一种支持动态XML文档上关键字查询的索引结构

缪丰羽,林宏康
(宁德师范学院 计算机与信息工程系,福建 宁德 352100)

摘 要:在关键字查询领域,目前提出的大多数索引结构主要考虑的是静态的XML文档。当XML文档出现频繁更新时,这些索引结构可能面临着大范围的重新编码,从而增加了数据库索引维护的代价。为了能在XML文档动态更新的环境下保持其索引结构的稳定,提出了一种支持动态XML文档上关键字查询的索引结构DLSS(DDE Level Structure Summary)。该索引结构采用了一种针对动态更新改进的Dewey编码,该编码只需在文档更新时对新的节点赋予相应的编码,而不需要调整原有的编码结构。实验证明,DLSS索引结构可以在XML文档频繁更新或者较少更新时都能保持索引结构的相对稳定,并能在其上实现较高的关键字查询效率。

关键词:动态XML文档;关键字查询;索引结构;倒排索引
中图分类号:TP311.131 **文献标识码**:A **文章编号**:1673-629X(2012)06-0100-04

An Index Scheme for Keyword Search over Dynamic XML Document

MIAO Feng-yu,LIN Hong-kang
(Department of Computer and Information Engineering,Ningde Normal University,Ningde 352100,China)

Abstract:The index of XML document is important for its retrieval efficiency. The most of existing index schemes for keyword search consider static XML documents. But these indexes will face with widely re-label to increase the cost for database index maintenance when the XML documents are update frequently. In order to keep the index steady in dynamic environment,propose an index scheme DLSS (DDE Level Structure Summary) which applies to keyword search in dynamic XML document. The DLSS is based on an improved Dewey label which gives the new node with corresponding label when the document updates but not to justify the whole index scheme. The experimental results show that the DLSS index scheme not only keeps the index scheme steady in dynamic environment,but also realizes efficient for keyword search.

Key words:dynamic XML document;keyword search;index structure;inverted index

0 引 言

XML(eXtensible Markup Language)[1]如今已成为了Internet上信息表示和信息交换的实际标准。随着网络技术的飞速发展和XML数据的广泛使用,XML文档的数量也在急剧增多。为了有效管理大规模的XML数据,人们提出了各种针对XML数据特征的数据库技术[2,3]。

目前在XML关键字查询领域,研究的方法主要是基于最小公共祖先LCA(Lowest Common Ancestor)[4]的计算。这种查询方法通过记录每一个关键词在XML文档中所出现的位置,将其编码信息保存在关键字倒排索引(Inverted Index)中,从而快速计算出最小公共祖先LCA[5,6]。

然而,随着XML文档规模的不断增大,关键字倒排索引的体积也迅速增大。为了尽量保持LCA原有的计算效率,文献[7]在路径结构索引的基础上,提出了一种结构索引LSS(Level Structure Summary),该索引基于分层的思想,将具有相同标签路径的节点进行合并,具有高效判断节点之间关系的能力。在关键字倒排表较大时,相对于CVLCA(Compact Valuable Lowest Common Ancestor)[8]、SLCA(Smallest Lowest Common Ancestor)[9],LSS可以体现出更加高效的查询性能。

但这种索引结构只考虑到了静态的XML数据,当XML数据动态更新时,则该索引可能面临着大范围的重新编码,也意味着需要重新建立索引,这将对索引结构的稳定性造成较大的影响。

收稿日期:2011-10-20;**修回日期**:2012-01-30
基金项目:福建省自然科学基金(2011J01357)
作者简介:缪丰羽(1983-),女,讲师,硕士,研究领域为XML数据库技术;林宏康,副教授,研究领域为高校计算机教学及研究、数据库技术等。

1 相关研究

由于Dewey码可以体现路径信息,因此也成了关键字查询的首选编码方式。在LSS中,采用Dewey编码和节点名(Deweyid,label)来表示一个节点,然而Dewey编码在面临节点更新时,尤其是插入节点时,需要对所有的后续节点进行重新编码,因此LSS不适用于动态环境。

