第32卷 第15期 计 算 机 工 程 2006年8月
Vol.32 № 15 Computer Engineering August 2006
·软件技术与数据库· 文章编号：1000—3428(2006)15—0090—03 文献标识码：A 中图分类号：TP312

一种新的基于 XML 的索引机制

姚全珠，丁晓剑，任雪利，张志锋
(西安理工大学计算机科学与工程学院，西安 710048)

摘 要：当前基于 Web 的半结构化数据越来越受到重视。该文分析了当前对 XML 数据检索的相关工作，提出了一种路径索引技术，并将之无缝结合了基于文本的倒排索引文档，以实现 XML 文档的内容和结构的双重检索。该方法只需要对文档库扫描一次，可以大幅度降低用户查询时间。

关键词：XML；信息检索；索引；倒排文档

A New Index Mechanism Based on XML

YAO Quanzhu, DING Xiaojian, REN Xueli, ZHANG Zhifeng
(College of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048)

【Abstract】 Semistructured data based on Web is taken more and more attention now. This paper analyzes data retrieval of XML currently, and proposes a kind of path index technology which have been combined with inverted file to implement retrieval both on context and structure. This method needs scan only once to file base, which can reduce user query time with a large number.

【Key words】 XML; Information retrieval; Index; Inverted file

目前的一些基于XML文档的查询语言，如XMLQL、Xpath和Xquery，只支持传统的关键字查询，从数据库的角度来存储和检索XML文档，借助于标准数据库的查询语言,检索结果能够准确地满足查询条件[1]。这种检索模式实际上是对数据库的检索，忽略了XML文档的文本性，体现不出XML文档检索的意义。

1 相关定义

针对本文出现的一些术语，给出如下定义：

定义 1 正则路径表达式(regular path expression)可以访问 XML 文档结构的任意层次，对常用的祖先/后裔关系的结构连接，如查询 //rock[contains(descendant-or-self ::,"dangerous")]则是返回所有自身及后裔元素的内容中至少拥有一个关键字"dangerous"的 rock 元素。

定义 2 连续路径就是指从开始节点到终止节点组成的有序集合，集合内节点的次序必须是不间断的。子连续路径也是一个连续路径，它是某一连续路径的片断。

定义 3 事务(transaction)是指由根节点到叶子节点之间的路径。

定义 4 若连续路径在文档集中出现的次数大于系统的阈值，称此路径为高频路径，反之，则称为低频路径。

2 相关工作

当前在数据库、XML数据管理、信息检索和多媒体数据管理等领域[2]都有了相当成熟的技术，表 1 给出相关的技术。

表 1 各种数据的相关索引技术

数据	用途	索引技术
结构化数据	数据库	B 树、位图索引等
半结构化数据	XML 和其它半结构化数据	值索引、字符串索引、路径索引、节点索引、链接索引
文本数据	信息检索	倒排文档
多维数据	多媒体应用和其它一些方面	R-tree, TV-tree,VA-file 等

Daniela等人提出了一种倒排索引结构[3]，利用扩展的XMLQL语言支持关键字搜索，其中XML文档的每个元素的层次数目都被记录在倒排文档的索引词条中。它的缺点是当从倒排文档获得精确的节点信息时，需要对整个文档树重新扫描。

当前存在的这些索引技术对有些特殊的应用可以说是高效的。但是其中大部分需要配合专门的查询语言才能使用，而不能用简单的关键字查询来检索结构文档。

3 新的索引解决方案

文献[4]提到可以把 XML 文档看成是带有附加标记的文本文档，把 XML 的结构信息集成到倒排文档中，实现对 XML 文档实行有效的索引。它虽然可以应付各种查询，但是存储这些所有路径信息将会需要大量的磁盘空间。再加上倒排表也是一种以空间换时间的策略，这种索引的构造和维护的成本都很高。

