一种基于 s-DOM 的 XML 文档索引算法

王申康, 张雪燕
(浙江大学 计算机学院, 浙江 杭州 310027)

摘　要: 改造 XML 树模型是提高 XML 查询效率的重要方法。通过分析现有的索引算法, 对 XML 树模型进行了改造, 提出了基于 Signature 的索引策略(s-DOM) 。采用该策略预处理 XML 文档可以大大缩小搜索范围, 从而提高了查询的效率。

关键词: XML; Signature; s-DOM
中图法分类号: TP301. 6　　**文献标识码**: A　　**文章编号**: 1001-3695(2005) 02-0087-03

A s-DOM-based XML Indexing

WANG Shen-kang, ZHANG Xue-yan
(School of Computer Science , Zhejiang University, Hangzhou Zhejiang 310027, *China)*

Abstract: Revising XML tree model is an important method for improving XML query efficiency. Analyses various indexing algorithm and proposes a signature based DOM indexing approach by revising XML tree model. Pre-processing XML documents with this approach can greatly cut down number of documents for query, thus improved query efficiency.
Key words: XML(eXtensible Markup Language) ; Signature; s-DOM

XML(eXtensible Markup Language) 是一种在 Internet 上传播的标记语言, 它不仅有效地解决了数据的表示问题, 而且很好地解决了不同系统之间的复杂数据交换问题。但由于 XML 比普通的文档多了用户自定义的标签, 增加了文档的长度, 当要查询 XML 文档内容的时候, 使用常规的文本查询算法将降低查询的效率。

自从 XML 出现以来, 很多学者对它的查询算法进行了研究, 并取得了不少的成果。特别就 XML 的树模型方面, 提出了有效的解决方法。例如 Serge Abiteboul 等人提出的受限的正则路径查询(Regular Path Queries with Constraints) [1]。R. Sacks-Davis 等人针对结构化文档索引的一般模式, 提出了两种索引方法: 以词为单位的基于位置(Position) 的索引算法和基于路径(Path) 的索引算法 [2]。Quanzhong Li 等人用 XISS (XML Indexing and Storage System) 来索引和存储 XML 数据[3]。Flavio-Rizzolo 等人建立的 TOXIN(Toronto XML Indexing Engine) 系统对 XML 文档进行索引[4]。Dao Dinh Kha 等人提出了 RRC(Relative Region Coordinate) 算法建立相对稳定的索引结构[5]。这些算法在 XML 文档树模型的基础上采用不同的索引策略, 有效地提高了 XML 文档的查询效率, 但均未考虑到以下情况, 即查询是否必须访问到叶子节点才能结束。若能解决这一问题, 将极大地提高 XML 文档的查询效率。

1　XML 查询

1.1　XML 文档的树模型

XML 的主要优点在于 XML 文档的树模型, 本文的查询基于此树模型, 并且是一个有序的带标志树。

∑和 r表示两个不相交的字符集, 文本节点内容在∑上取值, 复合节点标志在 r上取值。则对 XML 文档结构的抽象定义如下:

定义 1　文档结构是五元组 $D = (V_S, V_M, A, R, PL)$。其中,

(1) V_S, V_M, A, R 是一树集, V_S, V_M 分别表示不相交的文本节点集和复合节点集, A 表示边, R 表示节点之间的关系;

(2) PL 为映射: $V_S \cup V_M \cup A \rightarrow \sum \cup$ r。

以上的定义将文档抽象为带标志的有向树, 在此树中, 节点和边用来模型化文档的逻辑结构, 树的叶子是实际上的文本内容。

1.2　树查询

在 XML 的树模型下, 查询的目的就是从 XML 文档或 XML 文档集中检索出某种形式的子树, 根据这个思路, 引入树查询的概念。

定义 2　树查询是具有形式 $Q = (P \wedge C, x)$ 的查询。其中, P 是原子公式的合取并包含子公式 Ancestor(X_i, X_{i+1}) Parent(X_i, X_{i+1}) ($0 \leq i \leq n - 1$) ; C 是原子约束的合取并 C 中包含于 P 中的自由变量集合(记为 FreeVar(C) ∩ FreeVar(P)) , 以使下列条件满足: v 一被称为 Q 的根变量; $x \in$ FreeVar(Q) , 使得对P $y \in$ FreeVar(Q) 存在变量序列 $X = X_0, \cdots, X_n = y$($n \geq 0$) 。