对于Dewey编码的改进,目前主要有以下研究。在最新版的SQL Server里使用的是ORDPATH[10]编码方式。ORDPATH为更新预留了偶数编码,在更新频繁的时候具有较高的性能,然而更新次数较少的时候,却造成了编码的浪费,编码的插入技术造成了编码的复杂,Dewey编码有些优势也会不复存在。文献[11]中提出的编码方式可以将原来的编码方式转换成另一种格式以适应动态更新而不用重新编码,比起之前的编码方式,在频繁更新时,这种方法具有更好的性能。然而,在转换编码格式时也需要一定的编码代价,而且,在XML文档集中可能有些需要频繁更新而有些不需要,在这种情况下,如果只对那些更新频繁的文档进行编码转换,则文档集可能面临着两种不同的格式,不同的存储方式及不同的查询机制。而且系统管理员需要确定哪些文档是动态哪些是静态,这也是一个很困难的事情,因为一个文档有可能在一段时间的频繁更新后保持一段时间的静止不变。为了避免这种情况,只能将所有文档都统一应用一种编码方式。然而,这样却又导致了额外的编码代价,更重要的是,静态文档需要适应动态文档的编码格式,而它们更适合静态编码方式。文献[12]通过定义一种巧妙的顺序概念,将Dewey编码转换成完全的动态编码方式:动态Dewey(DDE)。与之前的编码方式相比,DDE最大的特点就是既可以适应静态文档也可以适应动态文档。当没有更新时DDE编码同Dewey编码是一样的,编码体积小,查询性能高。

2 动态Dewey编码DDE

DDE编码是Dewey编码的改进,可以适应XML文档的动态更新。在插入新节点时,不需要对原来的XML文档进行重新编码。Dewey编码是一个整数的序列,每个整数之间用"."隔开。为了下面叙述方便,将每个整数部分称为一个字段。每个节点的Dewey编码最后一个字段代表该节点在其兄弟节点中的位置,前面的字段代表其祖先节点的信息,即从根节点到该节点的路径信息。DDE编码将Dewey编码从正整数范围拓展到了整数范围。其初始值同Dewey编码相同。当文档中插入新的节点时,对新节点的编码设置可以分成以下几种情况:

(1) 若新节点在节点 $a_1.a_2.\cdots a_m$ 的左边插入,则新节点的编码为 $a_1.a_2.\cdots(a_m-1)$, a_m-1 有可能为0,也有可能为负整数;

(2) 若新节点在节点 $a_1.a_2.\cdots a_m$ 的右边插入,则新节点的编码为 $a_1.a_2.\cdots(a_m+1)$;

(3) 若新节点插入作为节点 $a_1.a_2.\cdots a_m$ 的子节点,则新节点的编码为 $a_1.a_2.\cdots a_m.1$;

(4) 若新节点在两个相邻节点 $a_1.a_2.\cdots a_m$ 和 $b_1.b_2.\cdots b_m$ 之间插入,则将两节点各字段的值相加作为新节点的DDE编码,即新节点的编码为 $(a_1+b_1).(a_2+b_2).\cdots(a_m+b_m)$。

前三种情况同Dewey编码基本相同,第4种情况在文献[12]中已经证明这种编码方式可以保持文档中各节点的相应顺序,并且不需要对任何节点进行重新编码。

例:图1展示了DDE编码对于各种方式插入的节点的编码方法。其中虚线部分为插入的新节点。节点A插入节点1.1之前,因此A的编码为1.0;节点B插入节点A之前,所以B的编码为1.-1。节点C插入节点1.4.1之后,因此C的编码为1.4.2;同理D的编码为1.4.3。节点E插入节点1.2.1和节点1.2.2之间,因此E的编码为(1+1).(2+2).(1+2),即2.4.3,同理节点F的编码为3.6.5,节点G为5.10.8。节点H插入作为节点1.2.1的子节点,因此节点H编码为1.2.1.1,节点I编码为3.6.5.1。