本文针对上面提到的问题，对路径索引进行了优化，利用经常在 XML 文档中出现的高频路径作为索引机制构造的依据，并利用 hashing 方法来进行路径搜索。该法只需扫描文档树一次，极大地节省了构造索引的时间，并有效地控制了索引所占用的空间。

3.1 路径查找技术

在XML文档中，所有的信息都存储在叶子节点处。信息的存取是由根节点沿着某路径(path)到叶子节点完成的。可以把XML文档看作成树状结构，节点间存在一种次序(order)关系，该文用搜索连续路径的方法，快速检索出文件中重要的连续路径。" 20%的高频路径可以检索出 80%的信息 " 这就是著名的帕累托定律，也称为 20/80 法则，" 多数，它们只能

作者简介：姚全珠(1960—)，男，博士、教授，主研方向：数据库，软件工程方法学，网络技术；丁晓剑、任雪利、张志锋，硕士
收稿日期：2005-10-23 **E-mail：**wjswsl@163.com

造成少许的影响；少数，它们造成主要的、重大的影响”。在此，主要工作就是实现高频路径的查找。下面将结合图 1 给出相应的高频路径生成技术：

```
<Database>
<Document>
<Course>
<Name> Architecture</Name>
<Name>Database</Name>
</Course>
<DBLab>
<Paper>
<XML>XML 1.0</XML>
<XML>X003</XML>
</Paper>
</DBLab>
</Document>
<Music>
<Rock>Dangerous</Rock>
<Lyric>Now and forever</Lyric>
<Pop> Automatic</Pop>
<Pop> Angela</Pop>
</Music>
</Database>
```

(a)文档集合中的文件 1

```
<Database>
<DBLab>
<Paper>
<XML>X001</XML>
<XML>X002</XML>
<XML>X003</XML>
 <Mobile>M001</Mobile>
<Mining>DM001</Mining>
</Paper>
</DBLab>
</Database>
```

(b) 文档集合中的文件 2

```
<FTP>
<Tool>Winrar</Tool>
<Movie>The others</Movie>
<Movie>E.T.</Movie>
<Music>
<Pop>Automatic</Pop>
<Pop> To be</Pop>
<Other>Belief</Other>
<Other>Fever</Other>
</Music>
</FTP>
```

(c) 文档集合中的文件 3

图 1 文档集合中的文件

图 1(a)~图 1(c)分别是 XML 文档库的 3 个文件，这 3 个 XML 文件共有 20 个事务。其中 Database. 草药 Document.Course.Name 是一个长度为 4 的连续路径，而 Database.Document.Course 和 Document.Course.Name 都是此路径的子连续路径。当系统预设的阈值为 4 时，Database. Document.Course 在文档库中只出现 2 次，为低频路径；而 Database.Document 在文档库中出现 4 次，是高频路径。

对于查找高频路径，首先从长度最长的路径开始判断，一直执行到长度＝2 为止。先检查 Hash table 中长度为 n 的路径是否为高频路径，只要一路径为高频路径，它的子连续路径也会是高频路径，所以不必再对这些子连续路径进行判断，以减少从 Hash table 中重复判断的时间。如果路径次数小于阈值，表示该路径为低频路径，将此路径从 Hash table 中删除。但是由于某一低频路径的子连续路径可能与其它事务或者该事务的子连续路径次数之和累加后成为高频路径，因此判断低频路径时，要将此路径进行分解再加入 Hash table 中。

3.2 利用高频路径产生索引机制

XML的路径索引必须根据输入的路径来判断文档库中有没有该路径再加以存取，可以将索引看成是一个路径识别器。利用自动机理论[5]，为所有被索引的路径建立一最小化的有限状态识别机，再将文档库对应的路径文件用B^+树存储，以加快查找。当用户查询时，有限状态识别机先判断查询路径(存在于文档库中)，再根据识别的状态号码进入B^+树查询对应此路径的文件。