由上面的定义直接可以得出下面的结论:

设 $Q = (P \wedge C, x)$ 为树查询, 则

(1) P 中没有任何 Ancestor 环, 即不存在任何变量序列 $X_0, X_1, \cdots, X_n, (n \geq 0)$ 使得对于 $0 \leq i \leq n - 1$, P 包含子公式 Ancestor(X_n, X_0) ;

(2) Q 的根是唯一的;

(3) 对于每一 $y \in$ FreeVar(Q) , P 中最多存在一个形式为 Parent(x, y) 的公式;

(4) 如果对于所有的 X_i($0 \leq i \leq n - 1$) , Q 都有关系 Ancestor (X_i, X_{i+1}) , 则 P 中最多只有一个形式为 Ancestor(x, y) 的公式。

以上理论是 XML 查询算法的基础。下面我们将描述

收稿日期: 2004-03-11; 修返日期: 2004-05-28

s-DOM索引策略。

2 用 Signature 技术提高查询效率

2.1 Signature 的概念

Signature 技术[6]广泛用于全文检索中,它的核心思想是用 Hash 函数对文档块中的每个字或词进行处理,产生的 Hash 值就是这个词的 Signature。为了方便起见,我们以后就称其为 Hash 值。文档 D 的 Signature 的定义如下:

定义 3 设文档 D 中不同个数的元素为 n 个,对应的 Hash 值分别为 $H_1, H_2, \cdots, H_n$。则文档 D 的 Signature $S_D = H_1 \vee H_2 \vee \cdots \vee H_n$。

本文中我们设 Hash 值的长度为 F,1 的个数为常数 m。产生 Hash 值,先要决定 F 和 m。本文采用 Faloutsos 在 1985 年提出的方程式来计算这两个要素[6],其中 L 是文档中不同字或词的个数,f 是误差(False Drop),

$$\log_2 f = -F/(L \cdot \log_2 e), \quad m = (\text{int})\ [F/(L \cdot \log_2 e)]$$

下面是本文算法的主要依据:

定理 1 设 S_D 为文档 D 的 Signature,$H_i(i = 1, 2, \cdots, n$,其中 n 为文档中词的个数) 是文档中各个节点的 Hash 值;则对某个词 w 的 Hash 值为 h,如果 $S_D \wedge h! = h$,那么 w 必定不在文档 D 中。

证明:假设 w 在 D 中,那么 $S_D = S_D \vee h$ 必定成立。这样我们有 $S_D \wedge h = h$,但这与我们的条件不符合,因此 h 必定不在 S_D 中。

