的核心技术是字符串的模式匹配.因而,这类索引记录数据的管理方法,有许多是来自于信息获取领域的.例如,Trie,Patricia trie 以及 Suffix tree,甚至 Suffix array 等结构.索引记录的基本模式为(数据单元标识,路径信息).由于这类索引的大部分处理模式与传统的模式匹配有很深的关系,在此不再展开叙述.仅对基于路径的多维索引的概念进行简单的叙述.

3.2.2 相关概念:路径多维索引-UB-tree

在路径索引记录数据的管理中,文献[53]突破了路径匹配只能基于字符串匹配的限制,借助于 Z-地址和 Z-区间的概念,通过对路径的转换,引入了 UB-tree 结构,实现了借助于转换对路径的多维管理.

定义 **1**. Z-地址.

设Ω表示 n 维空间,Ω空间中每条记录的第 i 个属性 A_i 具有 s 个可能的取值,表示为 $A_i=A_{i,s-1},A_{i,s-2}\wedge A_{i,0}$,那么,对$\Omega$空间中的任一记录 $O\in\Omega$,都有一个唯一的 Z-地址函数 $Z(O)$与之对应:

$$Z(O)=\sum_{j=0}^{s-1}\sum_{i=1}^{n}A_{i,j}2^{jn+i-1}.$$

利用 Z-地址,可以构建一个从 n 维空间到 1 维 Z-地址空间的映射.n 维空间中的每一点都对应 Z 地址空间的一个区间$\langle\alpha,\beta\rangle$,称为 Z-区间(Z-region).以 B+-tree 对 Z-区间集合中的数据做索引,就成为 UB-Tree 的基本思想.当将 XML 路径集合看作是某个 n 维空间(取 XML 数据中最长路径的长度作为维度 n 的值)中的一个实例时,就可以实现将 XML 路径到 Z-地址空间的转换,进而可以实现用 UB-tree 来对 XML 路径信息进行索引的目的.

虽然这种基于转换的多维管理结构在转换代价和路径转换空间冗余两个方面存在不足,但是,由于多维索引的概念能够突破对字符串只能进行连续模式匹配的方式,所以,在这个方向上作进一步的研究具有实际的意义.

3.2.3 节点路径类索引概述

文献[57]直接保存了节点标签路径的字符串,并利用了 SQL LIKE 的模式匹配实现对路径查询的处理.文献[56]则利用了可将 XML 文档的节点路径标签字符串和查询路径字符串都分解为后缀片断的特性:将文档的后缀片断用树结构组织,之后将查询路径对应的片断在文档树上匹配从而得到最后的结果.文献[54]的思路与之类似,只是将后缀换成了 Prüfer 编码.

3.2.4 节点路径类索引概述分析

3.2.4.1 XPath 查询处理

基于节点路径信息的查询处理,与节点序号类索引的查询处理是类似的,只是其中的结构关系操作换作路径字符串集合中对查询路径的匹配:将节点路径记录交由关系数据库管理系统管理时,通过巧妙的设计,直接利用 SQL 中的模式匹配功能(LIKE '%')实现查询路径中的祖先-子孙关系,如文献[57]中将 path 写成"a1#/a2#/a3#/a4#/a5"的形式.这样,所有满足"//a3//a5"结构的节点的 SQL 语句都可以写成"LIKE '#%/a3#%/a5'"的形式.尽管路径标签字符串能够满足 XPath 路径模式的匹配,但缺乏保序支持的能力,所以,在实际中多采取与

序号相结合的混合方式.

至于新颖的多维路径索引——UB-tree,在面向实际应用时还存在如下的缺陷:在处理查询时,必须经过字符串到 Z 地址空间的转换,其代价是不容忽略的;实际存储中,必须预先给定标签路径长度的最大值,这就造成存储的冗余,也为 Z-地址的计算增加了不必要的工作.尽管如此,作为新的处理方式,标签路径的多维索引方式仍然是颇具吸引力的方式.

3.2.4.2 数据更新支持

在这类索引的文献中,未见支持 XML 数据修改的探讨.

3.3 混合索引

同时保留节点的序号信息以及节点的路径作为索引记录的位置信息,就构成了此处的混合索引[38,59,60].

这类索引记录的模式多为多个模式的组合,基本的模式为(数据单元的标识,路径 ID,序号位置信息)和(路径 ID,标签路径)的组合.例如,文献[38,59]都采用了标签路径与序号对结合的方式.与之不同的是,文献[60]采取了将后缀树[56]与序号对结合的方式.

混合索引吸收了节点记录索引和路径索引的优点:既具有路径模式匹配是不必分解为最小片断的要求,同时还能够保证得到的结果符合保序的要求.但是,在支持 XML 数据修改问题上鲜见有益的探讨.

4 结构摘要类 XML 索引

4.1 基本思想

以 XML 树结构中节点的路径信息为基础,采取某种约简方式,使得约简后的树结构只维护不同的路径信息,而不会存在具有相同路径的两个节点.这里,结构摘要索引仍然采取标签有向图的结构.当基于结构摘要进行 XML 查询处理时,就可以避免第 1 节所述的基本 XML 查询处理的另一个缺陷.

4.2 相关概念

定义 **2**. 结构摘要.