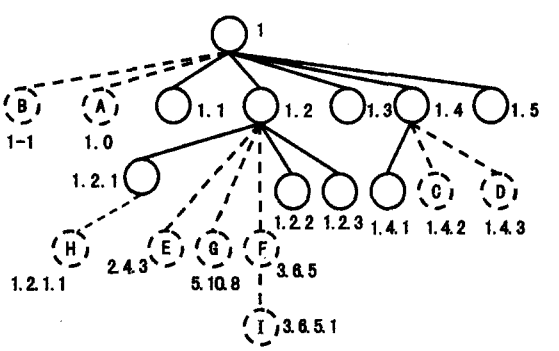

图1　DDE编码的插入操作

3 索引结构DLSS

接下来介绍基于DDE编码的索引结构DLSS。

XML文档可以用一棵有序的标签树 T 来表示,其中 $T=(r,V,E,L,\lambda)$,r 是XML文档树的根节点,V 是文档中的节点集合,E 是文档中各节点之间的关系集合,λ 为节点名,L 是所有节点名的集合。采用二元组(DDEid,Label)标识一个节点,其中DDEid代表节点的DDE编码,Label代表节点名。将一个元素的属性节点视为该元素的子元素节点。

从计算LCA的方法中可以得知,如果能有效减少

各个关键字倒排表的大小，就能大大减少 LCA 的计算时间。通过对 XML 文档结构的仔细分析，发现通常相同标签的节点在文档树中的深度都相同，也即这些相同标签的节点大都出现在同一层。因此，可以考虑采用结构索引的技术，对满足某一条件的同层节点进行合并，来减小 XML 文档树的体积。

定义 1(节点类型)[3]：XML 文档中一个节点的节点类型为从根节点到该节点的标签路径。

定义 2(等价节点)[3]：如果在一棵文档树中两个节点的节点类型相同，则为等价节点。

DLSS 索引结构将 XML 文档树中的等价节点进行合并，每一个索引项代表一个关键字对应的所有等价节点的集合，用组节点 GN<label，DSet>表示，其中 label 为等价节点的标签名，DSet 表示该标签对应的所有等价节点的集合，集合内的节点用 DDE 编码表示。DLSS 将 XML 文档树转化成一棵有序的标签树 LT = {GV，GE}，其中 GV 代表 DLSS 中所有 GN 的集合，GE 代表 GV 中各个 GN 之间的关系。

下面的图 2、图 3 分别代表 XML 文档树和 DLSS。

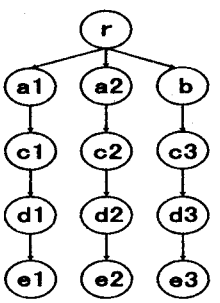

图 2 XML 文档树

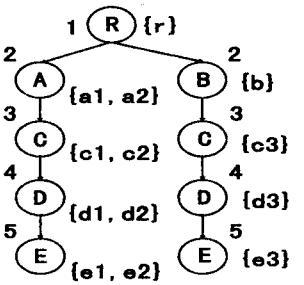

图 3 DLSS

●构建 DLSS 索引的 DSCAN 算法步骤如下：

(1)首先，选择 B+树来记录当前文档中所有可能的标签路径。

(2)当处理 XML 文档中的元素时，检测当前元素的节点类型，若 B+树中已有此节点类型的 DDE 编码信息，则将元素的节点类型编码设成 DDE，否则将该元素的 DDE 编码信息添加到 B+树中。

(3)当处理到属性值或者元素的文本值时，首先将其进行有效的分解，然后对处理后产生的每一个关键字，按照元素的方式处理。

(4)当处理元素结束后，在索引中查找与当前关键字相同的信息，若找到了一个节点，则更新此节点的 GN 信息，否则需要创建一个新的节点插入到索引中，此节点的标签为当前元素节点的标签名，同时新建一个 GN，将当前元素节点放入 GN 中。