2.2 采用 s-DOM 技术进行查询

采用 Signature 技术的查询过程如图 1 所示。

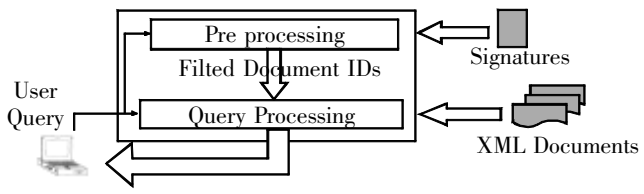

图 1 采用 Signature 技术的 XML 查询过程

下面通过例子来说明建立 Signature 的过程。图 2 是下述 XML 文档对应的 DOM 树。限定 f 为 0.2,由 2.1 节知 $L = 8$,则 $m = 2$,得到的 Hash 值如表 1 所示,这样就建立了文档的 s-DOM 树模型。

```
< ? xml version = " 1. 0"  encoding = " gb2312" ? >
< Families >
  < person >
    < name > 张小明 < / name >
      < father >
        < person >
          < name > 张大明 < / name >
        < person >
      < / father >
      < mother >
        < person >
          < name > 王宁 < / name >
        < / person >
      < / mother >
    < / person >
    < person name = " Carrie" >
      < wife >
        < person >
          < name > Lucy < / name >
        < / person >
      < / wife >
      < child >
        < name > Jerry < / name >
      < / child >
    < / person >
    < person >
      < name > Old David < / name >
      < brother >
        < person >
          < name > Mike < / name >
        < / person >
      < / brother >
    < / person >
< / Families >
```

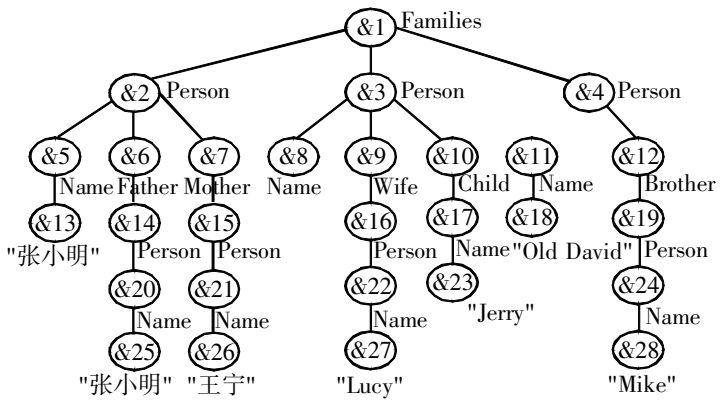

图 2 XML 文档的 DOM 树

根据图 2 中各个节点的 Hash 值(表 1),得到文档的 Signature 为 11101011。下面来初步验证一下 2.1 节中阐述的理论。Child 的 Hash 值是 00100010,00100010∧11101011 = 00100010; Children 的 Hash 值(在另外一个文档中计算出来的) 为 10010000,而 10010000∧11101011 = 10000000,显然不等于 10010000,则可判定 Children 不在此文档中。由此无须继续任何搜索,即可判定其不在本次查找的文档中了。但是需要注意的是,当 $S_D \wedge h = h$ 时,并不能就此判定其就在此文档中。如在本文实验所用的另一个文档片段中的一个词 Bible,它的 Hash 值为 11000000,11000000∧11101011 = 11000000,但 Bible 并不在图 2 所示的文档中。

2.3 改进的 s-DOM 技术

到目前为止,给定一个待查询的对象,把它的 Hash 值和文档 D 的 Signature 进行比较,就可以初步判断该对象是否在 D 中(误差为 f)。但这还是不够的,当 D 的 DOM 树比较复杂的时候,下一步的工作还是很费时,所以考虑改进 s-DOM 模型。

定义 4 $H_{i1}, H_{i2}, \cdots, H_{ik}$ 为某条路径上的节点的 Hash 值,则这条路径的 Signature(Path Signature, S_P),$S_{Pi} = H_{i1} \vee H_{i2} \vee \cdots \vee H_{ik}$。

由定理 1 可得到结论:若 $S_{Pi} \wedge H_{ij}! = H_{ij}$,则 H_{ij} 必定不在第 I 条路径上,反之不一定。

在具有 n 个叶子节点的 DOM 树中,共有 n 个 S_P,$S_D = S_{P1} \vee S_{P2} \vee \cdots \vee S_{Pn}$。这样在查询过程中通过先与 S_D 比较,再与 S_P 进行比较,就可以达到减少搜索范围的目的。表 2 为图 2 的各条路径的 S_P 值。

表 1 Hash 值表

节点名	Hash 值	节点名	Hash 值
Families	01001000	Mother	00000011
Person	10001000	Child	00100010
Name	01000010	Brother	10100000
Father	01000001	Wife	00001001