给定标签有向图 $G=(V_G,E_G,root_G,\Sigma_G)$,$G$ 的结构摘要是基于定义在 V_G 上的一个等价关系上的标签有向图 $I_G=(V_{I(G)},E_{I(G)},root_{I(G)},\Sigma_G)$.其中 $V_{I(G)}$的节点 n 称作索引节点,对应于 V_G 中满足等价关系的节点集合,即 n.extent;并且,$E_{I(G)}$中存在一条边(u_i,v_i),当且仅当对应于 E_G 中存在一条边(u_d,v_d),并且 $u_d\in u_i$.extent,$v_d\in v_i$.extent.

在这类结构摘要索引中,为了达到简化 XML 树的目的,最初多采用自动机理论中的 NFA 到 DFA 转化的思路[62].在文献[63]中引入了双拟和双似的概念以后,由于双似概念能够准确地表述结构摘要问题,所以,后续的这类索引多基于这个概念[64−69].

定义 **3**. 双拟关系(Bisimulation).

给定一棵有向树 G,$G=(V,E)$,其中 V 是 G 中节点的集合,E 是 G 中边的集合;对 G 进行转换,可以得到另外一种表达形式 $G'(V',E')$,它与原来的 G 应具有如下的关系:G'的点元素集合 V'与 G 中的 V 是一样的,同时 G'元素间的关系 E'与 G 中的 E 是一致的,但是二者的表现形式是不同的;否则,$G'=G$.基于 G 和 G'可以定义一个二元关系 R,它是 G 中节点 V 和 G'的节点 V'的笛卡尔积:$R=\{(p,q)|p\in V$,并且 $q\in V'\}$.如果 R 满足如下条件,则称 R 是 G 和 G'的一个双拟关系:

1. 如果在 G 中存在一个节点 p',它与 p 具有关系 $p\xrightarrow{\alpha}p'$.

即,在 G 中有边α将 p 和 p'连结,那么,在 G'中必也存在一个元素 q',它与 G'的元素 q 存在关系.

2. 类似地可以定义 G'中元素关系到 G 中节点间关系的映射:

如果在 G'中存在一个节点 q',使得从 q 到 q'经过元素间关系α,即 $q\xrightarrow{\alpha}q'$,那么,在 G 中必然也有一个节点 p',使得从某一节点 p 到 p'也有α关系,即 $p\xrightarrow{\alpha}p'$.

定义 **4**. 双似关系(bisimilarity).

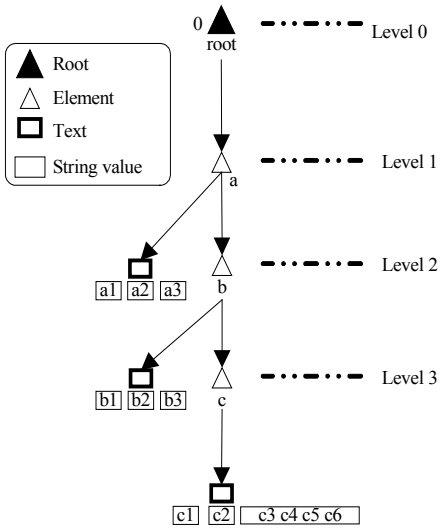

Fig.4 Structural summary of sample.xml by bisimilarity equivalence relation

图 4 sample.xml 的基于双似等价关系的结构摘要

如果 G 和 G'的两个节点 p,q 间存在双拟关系,那么称 p 和 q 是双似的(bisimilar),记作 $p\approx^{b}q$.由前面双拟的定义可知,两个元素双似是等价的关系.显然,两个节点满足双拟的关系,那么它们必然具有相同的标签路径.

图 4 即为 sample.xml 文档的基于双似等价概念的结构摘要结构.从图中可以看出,结构摘要只保留了唯一的路径,具有相同路径的文本数据都集中在该路径的节点之中.显然,只搜索一条路径就可以获取具有相同路径的所有节点.

考察到 XML 查询语句中更多出现的是结构片段查询,而不是整个从根节点开始的简单路径,所以,文献[62]放松了双似的限制,引入 k 阶双似的概念,捕捉 XML 数据中路径的局部特征以满足结构片段查询的特点.

定义 **5**. K 阶双似(K-similarity).

1) 对标签有向图 G 中的任意两个节点 u 和 v,如果它们具有 $u\approx^{0}v$,当且仅当 u 和 v 具有相同的标签.

2) 对标签有向图 G 中的任意两个节点 u 和 v,如果它们具有 $u\approx^{k}v$,当且仅当 u 和 v 的父亲节点 u'和 v'具有 $u\approx^{k-1}v$.

K 可以看作是控制整个索引结构的分辨率参数.

在实际应用中,如何在 G 中寻找节点的最大 K 阶相似性,即如何以尽可能大的相似性划分 G 的节点集合,成为这类索引的核心技术.

4.3 结构摘要索引概述

Stanford 大学的 Lore 项目中的 DataGuide[61]是结构摘要的最初形式,其基本思想是基于将 NFA 转换为 DFA 的形式对 XML 数据中的相同路径进行约简.这种基于处理的阐述缺乏形式的描述,不能很好地描述结构摘要的本质特征.1-index[62]引入了双似等价关系的概念,从而对结构摘要找到了形式化描述的理论基础.

