

基于元表的 XML 可查询压缩方法

姚英子, 蒋 溢, 熊安萍

(重庆邮电大学 计算机科学与技术学院, 重庆 400065)

摘 要: 将 XML 的树形结构转换为元表这种数据结构, 提出 XML 元表、结构统计表的概念及其生成算法, 结合元表来保存 XML 文档的结构数据, 设计一种基于元表的可查询 XML 压缩方法 MTQC (MetaTable Queryable Compression), 对 XML 数据进行压缩和查询。经实验证明, 在元表、结构统计表上的查询处理过程更加简单, 且 MTQC 具有较好的压缩性能和查询效率。

关键词: XML; 数据压缩; 查询处理; 元表; 结构统计表

中图分类号: TP311

文献标识码: A

文章编号: 1000-7180(2013)08-0151-04

XML Queryable Compression Method Based on Meta Table

YAO Ying-zi, JIANG Yi, XIONG An-ping

(Institution of Computer Science and Technology, Chongqing University
of Posts and Telecommunications, Chongqing 400065, China)

Abstract: This paper convert the XML's tree structure to meta table and propose a definition of XML metatable, structure statistics table and their generation algorithms, combined with the meta table to hold the structure of the XML structural data, designing a meta table-based querying XML compression method MTQC (MetaTable Queryable Compression), compression and queries on XML data. Experiments show that in the meta table, queries on structure statistics table processes will become more simple, and MTQC has an excellent compression performance and query efficiency.

Key words: XML; data compression; query processing; meta table; structure statistics table

1 引言

近年来 XML 已越来越广泛应用于互联网上各种数据交换与数据表示, 但是 XML 数据的自描述性使其存在大量冗余信息, 因此, 如何对 XML 进行高效的压缩, 同时又支持在不完全解压的情况下对 XML 进行查询, 是 XML 压缩技术必须实现的两项关键点。文献[1-4]对该项技术进行了有效研究。但压缩时间较长, 内存作用较多。本文提出一种新的压缩方案——基于元表的可查询压缩方法 MTQC, 并对 MTQC 进行试验验证。

2 基于元表的可查询压缩方法

2.1 相关概念

在 MTQC 里, XML 文档经 SAX Parser 解析器进行解析后, 分离原文档各节点的结构数据(标签名、属性名及节点间关系)和内容数据(元素内容和属性值)。结构数据存储在元表内, 元表中的父节点字段实现保存 XML 元素间的父子关系。结构统计表中的头结点将遍历后的 XML 文档标签 Id 依次保存, 内容数据存放至不同标签 Id 头结点的单链表中, 对单链表内的数据采用 7-Zip 压缩算法进行压缩。

收稿日期: 2012-10-30; 修回日期: 2012-12-28

基金项目: 重庆市信息产业发展专项基金资助项目(200811004)

是压缩过程,虚线对应的是查询处理.

2.3 元表、结构统计表的构建

构建元表、结构统计表的算法如下:

MetaTable_and_Structure_Statistics_Table_Construction
(SAX_Event)

```
{
    1. 解析 XML 文档;
    2. Id=1;
    3. While(! EOF)
    4. If(节点是新标签)
        {
            元表里添加一条记录,字段包括(TID, TNAME,
            TLEVEL,PID);
            TID++;
            将元表中该记录的 Id 字段添加至结构统计表头结
            点数组中,并将该表头结点链域设置为空;
            If(新标签包含数据内容)
                {
                    新建一数据结点,数据域存放数据内容,计数器
                    设置为 1,指针域设置为空,并将表头结点的链域指向该数
                    据结点;
                }
        }
    5. Else
        {
            If(该标签包含数据内容)
                {
                    If(表头数组标签 Id 头结点的链域为空)
                        {
                            新建一数据结点,数据域存放数据内容,计数器
                            设置为 1,指针域设置为空,并将表头结点的链
                            域指向该数据结点;
                        }
                    Else If (该表头结点的单链表中没有包含该数据
                    内容的数据结点)
                        {
                            新建一数据结点,数据域存放数据内容,计数器
                            设置为 1,指针域设置为空,并将单链表尾结点
                            指针域指向该数据结点;
                        }
                    Else If(该表头结点的单链表中已存在包含该数据
                    内容的数据结点)
                        {
                            该数据结点的计数器+1;
                        }
                }
        }
}
```

