

寻找 XLCA 的 XML 数据流 TOP-K 关键字查询算法

冯 静,余建桥,李雪娇

(西南大学 计算机与信息科学学院,重庆 400715)

摘 要:XML 关键字查询是一个用户比较方便的信息搜索方法,非常适用于用户在不熟悉 XML 查询语言和底层结构的情况下进行信息查询。现有的 XML 数据流上关键字查询多采用查找 SLCA 结果集的方式,为了解决基于 SLCA 结果集定义的不完备性,引入了基于 XLCA 的结果集定义,使其查询包含尽可能全的结果。文中对于 XML 数据流提出利用滑动窗口模型保存数据,基于 XLCA 的结果集定义,提出了一种 TOP-K 关键字查询算法,并从理论上证明了此算法的正确性和查询的完备性,分析了其时间复杂性和空间复杂性。

关键词:SLCA;XLCA;XML 数据流;滑动窗口

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2011)09-0095-04

TOP-K Keyword Query for Exclusive LCAs on XML Data Streams

FENG Jing, YU Jian-qiao, LI Xue-jiao

(College of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract: XML keyword query is a user-convenient information search technique, which is well suited for users who are not familiar with XML query language and the underlying structure. Existing keyword queries on XML data streams often were based on the result set of SLCA, and in order to solve the incompleteness of the SLCA result set, then a result set based on the definition of XLCA was introduced, as far as possible to the query contains all of the results. In this paper sliding window model was proposed to save the XML data stream, then based on the definition of the result set of XLCA, a TOP-K keyword query algorithm was proposed. The algorithm was proved the correctness and completeness theoretically, and analyzed its time and space complexity.

Key words: SLCA; XLCA; XML data streams; sliding window

0 引 言

可扩展标记语言(XML^[1])成为 Internet 环境上数据表示和交换的标准。当前,国内外学者对 XML 流数据上的查询技术进行了许多研究。这些研究主要集中在两个方面:XML 数据流上的基于 XPath&XQuery 语言查询^[2~4]和对于 XML 数据流上的关键字查询^[5]。数据库和信息检索技术的融合^[6,7]的研究目标也使基于 XML 流数据的关键字查询成为学术界研究的热点。在用户不了解复杂的 XML 查询语言,不知道 XML 数据流的文档结构模式和数据库底层结构的情况下,基于关键字的查询同样可以找到用户感兴趣的

文档片段。这些研究工作对有效地表达和处理 XML 数据流上的关键字查询存在着不足之处。

王小峰等^[8]首先提出在 XML 数据流上使用关键字查询的方式,提出的 Lookup 算法仅针对单个查询而言。黎玲利等^[9]在 Lookup 算法基础上提出了 XML 数据流上的 Top-k 关键字查询可以有效支持多查询处理,对 XML 数据流的流速及流向问题没有涉及,并且其查询结果集是基于可能丢失部分有意义的片段的 SLCA^[10~12](最小最近公共祖先)上的。

滑动窗口可以看成是一个数据区间,保存数据流中最近到来的数据,利用滑动窗口保存 XML 流数据,可以将无限的数据流转化为有限的关系,基于 Dewey 编码的滑动窗口模型又可以很好的处理 XML 节点间的祖孙兄弟关系。本研究就是在这种思想下,提出了 XML 数据流关键字的查询 TOP-K^[13,14]算法。

收稿日期:2011-02-16;修回日期:2011-05-06

基金项目:国家“863”计划资助项目(2006AA10Z1E6);西南大学资助项目(Z20100001)