DataGuide 和 1-index 能够很好地满足简单路径查询的要求,但在实现性能上稍显不足.为此,APEX[64]引入了依赖于 XML 数据查询分布的信息:将经常出现的 XML 查询语句对应的标签节点预先保存在一个哈希结构中.它的作用类似于 Cache 的功能:当有新的查询要求处理时,首先在哈希表中搜索是否有满足的节点集合.

为了增加对结构摘要树便利的灵活性,文献[65]进一步提出了 F&B 索引,并声称 F&B 索引是同类索引中回答 Twig 查询存储空间最小的数据结构,它可以满足全部分支路径查询的要求.但是,由于 F&B 索引在实现上必须依赖内存,而且在基于它处理查询的优化上还存在进一步的空间,这些因素都限制了 F&B 索引在实际中的应用.针对这个问题,演化出两类解决方案,一是所谓近似结构摘要索引,如 Xsketch,$A(K)$-index,$D(K)$-index,以及 $M(K)$-index;另一个就是文献[69]提出的基于磁盘存储的并引入序号对的 F&B 索引.

在 K 阶双似概念的基础上,类似于 DataGuides,可以创建相应阶的 XML 索引.$A(K)$-index[63]索引的创建过程是一个逐渐细化的过程:K 阶索引的建立是在(k–1)阶索引的基础上实现的.在 1-index[62]和 $A(K)$-index 的基础上,$D(K)$-index[67]吸收了 APEX 中动态反映查询分布的优点,将二者进行了结合,并具有如下的特征:对任意两个节点 u 和 v,如果 u 和 v 之间存在一条边,那么,$K(u)\geq K(v)-1$.其中,$K(u)$和 $K(v)$分别表示节点 u 和 v 的局部相似性(local similarity),即父节点的 K 值不应小于孩子节点的 K 值–1.$M(K)$-index[69]的索引结构与 $D(K)$-index 类似,在实际应用时,同时尝试随查询语句动态调整的方式.但是,这种类近似结构摘要索引只是部分减轻了索引空间以及查询处理效率的问题.

文献[69]中的 F&B 索引一方面实现了将结构摘要树保存于磁盘,而且,为了提高数据访问的效率,另一方面,

还深入探讨了在保存中引入聚集机制的方法,以及将编码机制(文中采用的是 B-E-L)结合进 F&B 索引的方案.但是,由于没有超越通过摘要树遍历得到查询结果的处理方式,其实际应用中的性能仍需要实践的结果来证明.

Table 1 Summarization for XML indices

表 **1** XML 索引综合概述表格

Scheme name	Ref.	Year	Class	Subclass	Sub-subclass	Query processing		Modification/Validation support	
						Basic computation	Ordering support	Modification	Validation
DataGuides	[61]	1997	Structural summary style index	–	–	Navigation on summary tree	No	Incremental maintenance	No discussing
1/T-index	[62]	1999						No discussing	
A(*K*)-index	[63]	2002							
APEX	[64]	2002						Incremental maintenance	
F&B index	[65]	2002						No discussing	
XSketch	[66]	2002							
D(*K*)-index	[67]	2003						Yes	
M(*K*)-index	[68]	2004						No discussing	
Disk-Based F&B	[69]	2005					Yes		
Edge	[41]	1999	Node record style index	Node numbering index	Sequence number pair	Unequal join & Mediate result merge	Yes	Straightforward SQL series	No discussing
Interval encoding	[27]	2002						Extended preorder and range	
Order-Size	[21,52]	2003							
B-E-L	[37–42,45]	1999~2003							
Edge^{++}	[52]	2005						No discussing	
Prime	[36]	2004			Prime	Prime modulo		SC table	
Dewey	[32]	2002			Local encoding	Number sequence matching		No discussing	
ORDPATH	[31]	2004							
PSL	[33]	2004						ρ-tight subtree	
Extended Dewey	[50]	2005				FST transducing		No discussing	
UB-Trees	[53]	2001~2002		Node path index	Multi-dim	Transfor-mation & Range computation	No discussing	No discussing	
XParent	[56]	2002			Label path	Path string matching	Yes	Dynamic scope allocation	
ViST	[55]	2003				Suffix tree scan	No		
PRIX	[54]	2004				Prüfer sequence matching	Yes	No discussing	
XRel	[59]	2001		Hybrid	–	Path string matching	Yes	No discussing	No discussing
Extended inverted index	[38]	2003							
BLAS	[60]	2004				Suffix tree scan			

4.4 结构摘要索引分析

4.4.1 XPath 查询处理

基于结构摘要的 XML 查询语句处理,与第 1.1 节中所述的在 XML 文档之上的 XPath 查询处理是相同的:扫描路径获取满足查询路径的节点集合.由于结构摘要已经尽可能地合并了相同标签路径的节点,所以,周游一条路径即可得到该标签路径下的所有节点,避免了相同标签路径的重复访问的缺陷,与直接基于 XML 原始数据匹配查询路径模式相比,性能有了大幅度的提高.而且,由于它维护了整体的信息,基于该结构就可以比较容易地实现对 XML 数据修改的支持.

