XQuery语言原理和实现技术

廖湖声 著

科学出版社

XQuery 语言原理和实现技术

廖湖声 著

科学出版社

北京

内 容 简 介

本书针对 XQuery 语言的原理，从抽象语法和形式语义出发分析了其语言特征，从数据库技术和编译技术两个角度全面介绍了 XQuery 语言的实现原理，深入讨论了适用于该语言的 XML 查询及其优化技术、查询计划描述语言的设计原理、编译优化技术和自动并行化方法，展示了 XQuery 查询引擎、XQuery 并行查询引擎、XQuery 编译系统等多种 XQuery 语言的实现方案，并且通过一个基于 XQuery 语言的网页编程工具的开发，介绍了面向应用领域的 XML 数据更新、脚本语言等多种功能的扩展方法。

本书适用于从事计算机科学与技术领域相关研究或开发工作的专业技术人员参考，也可以作为高等院校计算机软件与理论等相关专业研究生和本科生的教学用书。

图书在版编目(CIP)数据

XQuery 语言原理和实现技术/廖湖声著．—北京：科学出版社，2013

ISBN 978-7-03-037366-3

Ⅰ．X… Ⅱ．廖… Ⅲ．可扩充语言-程序设计 Ⅳ．TP312

中国版本图书馆 CIP 数据核字(2013)第 084275 号

责任编辑：魏英杰 杨向萍 / 责任校对：宋玲玲

责任印制：张 倩 / 封面设计：陈 敬

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

骏走印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2013 年 4 月第 一 版 开本：B5(720×1000)

2013 年 4 月第一次印刷 印张：14 1/2

字数：274 000

定价：58.00 元

(如有印装质量问题，我社负责调换)

前　言

随着计算机网络的普及和发展，信息技术已经广泛地应用于政府机关、学校教学、工矿企业、商业流通和人民生活的各个方面。通过计算机网络来共享信息、交换信息已经成为社会发展、生产活动和人民生活的组成部分。然而，不同的数据信息来自不同的提供者，也经常来自不同的计算机应用系统，因此往往具有不同的语义表示和数据组织，而数据格式的不同就是妨碍数据信息交换与数据信息共享的主要来源之一。为此，国际万维网组织（W3C）发布了可扩展标记语言（XML），以求实现网络数据交换和数据共享的标准化。经过多年发展，XML 得到广泛的应用，已经成为互联网中进行数据交换与数据共享的事实上的标准。与此同时，为了支持 XML 数据的查询和应用，2007 年国际万维网组织提出了 XML 数据查询语言的标准——XQuery 语言。近年来，XQuery 语言已经被产业界接受，几乎所有的数据库厂商都将 XQuery 语言的支撑功能扩展到其数据库产品中。

作为标准的 XML 数据查询语言，XQuery 语言提供了丰富的数据查询功能。同时，作为一种通用的 XML 数据处理语言，XQuery 语言提供 XML 文档构造功能和函数式程序设计功能，使得 XQuery 程序能够将面向 XML 数据的查询和处理结合在一起，更好地满足 XML 数据处理的需求。然而，鉴于 XML 数据是一种半结构化的数据，数据查询及其优化问题不同于传统的关系数据，多年来在数据库技术领域已经吸引了众多学者来深入研究 XML 数据的查询优化问题和数据存储问题。XQuery 语言所具有的丰富灵活的查询功能和计算能力对各种 XML 查询技术、查询优化技术以及语言实现技术的研究提出了更多的要求，带来了不少新问题。同时，XQuery 语言作为一种图灵完全的函数式程序设计语言，理论上可以描述任何 XML 数据处理算法。但是，在编译技术领域长期妨碍函数式程序设计语言发展的性能问题对于 XQuery 语言也同样存在。同时，函数式程序内在的并行性为多核环境下开发高性能的 XQuery 引擎提供了新的机会。因此，如何在同一语言实现框架内解决高性能 XML 查询处理和高效的程序执行，是 XQuery 语言在高性能引擎开发中面临的首要问题。在 XQuery 语言的相关研究中，人们习惯于采用数据库查询语言的实现方法，将 XML 查询技术的研究成果融入查询代数、查询计划的实现框架中。相比之下，本书试图将函数式语言的实现技术、数据查询语言的实现技术和 XML 查询技术结合起来，全面地探讨 XQuery 语言的实现技术。