因为高频路径的所有子连续路径也是高频路径，所以要对算法得到的高频路径扩展(找出所有高频路径的子连续路径)以得到全部的高频路径。对图 1 的文档库经过扩展得到的所有高频路径的集合为{ DBLab.Paper.XML , Database.DBLab.Paper,DBLab.Paper, Paper.XML, Database. DBLab, Database. Document, Database.Music, Music.Pop, Ftp. Music}，然后将这些路径视为一个 regular set，通过自动机的转换，产生最小化有限状态机(MFA)，作为索引构造的条件。图 2 就是根据所有的高频路径产生的最小化有限状态机及其索引结构。

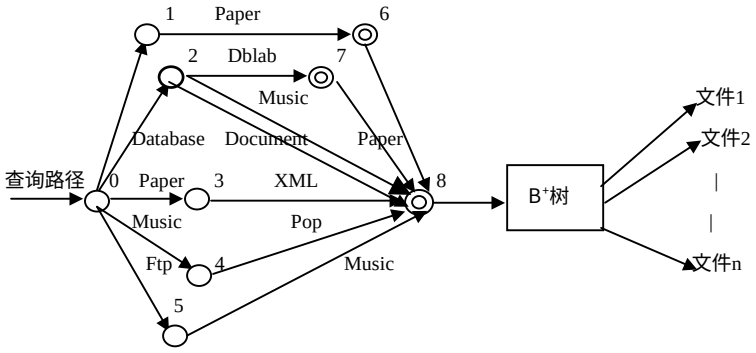

图 2 产生的最小化有限状态机和索引结构

3.3 扩展的倒排文档

对于 XML 文档，首先识别和抽取每一个元素和元素中的词条，并针对抽取出来的每个词条和元素标志，取有效词(Stop Words)，并抽取词根(stemmer)；然后统计有用的词条，生成一个倒排文档，并得到 XML 文档的概要树。然后把概要树的结构和内容数据分离开来，其中内容数据是指向新生成倒排文档的原始文本，结构则指向路径索引。文档的 id 号和相关路径索引的节点标记记录在置入列表中(置入列表存储在倒排文档中)，所生成路径索引的每个文件都有指针指向相应的置入列表，可以在结构和内容上同时检索。它的模型如图 3 所示。

XQuery 的核心是 XPath，XQuery 查询的核心是路径表

达式查询。但以往的系统只注重 XML 的结构性，而忽略了 XML 数据的文本性。算法 1 给出了把这两种查询方法相结合的方法。

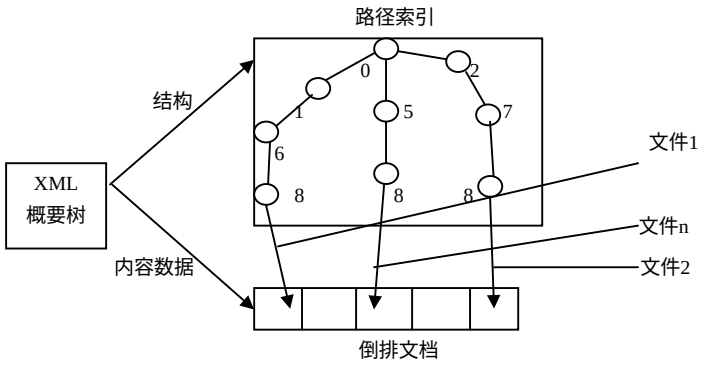

图 3 扩展的索引结构的模型

算法 1 双重索引查询算法

//输入：由连接的查询termX_1,X_2,…,X_n组成的query(Q)

//输出：查询结果集 S

[1] Query－Function(Q){

[2] Set S＝∮;

[3] 把 Q 分为几个连接的查询 term;

[4] For each term X_i,(1≤i≤n) of Q do {

[5] 在倒排表中找出和term X_i path匹配的集Ti ；

[6] 在文档词条中找出和term X_i path匹配的集A_i;

[7] Extract (置入表中指向X_i litetal部分指针的元素) from T_i 集合;

[8] Set 文档列表为集B_i;