但是,结构摘要虽然约简了路径信息,对查询路径的匹配本质上仍然是周游的方式,这就需要维护整体的数

据结构,难以适用于大规模数据;而且,由于只是保存约简后的树结构,对保序的支持就难以实现.

4.4.2 对数据修改的支持

针对数据修改的问题,结构摘要类索引中有文献[61−63],它们都试图寻找一个结构摘要类索引通用的增量式修改方法,基本思路可概括为:在发生数据修改的部分,根据修改的类型不同插入新数据,或删除某个节点,对涉及的节点分别进行分割(split)或合并(merge)处理.本质上,这种修改的支持还只是数据与索引间的一致性问题.

5 XML 索引技术总结及研究展望

5.1 XML索引技术总结

对 XML 数据管理的研究,是当前诸多领域的热点.有鉴于索引技术在数据管理中的突出地位,众多的 XML 文献也将研究集中到了 XML 索引的技术.如何捕获 XML 数据中的结构特征,并高效地支持路径(结构)查询的处理,是其中的核心.在给出 XML 索引应考虑的限制后,本文对当前分布于众多研究文献中的 XML 索引进行了汇总,按照解决现有基于 XML 文档进行 XML 查询存在的两类缺陷的方式的不同,将繁多的 XML 索引归入两大类:结构摘要类索引和节点记录类索引.并通过 XML 数据结构信息的捕获、如何高效处理结构关系的查询,以及索引对 XML 数据修改的支持 3 个方面进行了阐述.表 1 是全文内容的总结,包括索引的类别、查询性能以及对 XML 数据修改的支持.

5.2 研究展望

如果说 XML 数据可以看作由结构部分(即元素见结构关系的集合)和内容部分(包括元素属性的值和元素中所含文本的集合)组成,那么,当前有关 XML 数据处理的研究基本上属于这样一种模式:给定表征结构关系的查询路径,搜索 XML 数据空间寻找匹配结构模式的内容集合.也就是说,指到目前为止,经典的 XML 查询都是基于用户已经了解 XML 数据的组织结构为前提的.显然,这一要求大大提高了对普通用户搜索 XML 信息的要求,更不用说要求用户掌握查询语言的语法规则.为此,可以采取两种思路:一种是设计辅助功能,方便地帮助用户了解 XML 数据的组织结构;另一种是 XML 的信息获取.除此之外,XML 数据修改合法性验证是另一个必须关注的方向.

5.2.1 同时满足 XML Query 和 XML IR 效率要求的索引

如果说类似于 XPath 的查询处理,即首先精心定义满足 XML 内容和结构要求的查询语言,之后搜索 XML 数据、匹配满足使用查询语言描述的模式从而得到最后的结果,是近期 XML 数据处理的主要研究内容,那么,将传统信息获取中的优点纳入 XML 数据的处理——希望提供给用户一种友好的使用形式,成为当下 XML 数据处理的新的研究点.近期的研究可分为两个方面:一方面是扩展前述面向模式匹配的查询语言(如 XQuery)以支持部分 XML IR 中的特点[70−75];另一方面是将对用户使用友好的关键字查询延伸至 XML 数据[39,76−79].其中,由于第 1 种结合不能摆脱原有查询语言的基本要求——仍然要求用户必须知晓 XML 数据的结构模式以及掌握复杂的语法,所以,本质上不符合 XML IR 的要求.而对 XML 数据的关键字查询,其基本的操作就是找到满足关键字的 SLCA 节点(smallest lowest common ancestor)[78],针对这一问题多采用 Dewey 编码.而我们知道,Dewey 编码对 XML Query 处理的性能是不足的,另外,基于 Dewey 求解 SLCA 节点的基本操作主要是数字序列的模式匹配,效率堪忧,所以,设计既能够保证 XML Query 查询的性能,又能够确保 XML IR 处理效率的索引,就成为值得深入探讨的另一个方向.

5.2.2 内部结构信息的获取

与 XML 信息获取思路不同,将 XML 数据查询引向 XML 数据的另一部分——结构特征,就会得到具有实际意义的处理方式:给定 XML 数据片断(可以是元素文本内容中的一个字符,也可以是一段简单的路径信息),搜索 XML 数据的结构部分,寻找所有指向给定片段的路径集合.显然,这种处理方式对于引导那些不知道 XML 结构信息的用户进一步寻找更为详尽的目标是很有帮助的.粗看起来,似乎前面所述的基于节点路径信息的索引方式完全能够胜任,但是,由于上述结构中的路径是完整的字符串形式,不能够方便地提供路径内部的统计信

息,因而无法胜任诸如“以给定片断为核心,半径为 3 个字符的路径集合”类的查询.从这一角度来说,从信息获取领域引入到 XML 数据处理范畴的关键字查询[39,78]也是不敷使用的.而另外一大类节点记录索引则由于没有保存路径有关的信息,虽然能够满足传统给定查询路径寻找内容集合的方式,但不适于这里提到的逆向查询的处理.

