学校编码：10384　　分类号_____密级_____
学号：23020091152717　　UDC_____

厦门大学

硕 士 学 位 论 文

# 动态 XML 编码技术研究

# Research on Labeling Schemes over Dynamic XML Data

庄 灿 伟

指导教师姓名: 冯少荣 副教授
专 业 名 称: 计算机软件与理论
论文提交日期: 2012 年 5 月
论文答辩日期: 2012 年 6 月
学位授予日期: 2012 年 月

答辩委员会主席: _____
评 阅 人: _____

2012 年 6 月

# 厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规范（试行）》。

另外，该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年　　月　　日

# 厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（　　）1。经厦门大学保密委员会审查核定的保密学位论文，于2013 年 12 月 31 日解密，解密后适用上述授权。

（ √ ）2。不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年　　月　　日

# 摘 要

随着网络应用的快速发展，XML(eXtensible Markup Language)数据正成为主流的数据形式，如何对XML数据建立有效索引进而实现高效查询是当前的研究热点。大部分XML相关索引和查询技术基于某种对XML树的编码方法。XML编码方法保存了文档树的结构信息，使得在执行查询时不必遍历整个XML文档。传统的区间编码方法和前缀编码方法支持XML节点间位置关系和结构关系计算，但是不能有效处理文档更新，一旦更新发生，整个树需要重新编码，系统代价高。为解决该问题，研究人员提出了动态XML编码方法，包括浮点数区间、CDBS(Compact Dynamic Binary String)、QED(Dynamic Quaternary Encoding)以及DDE(Dynamic Dewey)等。动态XML编码方法一定程度上避免了文档更新时的重新编码，但仍存在时空开销大、对倾斜插入敏感、不能重用已删编码等问题。本文研究集中于动态XML编码机制的性能优化。

首先，XML 文档更新涉及节点插入和删除，当在删除位置插入新节点时，如果新节点能够对已删编码进行重用，则可以控制编码长度的增长速度，提高查询性能。CDBS 和 QED 的编码重用已经有相关研究，而对于 DDE 编码，却是一个难点。基于 Stern-Brocot 树，提出了 DDE 编码的改进方法——IDD(Improved DDE)。IDD 将最短位长中间编码赋予新节点，能够对已删编码进行重用，有效控制了删除和操作都发生的更新环境下 DDE 编码位长，提高了 XML 频繁更新时的编码效率和查询性能。

此外，针对已有动态区间编码方法普遍存在的初始编码空间复杂度高，倾斜插入编码长度增长迅速等问题，本文提出了新的适用于 XML 文档更新环境下的区间编码方法——DCLS(Dynamic Containment Labeling Scheme)。DCLS 利用整数进行初始编码，具有计算简单，额外空间复杂度低、存储效率和查询性能高等优点；同时，DCLS 将整数视为特殊向量，不仅支持文档更新，而且更新效率高，特别是倾斜插入时，DCLS 可以避免编码位长的快速增加。

实验结果表明，相比于已有动态 XML 编码方法，IDD 和 DCLS 有更好性能。

**关键词：** XML 文档更新；动态编码方法；编码重用；向量序

# Abstract

Along with the increasing development of Internet-based application, more and more information is being stored, exchanged and presented in XML format. The ability to efficiently index and query XML data sources become increasingly important. Most of XML indexing and querying techniques are based on labeling schemes which are designed to label the XML nodes so that both ordered and un-ordered queries can be processed without accessing the original XML file. Traditional containment labeling scheme and prefix labeling scheme support the computation of document order and structural relationships efficiently, however, it cannot avoid re-labeling in XML updates and has high update cost. Many dynamic labeling schemes, including Float labeling scheme, CDBS,QED, DDE and so on, have been proposed to effectively process updates in dynamic XML data. Compared to traditional static schemes, dynamic schemes support XML updates, but at the same time they need extra time and space cost, have a fast growth rate in label size when skewed insertions and cannot re-use the deleted labels. Our paper focuses on the optimization of the performance of dynamic labeling schemes.

Firstly, when a node is inserted at a place where a node has ever been deleted, it is natural for the labeling schemes to reuse the deleted labels to reduce the storage space and improve the query performance. The reuse algorithms for CDBS and QED have been studied while the algorithm for DDE is not considered in the previous researches. Based on Stern-Brocot Tree, we propose IDD scheme(Improved DDE) . IDD assigns new labels with smallest size and can reuse all the delete labels. In this way, IDD controls the label size increasing speed and enhance the query performance when nodes are deleted and inserted in the XML data.