另一方面，由于 XQuery 语言的应用发展，人们也在试图扩展这种技术，使其

满足更多的应用需求，以求发挥 XQuery 语言的查询描述能力和算法描述能力，为更多的应用领域提供先进的软件开发技术与工具。这些技术领域包括数据集成技术、分布式数据查询技术、服务组合技术、浏览器/服务器软件开发技术。这种应用发展趋势也为 XQuery 语言的实现提出了更多的技术问题。鉴于 XQuery 语言是一种函数式语言，其声明型语言的特征表现在使用者仅需描述数据查询和计算处理的需求，而不必关心算法实现的细节。因此，人们希望在更多的应用领域使用 XQuery 语言，代替传统的过程型程序设计语言，以减轻软件开发的难度。为了适应不同应用领域的需求，需要扩展的语言功能涉及 XML 数据更新、XML 脚本语言、分布式计算功能和服务计算功能等。

本书针对发展 XQuery 语言所面临的上述问题，在语义模型的基础上，探讨相关的数据库技术和语言编译技术，并且将两类技术统一在 XQuery 语言的实现原理中，比较完整地介绍了相关技术的发展和多种解决方案。

全书的组织结构如下：

第一章介绍 XQuery 语言的基础知识、发展趋势和实现技术的概貌。第二章根据 XQuery 语言的规范，介绍主要语言功能和处理模型，进而详细说明 XQuery 语言的抽象语法和形式语义，为 XQuery 语言的实现提供理论基础。第三章介绍用于支持 XQuery 语言实现的中间语言 FXQL。该语言在 XQuery 语言的多种实现方案中都用作查询代数描述语言。面向 XQuery 查询的查询代数及其表示和翻译方法也在本章得到深入讨论。第四章介绍适应于 XQuery 查询优化的逻辑优化技术和中间语言的代码优化技术。第五章讨论 XML 查询的核心操作——树模式查询及其语言支持，分析 XQuery 程序中的树模式查询特征，说明 FXQL 语言用于支持树模式查询的数据模型和语言功能，以及树模式识别算法。第六章重点介绍树模式查询的实现算法。首先分析现有算法的原理，进而给出一种改进的查询算法。随后，针对 XQuery 程序的特点，介绍多级树模式查询的概念和实现方法。第七章讨论 XQuery 编译系统的实现方法，介绍支持程序执行的抽象机模型，以及从 FXQL 语言到抽象机指令、从抽象机指令到 Java 字节码的翻译算法，并讨论运行时模块编译的实现方法。第八章给出 XQuery 语言的一个并行实现方案，讨论发掘程序中任务并行、数据并行、流水线并行等各种任务的方法，介绍一种基于代价和任务依赖关系的综合调度算法，实现各种并行执行方式的任务调度。第九章围绕一种基于 XQuery 语言的网页开发语言，介绍扩展 XQuery 语言的方法。系统设计与实现中包括 XML 数据更新功能、脚本语言功能的扩展，也包括浏览器对象模型和事件响应机制的支持，以及专用运行环境的设计。

本书内容来自作者所在课题组的研究成果，参与相关工作的师生主要有高红雨、苏航、廖伟、任宇、汤林、袁飞、陈悦、张慧涛、高增琦、杨科朝、李小青、张晓博、张开练、王磊、李燕宾、陈友杰、陈君鹏、李发金等。课题的研究得到了北京市自然科

学基金项目的多次支持。在此向对有关研究工作给予支持和作出贡献的所有人，表示衷心的感谢。

本书内容虽然涉及 XQuery 语言技术的多个方面，但仍然不能涵盖所有的相关技术领域，仍然需要在今后的研究中加以发展、改进和提高。XML 数据库技术和计算机语言技术的发展也不断地提出新的问题，需要不断地探讨和发掘。希望得到读者、研究人员和业界专家学者的批评指正。

廖湖声

于北京工业大学

2012 年 12 月

目　录

第1章 绪 论

1.1 引 言

随着计算机网络技术的飞速发展，互联网已经成为获取和发布信息的重要渠道，越来越多的信息通过网络进行交换和共享。为了充分有效地表示网络上丰富的数据，国际万维网组织(W3C)提出可扩展标记语言[1~3](extensible markup language，XML)作为信息共享的格式。XML 作为一种可扩展标记语言，凭借其跨平台、可扩展的优势，得到了广泛的应用，已经发展成为互联网上数据描述与交换的事实标准。同时，各大数据库厂商也纷纷推出了 XML 数据库系统，越来越多的数据通过 XML 来表示和存储。在各种信息系统中，人们更加需要高效地查询和处理 XML 数据。