5.2.3 节点记录索引对数据修改合法性的支持

仅仅是最近几年,尤其是 2003 年,2004 年,对 XML 数据修改合法性层次的支持才在几个会议中有了相关的讨论[15−17,80,81].与以前的 XML 文档有效性验证(validation of XML documents)[82]不同,这些文章大多采用增量有效性验证的概念(incremental validation of XML documents):

给定 XML 模式(DTD 或 XML Schema 的形式)τ,一个满足 τ所规定的限制条件的 XML 树 T,$T\in\text{sat}(\tau)$(表示 T 满足 τ的限制),以及将 T 转换为 T'的修改的操作序列 S,增量验证的问题就是如何设计一个检验机制 M,它能够有效检验 $T'\in\text{sat}(\tau)$.通常所指的修改操作包括如下类型:

(1) 替换指定节点的标签;

(2) 在指定节点后插入新的叶节点;

(3) 在指定节点中插入一个节点作为该节点的第 1 个孩子;

(4) 删除指定的叶节点.

显然,上述的修改操作没有涉及递归删除、插入等形式.为解决这个问题,上述文章采取了不同的方法,除了文献[80]采取了在修改语句中增加条件函数以外,其他解决方法都有一个共同点,就是借助于不同形式的自动机对字符串的接收、拒绝判断的能力.但是,所有这些修改合法性验证的方法都不能自然的扩展到节点记录类索引方式.如何设计针对节点记录类索引形式的 XML 数据修改合法性验证机制,因而具有实际的意义.

References:

[1] Meng XF, Zhou LX, Wang S. State of the art and trends in database research. Journal of Software, 2004,15(12):1822−1836 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/12/1822.htm

[2] Barashev D, Novikov B. Indexing XML to support path expressions. In: Manolopoulos Y, Návrat P, eds. Proc. of the 6th East European Conf. on Advances in Databases and Information Systems (ADBIS). Bratislava: Springer-Verlag, 2002. 1−10.

[3] Li QZ, Moon B. Indexing and querying XML data for regular path expressions. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB). Roma: Morgan Kaufmann Publishers, 2001. 361−370.

[4] Cooper BF, Sample1 N, Franklin MJ, Hjaltason GR, Shadmon M. A fast index for semistructured data. In: Apers PMG, Atzeni P, Ceri S, Paraboschi S, Ramamohanarao K, Snodgrass RT, eds. Proc. of the 27th Int'l Conf. on Very Large Data Bases (VLDB). Roma:Morgan Kaufmann Publishers, 2001. 341−350.

[5] Zhang C. Relational databases for XML indexing [Ph.D. Thesis]. Wisconsin: University of Wisconsin-Madison, 2002.

[6] Bruno N, Koudas N, Srivastava D. Holistic twig joins: Optimal XML pattern matching. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 310−321.

[7] Amer-Yahia S, Cho S, Lakshmanan LVS, Srivastava D. Minimization of tree pattern queries. In: Aref WG, ed. Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Santa Barbara: ACM Press, 2001. 497−508.

[8] Jiang HF, Wang W, Lu HJ, Yu JX. Holistic Twig joins on Indexed XML documents. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 273−284.

[9] Chen ZY, Jagadish HV, Korn F, Koudas N. Counting twig matches in a tree. In: Young DC, ed. Proc. of the 17th Int'l Conf. on Data Engineering (ICDE). Heidelberg: IEEE Computer Society, 2001. 595−604.

[10] Lee DW, Srivastava D. Counting relaxed twig matches in a tree. In: Lee YJ, Li JZ, Whang KY, Lee DH, eds. Proc. of the 9th Int'l Conf. on Database Systems for Advances Applications (DASFAA). LNCS 2973, Springer-Verlag, 2004. 88−99.

[11] Jagadish HV, Al-Khalifa S. TIMBER: A native XML database. The VLDB Journal, 2002,11(4):274−291.

[12] Fiebig T, Helmer S, Kanne CC. Anatomy of a native XML base management system. The VLDB Journal, 2002,11(4):292−314.

[13] Miklau G, Siciu D. Containment and equivalence for a fragment of XPath. Journal of the ACM, 2004,51(1):2−45.

[14] Tatarinov I, Ives ZG, Halevy AY, Weld DS. Updating XML. In: Aref WG, ed. Proc. of the 2001 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Santa Barbara: ACM Press, 2001. 413−424.

[15] Papakonstantinou Y, Vianu V. Incremental validation of XML documents. In: Calvanese D, Lenzerini M, Motwani R, eds. Proc. of the 9th Int'l Conf. (ICDT). LNCS 2572, Siena: Springer-Verlag, 2003. 47−63.

[16] Barbosa D, Mendelzon AO, Libkin L, Mignet L. Efficient incremental validation of XML documents. In: Titsworth F, ed. Proc. of the 20th Int'l Conf. on Database Engineering (ICDE). Boston: IEEE Computer Society, 2004. 671−682.

[17] Abrao MA, Bouchou B, Ferrari MH. Incremental constraint checking for XML documents. In: Bellahsene Z, Milo T, Rys M, Suciu D, Unland R, eds. Proc. of the 2nd Int'l XML Database Symp. (Xsym), Database and XML Technologies. LNCS 3186, Springer-Verlag, 2004. 112−127.