2.4 数据压缩

数据内容存放至结构统计表中,在经历结构统计表构建、原 XML 节点信息删除的情况下,数据压缩的对象也由原来的 XML 节点数据内容转变为结构统计表不同标签 Id 的单链表. 根据结构统计表构建效果可知,原 XML 文档中的重复数据在结构统计表数据结点的生成过程中已得到有效解决,因此,采用何种压缩方法以取得最高压缩率已不再是 MTQC 数据压缩追求的唯一目标,而是综合考虑高压缩率、较短压缩/解压时间以及较少的内存消耗. 现有压缩方法很多,本文采用 7-Zip 压缩方法正是基于以上三个因素考虑.

2.5 数据查询

MTQC 的查询处理步骤主要包括查询的验证、结构统计表映射、获取数据和输出查询结果.

查询算法如下:

```
Query( )
{
    1. 查询元表,根据查询语句末尾标签的父节点字段逆
    序查找至查询语句首个标签名,以此验证查询语句
    是否合法;
    2. If(查询语句种类是统计某标签或属性指定某个数
    据项出现次数)
        {
            根据查询语句末尾标签的 Id 号,查找结构统计
            表中对应头结点的数据链表,查询数据内容与查询语
            句数据项相同的数据结点,返回其计数值;
        }
    3. Else If(查询语句种类是返回某标签或属性包含的
    所有不同数据内容)
        {
            根据查询语句末尾标签的 Id 号,查找结构统计
            表中对应头结点的数据链表,返回链表中所有结点的
            数据内容;
        }
    4. Else If(查询语句种类是返回满足特定条件的标
    签名或属性名,如带有“=、!=、<、>”等比较运算
    符,以及“sum、average、minimum”等聚集函数查询
    语句)
        {
            根据查询语句特定条件,查找结构统计表中满足
            条件的数据链表的头结点,返回该标签 Id 的标签名;
        }
}
```

查询表达式首先经过查询验证器的验证,判断查询语句是否合法. 从查询表达式提供的条件,映射出所需记录所在的头结点,通过该头结点的数据链表检索出所求数据,最后将压缩记录解压并返回给

用户.

3 实验测试和分析

选取 XGrind 和 XMill 2 种压缩方法与 MTQC 进行对比实验,实验环境:所有实验都是在 CPU 为 1.73 GHz Genuine Intel(R) T2080,内存为 2 GB,操作系统为 Windows XP 的机器上进行的.数据集:采用 XBench^[5-6],这是一个反映多个应用领域 XML 数据的基准测试数据,本文实验数据部分来自 DBLP、Shakespeare、SwissProt 三个数据源.如表 2 所示.

表 2 测试数据集

数据集	大小/MB	元素个数	属性个数
DBLP	127	3 332 130	404 276
Shakespeare	7.5	179 690	0
SwissProt	109	2 977 031	2 189 859

3.1 压缩性能

压缩率方面:压缩率为衡量算法性能的一个依据,文中使用的数据压缩率的计算公式为

$$\text{压缩率} = \left(1 - \frac{\text{压缩后的数据总大小}}{\text{压缩前的数据总大小}}\right) \times 100\%.$$

实验结果:图 6 显示不同压缩方法对本文实验采用的 3 种数据集进行压缩后的结果.从中可以看出:MTQC 的压缩率与 XMill 较为接近,XGrind 最低.