表 2 S_p 值表

S_p名	S_p值	S_p名	S_p值
S_{p1}	11001010	S_{p5}	11001011
S_{p2}	11001011	S_{p6}	11101011
S_{p3}	11001011	S_{p7}	11001010
S_{p4}	11001010	S_{p8}	11101010

3 部分实验结果

本实验的环境是:CPU 赛扬 500,内存 256MB 和 Windows 2000 Professional, Java 2 SDK, Standard Edition v. 1. 3。

本文主要对基于路径的查询算法进行了大量的模拟实验。限于篇幅,仅以引言中提到的 Serge Abiteboul 等的 CRPQ 为例(表 3)。其中实际选中文档是存在该节点的文档/文档总数,其他的是指该算法选中的文档/文档总数;查询栏中 a/b 指

Parent(a, b) , a//b 指 Ancestor(a, b) 。

表 3　Signature 技术的筛选率(%) - (32 位 Signature)

查　询	CRPQ	s-CRPQ	实际选中文档
/set/book/title	100	61	54
/set/book//title	100	44	28
/set/book/body	100	100	0
/set//title//procedure	100	48	0
//li/_/media	100	53	25
//title/book	100	63	0
//title//book	100	43	0

由表 3 可以得出: 采用了 s-DOM 技术的 CRPQ 在预处理时可以减少查询的范围, 从而提高查询效率。对其他查询算法实验的结果也得到类似的结论。Path Signature 能减少在一个文档内的搜索范围, 由于篇幅有限, 不在此列出, 在我们的实验中平均可以减少 70% 的搜索范围, 但由于它和具体的文档结构及查询类型有很大关系, 因此指标不是很稳定, 还需要进一步论证。

4　结论

以上论述的是我们所做的主要工作。采用 s-DOM 技术, 需要增加计算 Signature 及 Hash 值的时间和增加额外的存储空间。就某些对存储空间有要求的系统(如嵌入式系统) 而言, 尤其需要考虑这些因素。但总的来说, 在考虑查询算法的时候, 在系统条件允许的情况下, 引进 s-DOM 技术是利大于弊的。本文的后续工作是采用反向索引方法[7] 进一步改造基于 s-DOM 索引技术的树模型。

参考文献:

[1]　erge Abiteoul, Victor Vianu. Regular Path Queries with Constraints [C] . The 16th ACM SIGACT-SIGMODSIGART Symposium on Principles of Database Systems, 1997. 122-133.

[2]　R Sacks-Davis, T Dao, J A Thom, *et al*. Indexing Documents for Queries on Structure, Content and Attributes[C] . Proc. of International Symposiumon Digital Media Information Base(DBIM) , Nara, Japan, 1997. 236-245.

[3]　Quanzhong Li, Bongki Moon. Indexing and Querying XML Data for Regular Path Expressions[C] . Roma: Proceedings of the 27th VLDB Conference, 2001. 361-370.

[4]　Flavio Rizzolo , Alberto Mendelzon. Indexing XML Data with ToXin [DB/ OL] . WebDB, 2001.

[5]　Dao Dinh Kha, Masatoshi Yoshikawa, Shunsuke Uermura. An XML Indexing Structure with Relative Region Coordinate[C] . Proceedings 17th International Conference on Data Engineering, 2001. 313-320.

[6]　Christos Faloutsos. Signature Files: Design and Performance Comparison of Some Signature Extraction Methods[C] . SIGMOD Conference, 1985. 63-82.

[7]　Chiyoung Seo, Sang-Won Lee, Hyoung-Joo Kim. An Efficient Inverted Index Technique for XML Documents Using RDBMS[J] . Information and Software Technology, 2003, 45: 11-22.