为了标准化 XML 数据查询和处理，国际万维网组织发展了 XQuery 语言[4,5]，于 2007 年 1 月将其作为标准的 XML 数据查询语言发布。XQuery 语言是一种功能强大的语言，能够用于各种 XML 数据源的查询，从 XML 文档中选择并提取出需要的数据，并把查询结果重新构造为用户所需的新的 XML 文档。同时，XQuery 语言又是一种图灵完全的函数式程序设计语言[6,7]，具有良好的数据组织能力和算法描述能力，在互联网软件开发中得到越来越广泛的应用。大量的 XML 数据及其越来越多的应用需求对 XQuery 语言的实现技术提出了越来越高的要求。

XML 数据的半结构化特征给高性能的 XML 数据查询提出了新的要求。面向关系数据库的查询代数和查询优化技术都无法直接应用于 XML 数据查询的实现。XQuery 语言的出现为 XML 数据查询提供了灵活的描述手段，其函数式程序设计功能能够将多种 XML 数据查询结合起来，从多个数据源选出需要的数据，完成必要的数据加工，构造出新的 XML 文档。因此，XQuery 语言实现不仅应该支持高性能的 XML 数据查询，而且也应该支持多种查询与加工方式的组合。同时，由于 XQuery 语言作为一种函数式程序设计语言，以函数调用和表达式作为基本的程序设计手段，所描述的程序具有内在并行性。在语言实现中，通过自动并行化来利用多核环境提供的并行计算能力，也是提高程序执行效率和加快 XML 数据查询的重要方法。另一方面，随着 XQuery 语言的应用扩展，XQuery 程序的规模越来越大，所描述的算法越来越复杂，这就使得高性能 XQuery 语言实现更加需要编译技术的支持。然而，鉴于 XQuery 语言经常应用于网络环境，查询程序经常是

动态生成的，有时也需要跨网络传输，致使 XQuery 语言的编译实现需要采用运行时的动态编译技术。综上所述，XQuery 语言的高性能实现具有多方面的技术需求，需要综合利用数据库技术、编译技术和并行处理技术，以求获得高性能的 XML 数据查询和数据处理，使其成为互联网环境下更为强大的软件开发工具。

1.2 XQuery 语言的基础知识

1.2.1 XML

XML 是标准通用标记语言（the standard generalized markup language，SGML）的简化版。它为数据的描述提供了一种既易于解读又易于机器识别的、文本化的数据编码方式，使得数据能够在异构平台中得到共享。在互联网系统中，网络计算机之间的数据交换和数据共享是一个重要问题。由于各种计算机系统可能采用不同的硬件平台和软件平台，数据描述语言和数据格式各不相同，异构环境中异构数据的共享需要一个统一标准来表示各种数据。针对互联网中的这种数据共享需求，XML 的设计重点在于数据表示的简单性、通用性和可扩展性。

考虑下面的 XML 文档案例：

```
<?xml version= "1.0" encoding= "gb2312"?>
<note>
    <to> 信息安全专业的学生</to>
    <from> 物理老师</from>
    <heading> 考试通知</heading>
    <reading> 4 月 2 日早 8 点在 3 教 402 室考试</reading>
</note>
```

从这个案例可以看出，XML 数据采用文本方式，数据格式简单明确，文档具有自描述性，各种标记名称直接说明了内容，而且不同的应用系统可以采用专用的标记来描述专用的数据信息。与 HTML 等网页语言相比，XML 更加严谨和规范，易于根据应用需求进行扩展，已经成为互联网事实上的数据交换和数据共享标准。

按照 XML 的规定，XML 文档由元素组成，每个元素具有标记（tag）和内容（content）。标记名是大小写敏感的，且每个元素都必须有开始标记和结束标记。开始标记和结束标记之间的数据称为该元素的内容。元素内容中可以包含文本或其他元素，而内部元素被称为该元素的子元素。每个 XML 文档中仅有一个根元素，其他标记必须封装在根元素之中，从而形成树型的文档结构。

下面给出一个简单的 XML 文档例：

```
<? xml version= "1.0" encoding= "UTF-8"?>
<catalog>
    <book type= "Computer">
        <title> XML Technology</title>
        <author> Serge Abiteboul</author>
        <price> 58</price>
    </book>
    <book type= "Math">
        <title> Basic Math</<title>
        <author<> Peter Buneman</author>
    </book>
</catalog>
```