作者简介:冯 静(1986-),女,湖北汉川人,硕士研究生,研究方向为 XML 数据库、流数据库;余建桥,教授,博士,研究方向为数据库技术、人工智能。

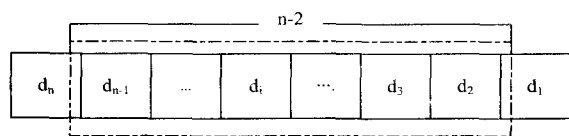
1 基于滑动窗口的 XML 数据流模型

XML 流处理是 XML 文档的节点一次性地按照某

种遍历的顺序流过,因为每次面对的总是单个的节点(元素、属性或 text),所以需要必要的将数据有效地缓存,故引入滑动窗口机制。

1.1 滑动窗口

数据流按照时间的先后顺序依次进入滑动窗口,滑动窗口可表示如图 1 所示。



$d_i (1 \leq i \leq n)$ 表示第 i 个进入滑动窗口的数据

$(n-2)$ 表示窗口大小

图 1 滑动窗口

1.2 带有 Dewey 编码的 XML 文档模型

通常 XML 文档模型可以看成是一棵带有标签的有序树,树中的节点表示文档中的元素、属性和文本节点,每个节点对应唯一的一个标签,树中的边代表了节点之间的祖孙后代关系。Dewey 编码可以很好地表达节点间的祖孙后代关系和兄弟关系,其编码规则如下:(1)根节点的编号为 0;(2)假设某节点的标号为 x ,则该节点的第 1 个子节点的编号是 x_0 ,依次为 x_1, x_2, \dots 。利用 Dewey 编码可以判断节点是否在同一层,是否有公共的父节点。带有 Dewey 编码的 XML 树如图 2 所示。

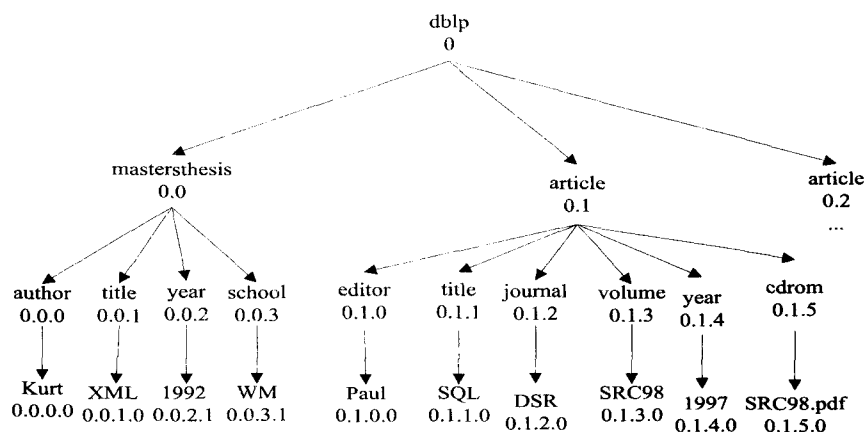


图 2 Dewey 编码的 XML 树

2 XTOP-K 算法的研究与实现

现有的 XML 关键字的常用方法是基于 SLCA,但是它有可能引起一部分有意义的结果的丢失。例如如图 3 所示,虚线框部分就会丢失,本研究就是基于滑动窗口模型,在 XLCA^[15]的基础上,提出 XLCA_TOP-K 关键字查询算法(以下简称 XTOP-K 算法),从而得到有效的 XML 文档片段。

2.1 基于 Dewey 编码的滑动窗口模型

滑动窗口可以看成是数据流上的一个区间,其保

存着数据流中最近到来的一部分数据。这样,对数据流的实时处理就可以转化为对滑动窗口中数据的实时处理。XML 数据流的滑动窗口模型是基于 Dewey 编码的。

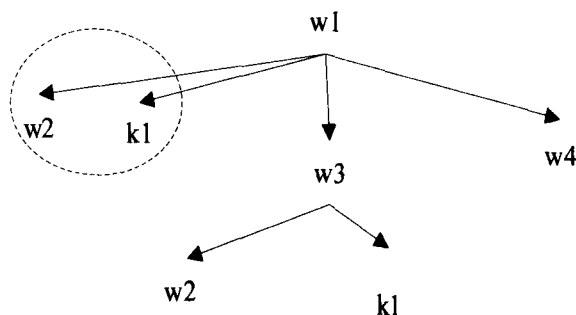


图 3 SLCA 丢失结果图示

用 $T(V, E)$ 来表示一个 XML 数据流数据,其中 V 表示所有节点的集合,并且每个节点用自己的 Dewey 编码作为标识符, E 表示边的集合。

针对当前 XML 数据流上关键字查询的问题,给出了以下相关概念的定义。

定义 1(基于 Dewey 编码的滑动窗口模型)基于 Dewey 编码的滑动窗口模型可以表示如下形式:

$$W = \{ \langle s_1, v_1 \rangle, \langle s_2, v_2 \rangle, \langle s_3, v_3 \rangle, \dots, \langle s_n, v_n \rangle \}$$