作者简介:

王申康(1945-) , 男, 教授, 博士生导师, 研究领域为数据库理论、智能信息系统、人工智能、多媒体技术; 张雪燕(1977-) , 女, 硕士, 研究领域为嵌入式系统。

(上接第 73 页)由 Web 服务器、GTS 应用层网关、Web Session 管理器、HTML 模板系统、用户管理应用服务器、文献检索应用服务器、原文订购应用服务器、账务管理子系统等几个模块组成。中心站点不仅通过 Web 向因特网用户提供各种服务, 而且还通过专用控制台工具向系统管理员提供全局管理功能。每个“科技文献共建共享”参加单位都建立一个分中心站点, 分中心站点服务器接收中心站点传来的原文订购请求, 通知原文提供单位向订购原文的用户发送原文。中心站点与分中心站点之间、中心站点的各个业务逻辑服务器之间都是通过消息中间件 ISMQ 相互通信。利用消息中间件连接整个系统的组成部分。

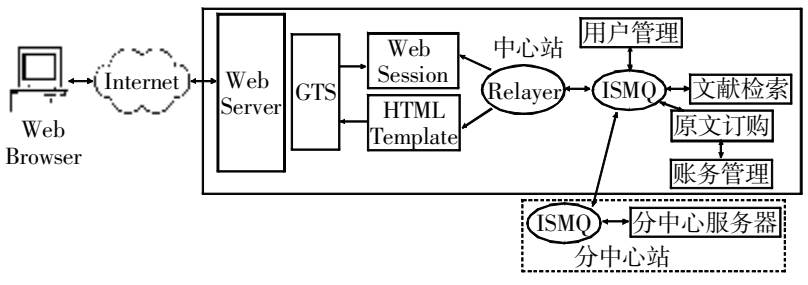

图 6　系统体系结构示意图

4　结论

在多层结构的框架中, 基于组件的软件重用思想为我们展示了一种崭新的软件设计思路, 以组件对象为中心的设计方法把硬件以芯片为中心的思想恰如其分地融合于软件的面向对象的分析、设计和施工之中, 使面向对象的概念和方法从工具语言的层次一下跃上了系统的应用层。

消息队列是软件组件间通信的重要方式, 与 RMI, RPC 等通信方式相比, 利用消息中间件通信, 具有以下特点: 系统的各个处理阶段的服务器进程可以异步运行; 服务流水线的拓扑结构没有特殊限制; 向业务逻辑代码屏蔽网络复杂性; 实现异构平台的互操作; 利用持久队列可以确保通信的可靠性等。

基于消息队列系统集成, 是以消息中间件作为主要的通信中间件, 集成各种软件组件、遗留系统, 建立具有多层结构的分布式应用系统。基于消息队列的应用系统中, 服务请求的处理过程被划分为若干个阶段, 不同阶段的处理由不同的应用服务器完成, 这些应用服务器按照流水线方式互相协作, 它们之间通过消息中间件互相耦合。基于消息队列系统集成技术充分利用了消息中间件的特点, 不仅结构灵活, 扩展性、伸缩性好, 而且还可以实现系统性能的自动优化、服务器质量控制等功能。

基于这样的设计我们已经成功地实现了国家科技文献服务网(NetDoc) 系统。

参考文献:

[1]　eri Edwareds. 3-tier Client/ Server at Work[M] . John Wiley & Sons, Inc. , 1997.

[2]　Jalote P. Fault Tolerance in Distributed Systems[M] . Prentice Hall, Inc. , 1994.

[3]　P A Bernstein. Middleware: A Model for Distributed System [J] . Services, Communication of ACM, 1996, 39(2) : 86-98.

[4]　中科院软件所对象技术中心. 消息队列中间件 ISMQ 的设计和实现[R] . 2000-10.

作者简介:

虞海江(1974-) , 男, 硕士研究生, 研究方向为网络分布式计算; 李京(1966-) , 男, 研究员, 博士生导师, 研究方向为软件体系结构、网络分布式计算、组合软件技术; 黄涛(1965-) , 男, 研究员, 研究方向为软件工程和方法学、软件工程环境和软件生产自动化、对象技术、分布式计算。