结果分析:XGrind、MTQC 两种压缩方法其底层压缩方案分别为 Huffman 和 7-Zip,其中,XGrind 保留了 XML 文档的原始结构,没有分离数据和结构,该方法只针对 XML 文档的冗余标签进行压缩,没有解决文档中的路径和数据重复问题,故其压缩率远低于 XMill 和 MTQC.同时,MTQC 为了实现对压缩 XML 文档直接查询,在压缩过程中保存了 XML 文档的结构信息以及结构与数据内容之间的映射关系,导致其压缩率不能优于非可查询压缩方法 XMill.但由于 MTQC 采取重复路径相同标签的数据内容集中放入一个数据链表中的方式,利用 7-Zip 特别针对相同数据类型出色的压缩性能,对链表内数据进行高效压缩,其压缩率约比 XGrind 高 11.6%,接近 XMill.

压缩时间方面:

实验结果:图 7 显示不同压缩方法对 3 个数据集的平均压缩时间.实验表明:

XMill 与其他 2 种 XML 压缩方法相比所用压缩时间最短.

可查询压缩方法中 MTQC 比 XGrind 压缩时

间短.

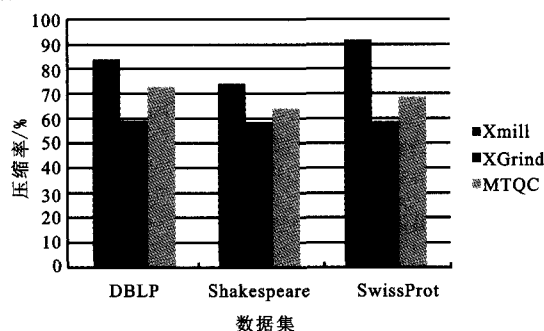


图 6 三种方法的压缩率对比

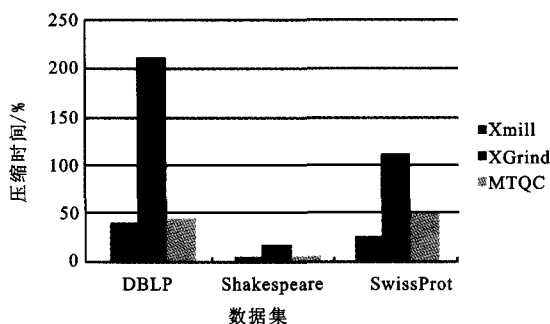


图 7 三种方法的压缩时间对比

结果分析:由于 XGrind 自身采用的底层压缩方案决定了其压缩时间略逊于 MTQC.其中,XGrind 采用的 Huffman 编码,需要对 XML 文档进行二次扫描来收集文档相关统计信息,故压缩时间较长.MTQC 索引结构统计表构造仅需扫描一遍 XML 文档,其构造时间复杂度为 $O(|V_t|)$, $|V_t|$ 表示原 XML 文档节点数目,由这些因素导致 MTQC 压缩时间开销较小.

3.2 查询性能

查询种类方面:XGrind 的查询处理器只能处理压缩数据的精确匹配和前缀匹配查询、解压数据的部分匹配和范围查询,而 MTQC 除了实现 XPath 大部分功能外,还扩展了统计各标签数据内容出现次数、快速返回各标签所有数据内容等查询功能.

查询时间方面:XGrind 的查询操作比 XMill 快约 2 倍,这是由 XMill 需要将文档完全解压后再执行查询处理造成的.与 XGrind 比较分析显示,MTQC 的查询效率更出色.

4 结束语

本文设计一种新的支持查询的压缩方法 MTQC.MTQC 通过对原 XML 文档进行一次 SAX 解析,利用创建的元表和结构统计表增加对查询的支

(下转第 159 页)

参考文献:

- [1] Murmann B, Boser B. A 12 bit 75 MS/s pipelined ADC using open-loop residue amplification[J]. IEEE J. Solid-State Circuits, 2003, 38(12): 2040-2050.
- [2] Kim J K R, Murmann B. A 12 b, 30 MS/s, 2.95 mW pipelined ADC using single-stage class-AB amplifiers and deterministic background calibration[J]. IEEE J. Solid-State Circuits, 2012, 47(9): 1-11.
- [3] Yuan J, Fung S W, Chan K Y, et al. An interpolation-based calibration architecture for pipeline ADC with nonlinear error[J]. IEEE Transaction on instrument and measurement, 2012, 61(1): 17-25.
- [4] Yuan J, Farhat N H, Van der Spiegel J. Background calibration with piecewise linearized error model for CMOS pipeline A/D Converter[J]. IEEE Transaction on instrument and measurement, 2008, 55(1): 311-321.
- [5] Sonkusale S R, Van der Spiegel J, Nagaraj K. Background digital error correction technique for pipelined analog-digital converters[C]//IEEE conference publications. Taipei, 2001: 408-411.
- [6] 石艳梅. 1.5 bit/级 pipelined ADC 典型单级电路结构

实现研究[D]. 成都: 电子科技大学, 2004.

- [7] Ozan E Erdoian, Paul J Hurst, Stephen H Lewis. A 12 b digital background calibrated algorithmic ADC with 90-dB THD[C]//IEEE International Solid-state Circuits Conference. Los Alamitos, 1999: 316-317.
- [8] 陈莉. 自适应滤波算法及应用研究[D]. 西安: 西安电子科技大学, 2006.
- [9] 彭蓓. 结合数字校正技术的纳米 CMOS 流水线 ADC 设计[D]. 北京: 北京工业大学, 2011.
- [10] Bang-sup Song, Michal F Tompsett, Kadaba R Lakshmikumar. A 12 bit 1 Msample/s capacitor error-averaging pipelined A/D converter[J]. IEEE journal of solid-state circuits, 1998, 23(6): 1324-1333.

作者简介:

宫月红 女, (1982-), 博士研究生. 研究方向为模拟混合信号电路设计与数字信号处理.

罗敏 男, (1978-), 博士研究生, 讲师. 研究方向为模拟混合信号电路设计.

(上接第 154 页)

支持, 方法实现简单, 压缩性能和查询性能也与现有几种 XML 压缩方法具有可比性. 本文尚存在一些不足之处. 首先, MTQC 采用的 7-Zip 压缩算法与主流的压缩方法相比, 还有一定差距. 其次, 生成的结构统计表在某标签所包含的数据项反复出现时, 查询效率表现优良; 但当标签数量较多, 且单个标签数据项重复出现次数较少, 则结构统计表对查询处理过程的辅助功能难以展现. 进一步的研究工作可以从两方面进行: 研究更有效的索引压缩算法; 研究如何提高结构统计表创建效率及实现部分更新, 使其性能更加出色.

参考文献:

- [1] Min J, Park M, Chung C. XPRESS: a queriable compression for XML data[C]//Proc of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, New York USA: ACM Press, 2003: 122-133.
- [2] Ferragina P, Manzini G. Indexing compressed texts[J]. Journal of the ACM, 2005, 52(4): 552-581.
- [3] Ferragina P, Luccio F, Manzini G, et al. Compressing

and indexing labeled trees, with applications[J]. Journal of the ACM, 2009, 57(1): 1-33.

- [4] Cheng J, Wilfred Ng. XQzip: querying compressed XML using structural indexing. [C]//Proc of the 9th International Conference on Extending Database Technology, Heraklion, Greece, Berlin, Germany: Springer-Verlag, 2004: 219-236.
- [5] Yao B B, Özsu M T, Khandelwal N. XBench benchmark and performance testing of XML DBMSs[C]//Proceedings of 20th International Conference on Data Engineering, Boston, MA: IEEE, 2004: 621-632.
- [6] 刘小珠, 彭智勇. 全文索引技术时空效率分析[J]. 软件学报, 2009, 20(7): 1768-1784.

作者简介:

姚英子 女, (1988-), 硕士研究生. 研究方向为数据库、信息检索.

蒋溢 男, (1969-), 高级工程师. 研究方向为面向服务计算、数据挖掘、软件工程等.

熊安萍 女, (1970-), 副教授. 研究方向为高性能计算、面向服务计算、软件体系结构等.