其中 s_n 是结点在 XML 树中的 Dewey 编码序列, v_n 是结点。由于主要的研究是 XML 数据流上的 TOP-K 关键字查询,所以没有深入考虑滑动窗口的大小,只是将其

定义为一个完整的 XML 文档上。

2.2 XTOP-K 算法的相关定义

定义 2(关键字匹配)

给定查询关键字集合 $wk = \{k_1, k_2, \dots, k_n\}$ 和 XML 数据流上的一个叶节点 v ,如果 v 的文本内容包含在关键字 k_i 里面,则说明节点 v 和关键字 k_i 匹配。

定义 3(互斥最低公

共祖先 XLCA: eXclusive Lowest Common Ancestor) 若节点 x 是 XML 数据流上的一个 XLCA 节点,当且仅当以节点 x 为根的片段包含了关键字集合 $wk = \{k_1, k_2, \dots, k_n\}$,且以节点 x 为根的片段中不存在某个子孙节点已经包含了关键字集合 $wk = \{k_1, k_2, \dots, k_n\}$ 。

定理 1 $SLCA(T) \subseteq XLCA(T)$, $XLCA(T) - SLCA(T)$ 是 $SLCA(T)$ 丢失的那部分有意义的结果。

证明:寻找 XLCA 的方法是首先找出 SLCA,然后将其删除,再在剩余的 XML 文档树中寻找 SLCA。故 SLCA 一定包含于 XLCA,而 XLCA 不一定是 SLCA。

定义 4 (相关连通子树 RCT: Related Connected Subtree) 以 XLCA 为根, 连结所有关键字 k_1, k_2, \dots, k_n 节点的最小子树。

定义 5 (XML 数据流上的 TOP-K 关键字查询) XML 数据流上的 TOP-K 关键字查询是一个二元组 $Q = \{wk, K\}$, 其中 $wk = \{k_1, k_2, \dots, k_n\}$ 是关键字的集合, K 是查询结果数量。这个查询返回的结果是当前 XML 数据流到达的数据片段中前 K 个规模最小的 w 中的相关子树。

2.3 XTOP-K 算法的数据结构

针对查全的问题, 将 XML 数据流以 XML 文档树的模式保存在一个足够大的大小可变的滑动窗口模型里面。

在 XTOP-K 算法中用栈 s 保存当前层的叶子节点。节点 x .parent (节点 x 的父节点) 状态由长度为 N 的二进制序列 L 表示。第 i 位为 1, 表示该节点的父节点存在某个后代叶节点的值包含第 i 个关键字; 反之, 则表示不存在。每个非叶节点的 L 的初始态每一位均为 0, 表示其没有任何后代包含关键字。用大小为 K 的数组保存查询到的 K 个 XLCA 节点。

2.4 XTOP-K 算法描述

处理 XML 数据流常用的事件驱动接口是 SAX (Simple APIs for XML)。它的基本原理是由接口的用户提供符合定义的处理函数, XML 解析时遇到特定的事件, 就去调用特定事件的处理函数。它提供了顺序访问 XML 文档的模式, 在访问过程中会触发一系列的事件。用 Begin(tag)、End(tag)、Text() 三个事件, 分别表示遇到一个元素 tag 的开始标签时触发的事件, 遇到元素 tag 的结束标签时触发的事件, 遇到元素的值时触发的事件。

算法的基本思想是: 从 XML 文档树的最低层开始, 自底向上逐层向上传递关键字的出现信息, 如全部关键字同时到达某一节点, 就得到一个 XLCA 节点, 并将以这个 XLCA 为根的子树删除, 接着再递归地进行查找, 直到找到 K 个相关连通子树或 XML 文档树根节点为止。

算法的具体步骤如下:

对于当前层 level 来说:

1) 给每个非叶子节点分配一个 n 长度的二进制序列 L , 若第 i 位为 1 表示此节点存在某个后代叶节点包含关键字 k_i ; 反之, 不存在。

2) 若某个叶节点包含任意关键字 k_i , 则将其父节点 (第 level-1 层) 的第 i 位置 1, 若存在有一父节点的 L 全为 1, 则输出与该父节点对应的子树并将该子树从 XML 文档树中删除。

3) 检查完 level 层后, 依此检查 level-1 层直到找

到 K 个最小相关连通子树或是遇到 XML 文档树的根节点。

算法 1 XLCA_TOP-K 关键字查询算法

输入: 关键字查询 $Q = \{w, K\}$

输出: Q 的查询结果

栈 s 为空, 查询到的结果个数 num=0

level=docdepth+1;

while (level>0&&num≤K) {

Begin(x)

if(x 是叶子节点) {

s.push(x);

L(x.parent)=00...0(N 位);

else Lx=00...0(N 位);

endfunction

Text(x.v)

if(! (k_i! =x.v)) {

s.pop();

delete(v); }

else 将 v.parent 二进制序列第 i 位置 1;

endfunction

End(v)

if(L(v.parent)=11...1(N 位)) {

s.pop();

将 v.parent 保存;

delete(以 v.parent 为根节点 RCT);

num++; }

endfunction

}

output(Q); //最后输出满足条件的前 K 个规模最小的查询结果

3 性能分析

3.1 算法的正确性和完备性分析

从以下两点证明 XTOP-K 算法的正确性和查询的完备性。

1) XTOP-K 算法对于每一个查询结果都可以找到它。

2) 对于每一个查询结果, 确实是 XTOP-K 算法输出的结果。

证明: 1) 因为每个应有的查询结果都是 XML 数据流的一个树形片段 (以 XLCA 为根的 RCT), 若设该 RCT 片段的根节点是 r , 因为 XTOP-K 算法会把关键字存在的信息传递给其父节点, 最后全部汇集到节点 r 上, 接着返回这个查询结果, 所以只要满足条件的搜索结果存在, XTOP-K 算法就一定可以找到它。2) 因为 XTOP-K 算法输出结果的一个前提条件是该片段的 XLCA 节点的二进制序列为全 1 (即其后代节点包含所有的关键字), 所以它一定是准确的搜索结果。

3.2 时间复杂性和空间复杂性分析

3.2.1 时间复杂性分析

由于 XML 片段形态和查询都较大地影响算法的时间复杂性,因此本研究仅分析 XTOP-K 算法在最坏的情况下时间复杂性。

设树的深度是 h , 数据流中接收到的叶节点个数是 N_1 , 非叶节点个数是 N_2 。

显然有处理 $\text{Begin}(x)$ 事件的时间复杂性是 $O(l)$ 。当处理 $\text{Text}(x)$ 事件时,判断当前节点的值是否包含关键字结合中某个关键字所需要的时间复杂性假设是 f , 做删除算法的时间复杂性显然是 $O(l)$, 该关键字集合大小是 N , 故 Text 事件的时间复杂性是 $O(N * \max\{f, O(l)\})$ 。处理 $\text{End}(x)$ 事件的时间复杂性: 将 XLCA 节点保存到数组的时间复杂性是 $O(l)$, 删除以 XLCA 为根节点的相关连通子树 RCT 在最坏的情况下的时间复杂度是 $O(N * h)$ 。

此算法的时间复杂度是 $\max(N_1 * (N * \max\{f, O(l)\}), N_2 * O(N * h))$ 。

3.2.2 空间复杂性分析

将 XML 数据流保存在滑动窗口模型里面,所需的存储空间是 $O(N_1 + N_2)$ 。所需栈的存储空间是 $O(N_1)$ 。保存查询结果的数组 Q 所需要的存储空间是 $O(\lambda * N)$ (λ 表示查询返回的个数)。此算法的空间复杂度为 $\max(O(N_1 + N_2), O(\lambda * N))$ 。