其中,第一行是 XML 文档的开始元素,给出了版本号和编码方式。在这个树型的 XML 文档中,<catalog>是根元素的开始标记。这个 catalog 元素中包含两个标记为 book 的子元素,表示两本书的信息。book 元素具有一个名为 type 的属性和若干个子元素,分别给出了书名、作者和价格等信息。由此可见,XML 以一种层次化名值对的方式,描述了数据信息。除了例中的各种 XML 元素外,XML 文档中还可以包含注释、处理指令等特殊的 XML 数据项。表 1.1 给出了各种 XML 数据项的说明。

表 1.1 各种 XML 数据项

XML 数据项	格式	注释
XML 声明	<? xml…? >	文档首行
元素	<tag> 内容</tag>	内容可以是文本或元素
属性	name= "value"	—
文本项	<! [CDATA[文本]]>	文本数据
注释	<! -注释信息 -->	—
处理指令	<? 指令 数据? >	应用领域专用
声明	<! 名字 数据>	专用名字的定义

完整的标记名采用“命名空间:局部名”的格式。从数据组织上来看,标记名具有说明元素内容的作用,使得 XML 数据成为一种自描述数据。从数据结构特征来看,XML 数据具有半结构化数据的特征。每个 XML 元素具有标记名,但标记名并未规

定元素的数据格式。例如，上例中 book 元素具有不同的子元素，子元素的标记名和数量都可能不一样，也就是说，不同的 book 元素具有不同的数据模式。针对数据模式的这种复杂性，XML 数据可以利用 DTD 和 XML Schema 等 XML 模式描述语言来描述 XML 数据的数据模式。按照 XML 数据的树型结构，上例中的 XML 文档可以被抽象为如图 1.1 所示的 XML 树。其中，每个节点对应一个 XML 元素、属性或文本，标注为标记名、属性名或文本，并且每个节点的子节点是有序的。

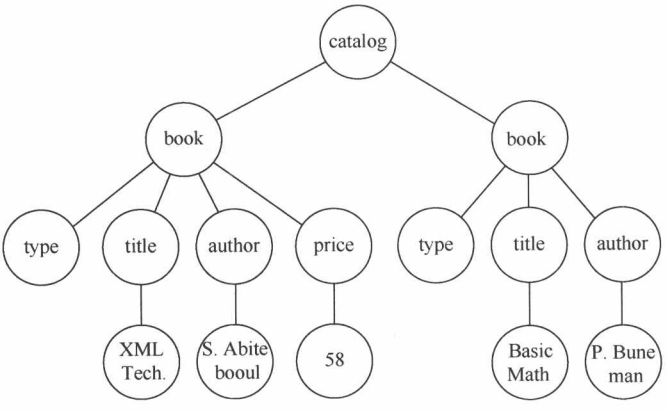

图 1.1 XML 树的案例

在 XML 的基础上，人们可以根据不同应用领域的需求，设计领域专用的数据描述语言来支持该领域的专用数据共享。具体方法如下：

① 规定应用领域的命名空间名。

② 使用应用领域的专业词汇来设计一组专用的标记。

③ 按照应用领域的信息组织，采用 DTD 或 XML Schema 来设计专用的数据模式。

通过这种方法设计的专用 XML 就可以描述该领域的专业数据，支持该领域的数据共享。人们已经设计出数百种专用 XML，如网页描述语言(XHTML)、Web 服务描述语言(WSDL)、同步多媒体集成语言(SMIL)、可缩放矢量图像标记语言(SVG)、数学标记语言(MathML)、生物信息序列标记语言(BSML)、人力资源管理标记语言(HRMML)、音乐标记语言(MusicML)、天文数据标记语言(AML)、化学标记语言(CML)等。这些基于 XML 的专用语言多数已经成为各个应用领域的信息描述标准。

1.2.2 XPath 语言

为了规范化 XML 数据的查询，1999 年 11 月国际万维网组织发布了 XPath 语言(XPath 1.0)[8]，作为标准的 XML 数据查询语言。XPath 1.0 已经成为 XSLT、

XPointer、XForm等多种计算机语言的子集，成为最常见的XML查询语言。2007年1月国际万维网组织又发布了XPath 2.0语言[9]，作为XQuery语言的子集。

针对XML数据具有的树型表示，XPath语言提供了一组轴操作，支持基于树节点结构关系的查找功能；祖先descendant、双亲parent、后继兄弟following-sibling、属性attribute等各种结构关系在XPath语言中都具有同名的轴操作。几个常用的符号是