[9] }

[10] }

[11] 结果集 S 为 $S=\bigcap_{i=1}^{n}(A_i \cap B_i)$

4 实验分析

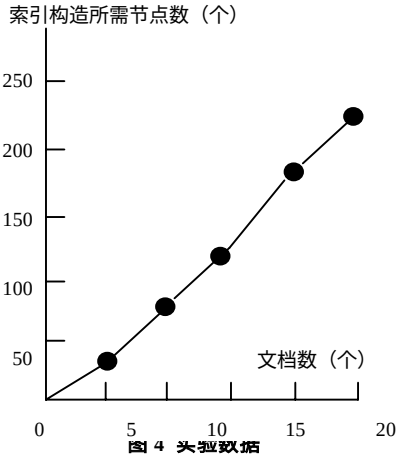

图 4 实验数据

在此采用通用标准数据 Canterbury Corpus 作为我们的实验数据，可在“ http://www.racai.ro/EUROLAN-2001/page/resources/software/windows/XMLTools/MLTools-10/docbook/test/ ”中下载，有 20 多个 XML 文档。 所有的实验过程是在 Pentium4 2.4GHz,256MB 内存，Windows 2000 Server 操作系统上的机器上完成的。用 VC++编程实现。试验数据如图 4 所示。

当用 bib.xml 等 4 个 XML 文档试验时，总节点的数目为 298，如图 5 所示。此时索引构造所需节点数不到 40 个，可大大减少索引构造的时空开销。基于双重结构查询的精确度很高，且查询的回馈时间范围在 0.1s~0.25s 之间。

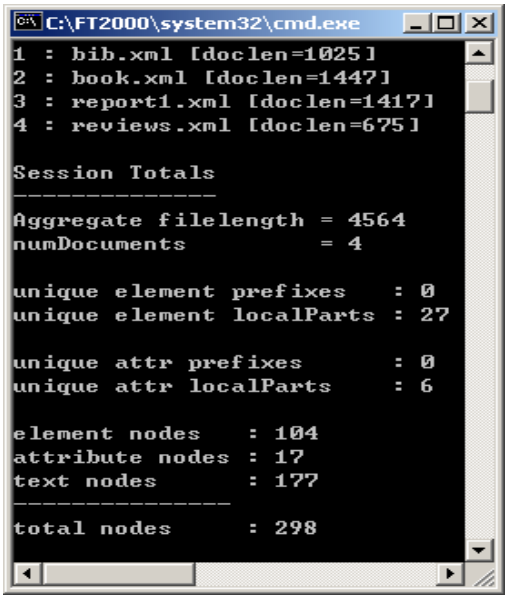

图 5 XML 个数为 4 时总节点数目

5 结论

本文提出了一种基于高频路径的路径索引的构造方法，本方法只需对文档库扫描一次，可大幅降低用户查询的时间。并把路径索引集成到基于传统文本检索的倒排文档中，有效地利用了 XML 文档的特点，其中路径索引包含了规范的标记，这种特性有利于内容和结构的相似性检索。

参考文献

1 Shimura T, Yoshikawa M, Uemura S. Storage and Retrieval of XML Documents Using Object-relational Databases[C]. Proc. of Int'l Conf. on Database and Expert Systems Applications, 1999: 238-242.

2 Wang Xiaoling, Wen Jirong, Liu Wenyin, Enhancive Index for Structured Document Retrieval[EB/OL]. http:// research.microsoft.com/asia/dload_files/group/mediasearching/2002p/10_wang.pdf, 2002.

3 Daniela F, Donald K, Loana M. Integrating Keyword Search into XML Query Processing[C]. Proceedings of then International World Wide Web Conference, 2000: 119-135.

4 Kotsakis E. Structured Information Retrieval in XML Documents[Z]. http://citeseer.nj.nec.com/cs.

5 Lewis H, Papadimitriou C. Elements of the Theory of Computation[M]. Prentice Hall, Inc, 1998.