Additionally, after pointing out the limitations of existing dynamic containment labeling schemes which include a high space cost when initial labeling and a fast growth rate in label size when skewed insertions and so on, we propose a novel containment scheme called DCLS to effectively process updating in dynamic XML data. DCLS labels for initial XML using integers, which yields low time and space cost, compact size and excellent query efficiency for static documents. Meanwhile, DCLS takes the integer as special vector, which can not only deal with the case of

document updating, but also achieve high query performance. Especially DCLS can effectively avoid the rapid increase of labeling size for the case of skewed insertions.

Experimental evaluations confirm the benefits of our approaches compared to previous solutions.

**Key words:** XML documents updating; Dynamic labeling schemes; Reuse the deleted labels; Vector order

# 目 录

# Table of Contents

# 第一章 绪论

编码机制是实现 XML 高效查询的基础，设计支持文档更新的动态 XML 编码机制具有重要的实际意义。本章首先阐述 XML 编码机制的研究背景——大规模 XML 数据的出现和 XML 数据查询的要求，接着引出本文的研究课题——支持文档更新的动态 XML 编码机制研究，最后对本文的主要研究工作和文章内容安排进行说明。

## 1.1 XML和XML查询处理

### 1.1.1 XML的出现及其特点

因特网的出现和发展改变了人们的生活和思维方式，相应地，与之相关的技术也得到飞速的发展。HTML( Hyper Text Markup Language, 超文本标记语言) 和 SGML( Standard Generalized Markup Language, 标准通用标记语言) 是其中两种重要技术，HTML 功能简单，但无法处理大量的结构化信息；SGML 功能完善，却非常复杂。基于 HTML 技术和 SGML 技术所存在的不足，XML (eXtensible Markup Language)技术得以出现，它即有 SGML 的强大功能和可扩展性，同时又兼具 HTML 的简单性。XML 具有以下几个特点[1-3]：

(1). 数据内容和显示格式分离：XML 标记语言将信息的数据部分和信息的样式部分进行了区分，XML 的重点是管理信息的数据本身，而不是数据的样式，数据的显示交给另外的技术解决，如 CSS、XSL 等。XML 这种明确的分工带来的是更高效的 Web 程序设计、更快的搜索引擎、更统一的数据表示和更方便的数据交换的出现。

(2). 结构化和自描述性：XML 在数据中附加标记来表达数据的逻辑结构和含义，保留了数据之间的语义关系，是一种机器和人都能看懂的语言。XML 的自描述性使其成为一种程序能自动理解的规范，便于机器的自动化处理。一个典型的应用是 Internet 环境中实现了数据的客户端处理。当 XML 格式的数据被发送到客户端，客户可以通过应用软件从 XML 文档中提取数据，进而对它进行编辑和处理，而不仅仅是显示结果。而

原来的 HTML 标记语言，即便是对一个字符的修改也都必须在服务器上进行，从而导致整个页面的重新传输。XML 数据将原来由服务器端处理的负载分配到客户端处理，从而降低了服务器的负担，优化了服务器的性能。

(3). 一种元标记语言：XML 是一种元标记语言，它允许用户自行定义适应于自己的标记，由于这种自定义性和可扩展性，使得它足以表达各种类型的数据。例如，企业可以用 XML 为电子商务和供应链集成等应用定义自己的标记语言；特定行业定义该领域的特殊标记语言，作为该领域信息共享与数据交换的基础，如数学领域的 MathML(Math Markup Language)、移动通信领域的 WML(WAP Markup Language)等。

(4). 跨平台性：XML 独立于任何操作平台，如操作系统、程序语言、各国语言、甚至应用程序。XML 提供了一种不同的数据源之间进行数据交换的公共标准，是一种公共的交互平台。

XML 上述特性使得 XML 技术得以广泛应用，如银行之间的数据交换、证券公司对其上市公司相关数据进行统计、企事业单位对其文件档案进行管理、以及电子商务、搜索引擎等。XML 技术在当前的互联网络和 IT 环境中扮演着越来越重要的角色，它已成为 Internet 数据表示和数据交换事实上的标准。