以上从理论上对 XTOP-K 算法的正确性、完备性、时间复杂性及空间复杂性进行了分析,证明了算法的正确性和完备性,且有较好的时间复杂度。

4 结束语

在 XML 流数据库中, TOP-K 查询是一类重要查询。本研究在 Dewey 编码的基础上,提出用滑动窗口模型保存 XML 数据流信息;且针对 XML 数据流关键字查询中 SLCA 丢失一部分有意义查询结果,改进了 XLCA 松弛语义查询结果,并提出了基于 XLCA 的 XML 数据流上的 TOP-K 关键字查询算法。从理论上证明了此算法的正确性和查询的完整性,分析了该算法有较好的时间复杂性。本研究是在尽量不丢失 XML 数据流信息的前提下进行尽可能全的查询,故对如何利用最少的存储空间去存储 XML 数据流数据没有太多的考虑。进一步的工作包括 XML 数据流查询

中滑动窗口的大小等问题的讨论。

参考文献:

- [1] 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):218-221.
- [2] Min J K, Park M J, Chung C W, et al. XTREAM: An efficient multi-query evaluation on streaming XML data[J]. Information Sciences, 2007, 177: 3159-3538.
- [3] Bose S, Fegaras L, Leine D, et al. A Query Algebra for Fragmented XML Stream Data[C]//In Proceedings of the 9th International Conference on Data Base Programming Languages (DBPL). Postdam, Germany: [s. n.], 2003: 195-215.
- [4] Weim M Z, Rundensteiner E A, Mani M, et al. Processing recursive XQuery over XML streams: The Raindrop approach[J]. Data & Knowledge Engineering, 2008, 65: 243-265.
- [5] 李波,杨卫东. XML 数据流上的关键字查询算法[J]. 计算机工程, 2009, 35(4): 35-37.
- [6] 王春华. 2008 年中国计算机科学技术发展报告[R]. 北京: 机械工业出版社, 2009: 61-64.
- [7] 孟小峰,周龙骧,王珊. 数据库技术发展趋势[J]. 软件学报, 2004, 15(12): 1822-1836.
- [8] 王小峰,孟小峰,周军峰,等. XML 数据流上的关键字查询[J]. 计算机研究与发展, 2006, 43(S): 484-489.
- [9] 黎玲利,李建中,骆吉洲,等. XML 数据流上的 TOP-K 关键字查询处理[J]. 计算机科学, 2008, 35(10): 159-164.
- [10] Xu Y, Papakonstantinou Y. Efficient Keyword Search for Smallest LCAs in XML Databases[C]//In Proceedings of SIGMOD. [s. l.]: [s. n.], 2005: 527-538.
- [11] Sun C H, Chan C Y, Goenika A K. Multi-way SLCA-based Keyword Search in XML Data[C]//In Proceedings of World Wide Web. [s. l.]: [s. n.], 2007: 1043-1052.
- [12] 谢涛,王晓玲,周傲英,等. XML 关键字检索的最低公共祖先快速查找方法[J]. 计算机研究与发展, 2006, 43(S): 477-483.
- [13] Luo Y, Lin X M, Wei W, et al. SPARK: Top-k Keyword Query in Relational Databases[C]//In Proceedings of SIGMOD. [s. l.]: [s. n.], 2007: 115-126.
- [14] Jin C Q, Yi K, Chen L, et al. Sliding-Window Top-k Queries on Uncertain Streams[J]. VLDB Journal, 2010, 19(3): 411-435.
- [15] 朱皓,杨卫东,施伯乐,等. XML 关键字搜索中一个高效的寻找 XLCA 的算法[J]. 计算机研究与发展, 2008, 45(S): 383-389.

(上接第 94 页)

- [9] 盛跃宾,陈定昌. 有等式约束优化问题的粒子群优化算法[J]. 计算机工程与设计, 2006, 27(13): 2412-2418.
- [10] 贺俐,陈桂兴. 计算方法[M]. 武汉: 武汉大学出版社, 1998.

- [11] 李庆阳,王能超. 数值分析[M]. 北京: 清华大学出版社, 施普林格出版社, 2004.
- [12] 李太勇,唐常杰. 基因表达式编程种群多样性自适应调控算法[J]. 电子科技大学学报, 2010, 39(2): 279-283.