[18] Kaushik R, Bohannon P, Naughton JF, Shenoy P. Updates for structure indexes. In: Bressan S, Chaudhri AB, Lee ML, Yu JX, Lacroix Z, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases (VLDB). LNCS 2590, Hong Kong: Morgan Kaufmann Publishers, 2002. 239−250.

[19] Yi K, He H, Stanoi I, Yang J. Incremental maintenance of XML structural indexes. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 491−502.

[20] Deutsch A, Fernandez M, Suciu D. Storing semistructured data with STORED. In: Haas LM, Tiwary A, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD'99). Philadelphia: ACM Press, 1999. 431−442.

[21] Shanmugasundaram J, Tufte K, He G. Relational databases for querying XML documents: Limitations and opportunities. In: Atkinson MP, Orlowska ME, Valduriez P, Zdonik SB, Brodie ML, eds. Proc. of the 25th Int'l Conf. on Very Large Data Bases (VLDB). Edinburgh: Morgan Kaufmann Publishers, 1999. 302−314.

[22] Kanne CC, Moerkotte G. Efficient storage of XML data. In: Young DC, ed. Proc. of the 16th Int'l Conf. on Data Engineering (ICDE). San Diego: IEEE Computer Society, 2000. 198.

[23] Klettke M, Meyer H. XML and object-relational database systems-enhancing structural mappings based on statistics. In: Suciu D, Vossen G, eds. Proc. of the Int'l Workshop on the Web and Databases (WebDB). LNCS 1997, Dallas: Springer-Verlag, 2000. 151−170.

[24] Schmidt AR, Kersten ML, Windhouwer M, Wass F. Efficient relational storage and retrieval of XML documents. In: Suciu D, Vossen G, eds. Proc. of the Int'l Workshop on the Web and Databases (WWW). LNCS 1997, Amsterdam: Springer-Verlag, 2000. 137−150.

[25] Bohannon P, Freire J, Roy P, Siméon J. From XML schema to relations: A cost-based approach to XML storage. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering (ICDE). San Jose: IEEE Computer Society, 2002. 64−92.

[26] Hwang JH, Nguyen VT, Ryu KH. A new indexing structure to speed up processing XPath queries. In: Zhou LZ, Ooi BC, Meng XF, eds. Proc. of the 10th Int'l Conf. on Database Systems for Advanced Applications (DASFAA). LNCS 3453, Trondheim: Springer-Verlag, 2005. 900−906.

[27] Grust T. Accelerating XPath location steps. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 109−120.

[28] DeHaan D, Toman D, Consens MP. A comprehensive XQuery to SQL translation using dynamic interval encoding. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 623−634.

[29] Grust T, Teubner J. Accelerating XPath evaluation in any RDBMS. ACM Trans. on Database Systems, 2004,29(1):91−131.

[30] Amagasa T, Yoshikawa M, Uemura S. QRS: A robust numbering scheme for XML documents. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering (ICDE). Bangalore: IEEE Computer Society, 2003. 705−707.

[31] O'Neil P, O'Neil E, Pal S, Cseri I, Schaller G. ORDPATHs: Insert-Friendly XML node labels. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 903−908.

[32] Tatarinov I, Viglas SD. Storing and querying ordered XML using a relational database system. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 204−215.

[33] Cohen E, Kaplan H, Milo T. Labeling dynamic XML trees. In: Popa L, ed. Proc. of the 21st ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems (PODS). Madison: ACM Press, 2002. 271−281.

[34] Han ZM, Fu NY. Efficiently coding and querying XML document. In: Bhalla S, ed. Proc. of the 4th Int'l Workshop on Databases in Networked Information Systems (DNIS). LNCS 3433, Springer-Verlag, 2005. 54−69.

[35] Han ZM, Xi CT, Le JJ. Efficiently coding and indexing XML document. In: Zhou LZ, Ooi BC, Meng XF, eds. Proc. of the 10th Int'l Conf. on Database Systems for Advanced Applications (DASFAA). LNCS 3453, Beijing: Springer-Verlag, 2005. 138−150.

[36] Wu XD, Lee ML, Hsu W. A prime number labeling scheme for dynamic ordered XML trees. In: Proc. of the 20th Int'l Conf. on Database Engineering (ICDE). Boston: IEEE Computer Society, 2004. 66−78.

[37] Runapongsa K. Methods for efficient storage and indexing in XML databases [Ph.D. Thesis]. Michigan: University of Michigan, 2003.

[38] Seo CY, Lee SW, Kim HJ. An efficient inverted index technique for XML documents using RDBMS. Information and Software Technology, 2003,45(1):11−22.

[39] Florescu D, Kossmann D, Manolescu I. Integrating keyword search into XML query processing. http://www9.org/w9cdrom/index.html

[40] Wu YQ, Patel JM, Jagadish HV. Estimating answer sizes for XML queries. In: Jensen CS, Jeffery KG, Pokorný J, Saltenis S, Bertino E, Böhm K, Jarke M, eds. Proc. of the 8th Int'l Conf. on Extending Database Technology (EDBT 2002). Prague: Springer-Verlag, 2002. 590−608.

[41] Florescu D, Kossman D. A performance evaluation of alterative mapping schemes for storing XML in a relational database. Technical Report, No. 3680, INRIA, France, 1999.