/ 表示双亲子女关系

// 表示祖先子孙关系

. 表示当前节点

.. 表示双亲节点

@ 表示属性

* 表示任意元素

同时，XPath语言提供了基于标记名、属性、元素类型和各种谓词的多种选择机制，用于筛选符合条件的XML节点。对于以bookstore为根节点的文档，三个采用绝对路径方式描述的查询如下：

```
/bookstore/book[1]          bookstore 的第一个 book 子节点
/bookstore/book[price>35.00]
                            bookstore 的价格大于 35 的 book 子节点
//title[@*]                 文档中具有有属性的 title 子孙节点
```

由此可见，一个XPath语言的句子就是一个由符号'/'分割的若干个查询步组成的路径表达式，表示针对上下文节点的一个查询请求。在每个查询步中可以指定轴操作、节点检测和谓词。例如，节点检测node()和text()分别用于一般节点和文本节点的识别，而标记本身也是一种节点检测。首字符不是'/'的路径式，是相对路径，表示针对上下文节点的查询。表1.2说明了一组路径表达式所表示的查询请求。

表1.2 XPath路径表达式的案例

路径表达式	查询请求的目标节点
*/para	上下文节点的所有标记为para的子孙节点
chapter[5]/text()	上下文节点的第五个chapter子节点的文本子节点
//olist/item[2]	同一XML文档中所有olist节点的第二个item子节点
../@lang	上下文节点的双亲节点的lang属性
para[@type= 'warning']	上下文节点的子节点中属性为warning的para节点
chapter[title]	上下文节点的子节点中具有title子节点的chapter节点
para[last()]	选出上下文节点的最后一个para子节点

其中，对于任何标记 tag，tag 是 child::tag 的简写，//是/descendant-or-self::node()/的简写；.. 是 parent::node() 的简写；[2]是[position()＝2]的简写；@是 attribute::的简写。last()和 position()是 XPath 语言专用函数，分别表示当前节点的个数和索引值。XPath 函数库提供了 100 多个函数。

XPath 语言的路径表达式所表示的查询语义基本上遵循了按照轴操作来调整上下文节点，根据标记名、属性名等节点检测标志和谓词来筛选节点的原则。同时，XPath 1.0 采用节点集合作为数据模型，基本数据类型包括 XML 节点、数值、布尔值和字符串。所有轴操作、节点检测和谓词都作用于 XML 集合，针对集合中每个元素进行相应的查询和筛选。

2007 年发布的 XPath 2.0 版本是 XQuery 语言的子集。在这个版本中，采用 XDM 数据模型(XQuery 1.0 and XPath 2.0 Data Model)。这种数据模型以 XML 节点序列代替了 XML 节点集合，使得 XPath 语言给出的所有轴操作、节点检测和谓词都作用在 XML 节点序列上。在 XDM 模型中，即使是单个 XML 节点，也被看作是具有一个元素的节点序列。

XPath 2.0 语言除了路径表达式之外，扩展了多种表达式，包括支持循环处理 For 表达式、存在谓词和全称谓词的量化表达式、序列运算表达式、条件表达式。XPath 2.0 语言的另一重要特征是允许采用 XML Schema 来描述 XML 数据模式，并且提供一套类型描述方法，以及类型检查和类型转换的描述功能。

1.2.3 XQuery 1.0 语言

XQuery 语言是标准的 XML 查询语言。2007 年 1 月，国际万维网组织发布了 XQuery 1.0 版本[1]。XQuery 语言将 XPath 2.0 作为子集，以支持 XML 数据的提取，同时又提供了 XML 文档构造和函数式程序设计等功能。与 SQL 语言在关系数据库技术中的作用相似，XQuery 语言被认为是 XML 数据的 SQL。XQuery 语言是从 Quilt、XQL、XML-QL 和 OQL 等语言发展而来的，目前已经成为 XML 数据查询和数据处理的主流语言。所有知名数据库厂商的商业数据库产品已经支持 XQuery 语言的使用，如 IBM、Oracle 和 Microsoft 等公司。

鉴于 XML 数据已经成为互联网上的数据交换和数据共享事实上的标准，来自不同网络数据源的异构数据往往会包装成 XML 数据，各种数据处理经常包括 XML 数据查询、XML 数据加工，计算结果也需要组装成 XML 数据。XQuery 语言为 XML 数据处理的全过程提供了完整的开发语言，包括基于 XPath 语言的数据查询功能、基于函数式程序设计的数据加工功能和基于节点构造的 XML 文档构造功能。

XQuery 是一种模块化的语言。从程序结构来看，一个 XQuery 程序由以下三部分构成：