### 1.1.2　XML文档和XML树模型

XML 数据的基本形式是 XML 文档，一个 XML 文档通常包括 XML 声明、元素、属性、文本数据等。XML 声明一般出现在文档开头部分，包括版本号、可能的语言编码等。XML 通过元素组织 XML 数据，元素是 XML 文档内容的基本单元，一个元素包含一个起始标记、一个结束标记以及标记之间的数据内容。属性是元素的性质，在元素的开始标签内定义，一个元素可以定义多个属性，但它们的名称必须不同。一个元素的形式是：<标签 属性名=属性值> 数据内容 </标签>，其中数据内容可以是元素的值(即文本内容)或者其它元素，也可以是两者的混合。XML 元素之间形成嵌套包含关系，包含所有其它元素的元素称为根元素，每个 XML 文档有且仅有一个根元素。图 1.1 显示了一个 XML 文档示例。

尽管 XML 数据表现形式灵活，可以描述着复杂的结构，但其本质仍然是树状模型。一个 XML 文档可用 XML 文档树表示，树上每个节点对应于 XML 文

档元素、属性或文本(分别称为元素节点、属性节点、文本节点)，树上的边对应于节点间的父子关系。图 1.2 为图 1.1 对应的 XML 文档树，为了区分三种类型的节点，元素节点、属性节点和文本节点分别用三角形、圆形和长方形表示。

```
<?xml version= “1.0” encoding= “UTF-8”>
<articles>
      <article >
                <title>Bibliography on data design</title>
        <authors>
          <author    position= “1”> Karen Botnich </author>
          <author    position= “2”> Cola Cohen</author>
        </authors>
      </article>
      <article>
           <title>Native XML Databases Survey</title>
           <authors>
             <author    position= “1”>    Karen Botnich    </author>
           </authors>
      </article>
</articles>
```

**图 1.1.**　XML 文档示例

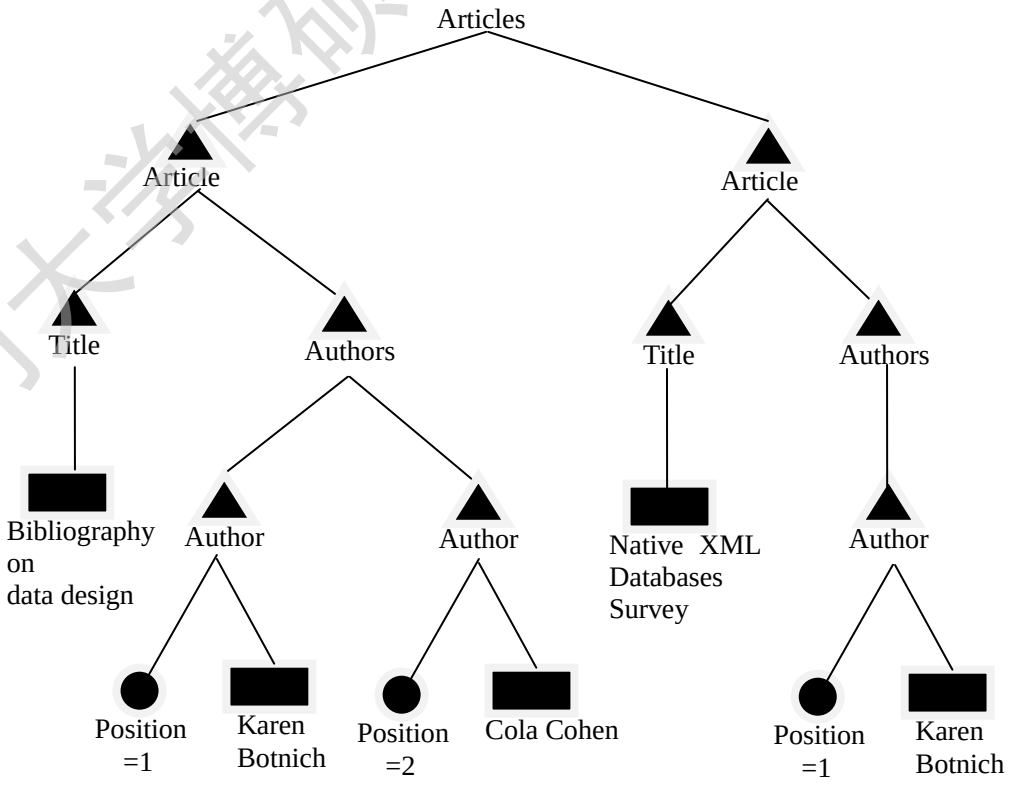


**图 1.2.**　XML 文档树示例