[42] Al-Khalifa S, Jagadish HV, Koudas N, Patel JM, Srivastava D, Wu Y. Structural joins: A primitive for efficient XML query pattern matching. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering (ICDE). San Jose: IEEE Computer Society, 2002. 141−152

[43] Barbosa D, Freire J, Mendelzon AO. Designing information-preserving mapping schemes for XML. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC. eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 109−120.

[44] Kriegel HP, PStke M, Seidl T. Managing intervals efficiently in object-relational databases. In: Abbadi AE, Brodie ML, Chakravarthy S, Dayal U, Kamel N, Schlageter G, Whang KY, eds. Proc. of the 26th Int'l Conf. on Very Large Data Bases (VLDB). Cairo: Morgan Kaufmann, 2000. 407−418.

[45] Jiang HF, Lu HJ, Wang W, Ooi BC. XR-Tree: Indexing XML data for efficient structural joins. In: Dayal U, Ramamritham K, Vijayaraman TM, eds. Proc. of the 19th Int'l Conf. on Data Engineering (ICDE). Bangalore: IEEE Computer Society, 2003. 253−264.

[46] Zhang C, Naughton J, DeWitt D, Luo Q, Lohman G. On supporting containment queries in relational database management systems. In: Aref WG, ed. Proceedings of the 2001 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Santa Barbara: ACM Press, 2001. 425−436.

[47] Chien SY, Vagena Z, Zhang D, Tsotras VJ, Zaniolo C. Efficient Structural Joins on Indexed XML Documents. In: Bressan S, Chaudhri AB, Lee ML, Yu JX, Lacroix Z, eds. Proc. of the 28th Int'l Conf. on Very Large Data Bases (VLDB). LNCS 2590, Hong Kong: Morgan Kaufmann, 2002. 263−274.

[48] Wang W, Jiang HF, Lu HJ, Yu JX. Containment join size estimation: Models and methods. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 145−156.

[49] Wang J, Meng XF, Wang S. Structural ioin of XML based on range partitioning. Journal of Software, 2004,15(5):720−729 (in Chinese with English). http://www.jos.org.cn/1000-9825/15/720.htm

[50] Lu JH, Ling TW, Chan CY, Chen T. From region encoding to extended dewey: On efficient processing of XML twig pattern matching. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC. eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 193−204.

[51] Harding PJ, Li QZ, Moon B. XISS/R: XML indexing and storage system using RDBMS. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 1073−1076.

[52] Silberstein A, He H, Yi K, Yang J. BOXes: Efficient maintenance of order-based labeling for dynamic XML data. In: Stephanie Kawada, ed. Proc. of the 21st Int'l Conf. on Data Engineering (ICDE). Tokyo: IEEE Computer Society, 2005. 285−296.

[53] Kratky M, Pokorny J, Snasel V. Indexing XML data with UB-trees. In: Manolopoulos Y, Návrat P, eds. Proc. of the 6th East European Conf. on Databases and Information Systems (ADBIS). Bratislava: Springer-Verlag, 2002. 155−164.

[54] Rao P, Moon B. PRIX: Indexing and querying XML using prüfer sequence. In: Titsworth F, ed. Proc. of the 20th Int'l Conf. on Database Engineering (ICDE). Boston: IEEE Computer Society, 2004. 288−300.

[55] Wang HX, Park S, Fan W, Yu PS. ViST: A dynamic index method for querying XML data by tree structure. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Intl. Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 110−121.

[56] Bayer R. XML Databases: Modeling and multidimensional indexing. Invited talk at DEXA Conf. München, 2001. http://link.springer.de/link/service/series/0558/bibs/2113/21130001.htm

[57] Jiang HF, Lu HJ, Wang W, Yu JX. Path materialization revisited: An efficient storage model for XML data. In: Zhou XF, ed. Proc. of the 13th Australasian Database Conf. on Database Technologies 2002 (ADC). Melbourne: Australian Computer Society, 2002. 85−94.

[58] Wang HX, Meng XF. On the sequencing of tree structures for XML indexing. In: Stephanie Kawada, ed. Proc. of the 21st Int'l Conf. on Data Engineering (ICDE). Tokyo: IEEE Computer Society, 2005. 372−383.

[59] Yoshikawa M, Amagasa T. XRel: A path-based approach to storage and retrieval of XML documents using relational databases. ACM Trans. on Internet Technology, 2001,1(1):110−141.

[60] Chen Y, Davidson SB, Zheng YF. BLAS: An efficient XPath processing system. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 47−58.

[61] Goldman R, Widom J. Dataguides: Enabling query formulation and optimization in semistructured databases. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA, eds. Proc. of the 23rd Int'l Conf. on Very Large Data Bases (VLDB). Athens: Morgan Kaufmann, 1997. 436−445.

[62] Milo T, Suciu D. Index structures for path expressions. In: Beeri C, Buneman P, eds. Proc. of the 1999 Int'l Conf. on Database Theory (ICDT). LNCS 1540, Jerusalem: Springer-Verlag, 1999. 277−295.