由于论文篇幅限制，算法的具体实现在此省略。

●基于 DLSS 的查询处理步骤如下：

(1)根据用户输入的关键字集合，从 DLSS 索引中取出其对应的各个 GN 集合，以及每一个关键字对应的 DSet 集合。

(2)在 GN 上采用 XRANK 中的 DIL 算法计算出所有的 GLCAS。

(3)根据生成 GLCAS 的关键字的 DDE 信息，取出关键字对应的 DSet 集合。

(4)在每个关键字的 DSet 集合上采用快速查找算法得出所有的 LCA，并根据 CVLCA 的判断方法得到最终的 CVLCA 集合。

由于论文篇幅限制，算法的具体实现在此省略。

4 实验分析

实验的硬件平台：AMDAthlon(tm) Ⅱ X4 640 Processor，主频为 3GHz，3.25GB 内存。软件平台：Microsoft Windows XP SP3，SAX2 API，SQL Server 2000 以及 JAVA 编程语言。

测试数据为 Sigmod Record、NASA、TREEBANK 以及 DBLP。这四个数据集的大小分别为 704kB、25MB、84MB 和 134MB。

索引大小：图 4 描述了 DLSS 和原始数据在 NASA、TREEBANK、DBLP 这三个数据集上的空间占用情况。其中 TREEBANK 数据集上的 DLSS 同原始数据相差最小。这是因为 TREEBANK 分布在每一层上的数据比较平均。NASA 次之，DBLP 差距最大。

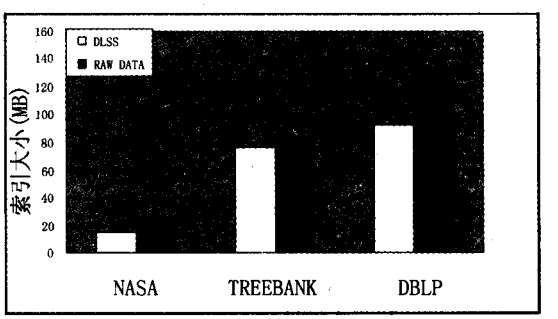

图 4 索引大小

动态更新：从 DDE 编码的特点可知，如果增加新的节点，不需要对原有的节点进行重新编码，只需在索引中添加新的节点即可。如果删除一个节点，也不会对原有的节点编码产生影响，因此也只需要删除索引中相应的节点信息即可。更新操作的时间复杂度都为 O(1)。不论是频繁更新还是较少更新，对于 DLSS 索引方式，都不会产生很大的影响。而对于 LSS 索引方式，XML 文档的动态更新将会导致重新建立索引。实验选择 Sigmod Record 数据集进行 4 种不同的更新操作：

1)在标签为 title 的节点前插入一个节点 a；

2)在标签为 author 的节点后插入一个节点 b；

3)插入一个节点 c 作为节点 paper 的子节点；

4)在相邻的两个 paper 节点之间插入节点 d。

实验结果如图 5 所示。图中 LSS 的更新时间为其重索引时间。

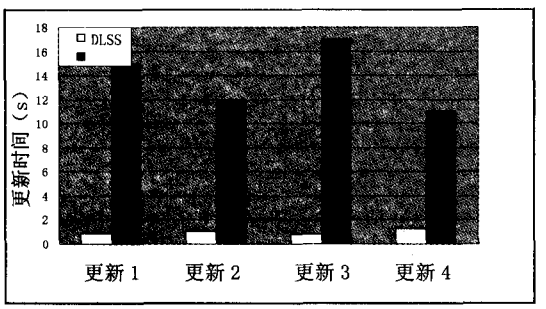

图 5　更新时间

查询效率:选择 DBLP 数据集,在其上选取 5 个不同的关键字集合,其对应的关键字个数为 2 ~ 6 个。图 6 描述了 DLSS 索引方式和 LSS 索引方式在这 5 个关键字查询上的查询效率。可以看出,两种索引方式的查询效率基本相同。