[63] Kaushik R, Sheony P, Bohannon P, Gudes E. Exploiting local similarity for efficient indexing of paths in graph structured data. In: Agrawal R, Dittrich K, Ngu AHH, eds. Proc. of the 18th Int'l Conf. on Data Engineering (ICDE). San Jose: IEEE Computer Society, 2002. 129−140.

[64] Chung C, Min J, Shim K. APEX: An adaptive path index for XML data. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 121−132.

[65] Kaushik R, Bohannon P, Naughton JF, Korth HF. Covering indexes for branching path queries. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 133−144.

[66] Polyzotis N, Garofalakis M. Statistical synopses for graph-structured XML databases. In: Franklin MJ, Moon B, Ailamaki A, eds. Proc. of the 2002 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Madison: ACM Press, 2002. 358−369.

[67] Chen Q, Lim A, Ong KW. D(k)-index: An adaptive structural summary for graph-structured data. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 134−144.

[68] He H, Yang J. Multiresolution indexing of XML for frequent queries. In: Titsworth F, ed. Proc. of the 20th Int'l Conf. on Data Engineering (ICDE). Boston: IEEE Computer Society, 2004. 683−694.

[69] Wang W, Wang HZ, Lu HJ, Jiang HF, Lin XM, Li JZ. Efficient processing of XML path queries using the disk-based F&B index. In: Böhm K, Jensen CS, Haas LM, Kersten ML, Larson P, Ooi BC. eds. Proc. of the 31st Int'l Conf. on Very Large Data Bases (VLDB). Trondheim: ACM Press, 2005. 145−156.

[70] Barg M, Wong RK. Structural proximity searching for large collections semi-structured data. In: Paques H, Liu L, Grossman D, eds. Proc. of the ACM Conf. on Information and Knowledge Management (CIKM 2001). Atlanta: ACM Press, 2001. 175−182.

[71] Cohen S, Mamou J, Kanza Y, Sagiv Y. Xsearch: A semantic search engine for xml. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 45−56.

[72] Curtmola E, Amer-Yahia S, Brown P, Fernàndez M. GalaTex: A conformant implementation of the XQuery FullText language. In: Florescu D, Pirahesh H, eds. Proc. of the 2nd Int'l Workshop on XQuery Implementation, Experience, and Perspectives (XIME-P). Baltimore: ACM Press, 2005. 1024−1025.

[73] Amer-Yahia S, Botev C, Shanmugasundaram J. TeXQuery: A FullText search extension to XQuery. In: Feldman SI, Uretsky M, Najork M, Wills CE, eds. Proc. of the 13th Conf. on World Wide Web (WWW). Manhattan: ACM Press, 2004. 583−594.

[74] Amer-Yahia S, Lakshmanan LV, Pandit S. FleXPath: Flexible structure and full-text querying for XML. In: Weikum G, König AC, Deßloch S, eds. Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Paris: ACM Press, 2004. 83−94.

[75] Fuhr N, Großjohann K. XIRQL: A query language for information retrieval in XML documents. In: Croft WB, Harper DJ, Kraft DH, Zobel J, eds. Proc. of the 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR). New Orleans: ACM Press, 2001. 172−180.

[76] Balmin A, Papakonstantinou Y, Hristidis V. A system for keyword proximity search on XML databases. In: Freytag JC, Lockemann PC, Abiteboul S, Carey MJ, Selinger PG, Heuer A, eds. Proc. of the 29th Int'l Conf. on Very Large Data Bases (VLDB). Berlin: Morgan Kaufmann Publishers, 2003. 1069−1072.

[77] Weigel F, Meuss H, Schulz KU, Bry F. Content and structure in indexing and ranking XML. In: Amer-Yahia S, Gravano L, eds. Proc. of the 7th Int'l Workshop on the Web and Databases (WebDB). Maison de la Chimie: ACM Press, 2004. 67−72.

[78] Xu Y, Papakonstantinou Y. Efficient keyword search for smallest LCAs in XML databases. In: Proc. of the ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). Baltimore: ACM Press, 2005.

[79] Guo L, Shao F, Botev C, Shanmugasundaram J. XRANK: Ranked keyword search over XML documents. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data (SIGMOD). San Diego: ACM Press, 2003. 16−27.

[80] Kane B, Su H, Rundensteiner EA. Consistently updating XML documents using incremental constraint check queries. In: Chiang RHL, Lim EP, eds. Proc. of the 4th ACM CIKM Int'l Workshop on Web Information and Data Management (WIDM). McLean: ACM Press, 2002. 1−8.

[81] Bouchou B, Halfeld M, Alves F. Updates and incremental validation of XML documents. In: Lausen G, Suciu D, eds. Proc. of the 9th Int'l Conf. on Data Base Programming Languages (DBLP). LNCS 2921, Potsdam: Springer-Verlag, 2003. 216−232.

[82] Thompson H. xsv: Schema validator. 2002. http://www.w3c.org/2001/03/webdata/xsv

附中文参考文献:

[1] 孟小峰,周龙骥,王珊.数据库技术发展趋势.软件学报,2004,15(12):1822−1836. http://www.jos.org.cn/1000-9825/15/1822.htm

[49] 王静,孟小峰,王珊.基于区域划分的 XML 结构连接.软件学报,2004,15(5):720−729. http://www.jos.org.cn/1000-9825/15/720.htm