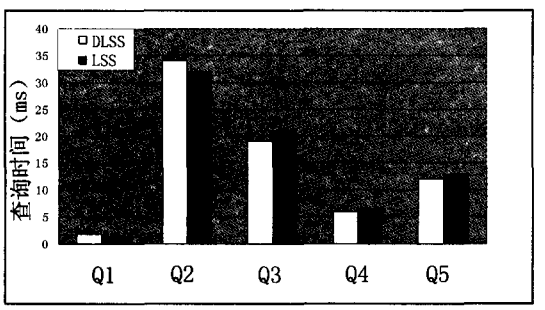

图 6　查询时间

综上所述,DLSS 索引方式不仅可以在动态环境下保持了索引结构的相对稳定,也保持了高效的查询性能。

5　结束语

文中分析了当前关键字查询在动态环境下的不足,以及目前的几种支持更新的编码方式,在此基础上提出了一种支持动态XML文档上关键字查询的索引 DLSS。实验证明,该索引方式不仅可以在动态环境下保证索引结构的稳定,而且保持了高效查询性能。

参考文献:

[1] eXtensible Markup Language(XML)[EB/OL]. 2004. http://www.w3.org/xml.

[2] 周爱武,李孙长,程　博,等. XML 数据库的研究与应用[J]. 计算机技术与发展,2009,19(9):218-224.

[3] 华珊珊,谢铉洋. XML 查询语言 XQuery 的研究与实现[J]. 计算机技术与发展,2009,19(4):48-50.

[4] Guo L,Shao F,Botev C,et al. Xrank:ranked keyword search over xml documents[C]//SIGMOD. [s. l.]:[s. n.],2003:16-27.

[5] 郑榕增,林世平. 基于 Lucene 的中文倒排索引技术的研究[J]. 计算机技术与发展,2010,20(3):80-83.

[6] 韩　萌,陈　群,王　鹏. 基于 LCA 的高效 XML 关键字检索算法[J]. 计算机工程,2010,36(23):59-62.

[7] 娄　颖,李战怀,郭文琪,等. 一种基于 XML 文档关键字检索的结构索引[J]. 计算机科学,2010,37(12):120-124.

[8] Li Guoliang,Feng Jianhua,Wang Jianyong,et al. Effective keyword search for valuable lcas over xml documents[C]//Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management. New York:ACM,2007:31-40.

[9] 高丹丹. 基于 SLCA 的 XML 关键字查询研究与改进[D]. 济南:山东大学,2009:23-24.

[10] O'Neil P,O'Neil E,Pal S,et al. ORDPATHs:Insert-friendly XML Node Labels[C]//Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris,France:ACM,2004:903-908.

[11] Li C,Ling T W,Hu M. Efficient Updates in Dynamic XML Data:from Binary String to Quaternary String[J]. the VLDB Journal,2008,17(3):573-601.

[12] Xu Liang,Ling T W,Wu Huayu,et al. DDE:From Dewey to a Fully Dynamic XML Labeling Scheme[C]//Proceedings of the 35th SIGMOD International Conference on Management of Data. New York,USA:ACM,2009.

(上接第 99 页)

[9] 史荣昌,魏　丰. 矩阵分析[M]. 北京:北京理工大学出版社,2005.

[10] Noh M,Lee Y,Park H. Low complexity LMMSE channel estimation for OFDM[J]. Communications IEEE Proceedings,2006,153(5):645-650.

[11] 梁　琳,李小文. LTE 上行信道估计的算法与性能分析[J]. 广东通信技术,2010,30(3):29-31.

[12] Lee Dae-Hong,Im Se-Bin,Choi Hyung-Jin. A novel pilot mapping method for channel-quality estimation in SC-FDMA system[C]//Asia-pacific Conference on Communications,APCC 2007. [s. l.]:[s. n.],2007:307-310.

[13] Dong Min,Tong Lang. Optimal design and placement of pilot symbols for channel estimation[J]. IEEE Transactions on Signal Processing,2002,50(12):3055-3069.