文章编号: 1671-8860(2006)10-0936-04　　文献标志码: A

一种原生 XML 空间索引及查询语言

余　亮[1]　边馥苓[1]

(1　武汉大学国际软件学院空间信息与数字工程研究中心, 武汉市珞喻路 129 号, 430079)

摘　要: 提出了一种用于原生 XML 数据的空间索引方法, 有效解决了在处理原生 XML 空间数据时遇到的效率问题, 并在此基础上构建了结合 XML 和关系数据库特性的 XML 空间查询语言 XML-GSQL。该查询语言具有良好的结构, 并且针对空间操作进行了扩展, 使空间操作更加简洁和高效。

关键词: 原生 XML; 空间索引; R^+ 树; XML-GSQL; XQuery

中图法分类号: P208

如何快速地从海量 XML 数据中抽取信息是 XML 研究的一个热点, 之前的大部分研究是基于现有的关系数据库平台的, 通过模式转换将 XML 模式或者 DTD 映射到关系数据库中, 产生了一系列从 XML 空间数据到关系数据库的模式匹配算法[1]。原生 XML[2] 的核心思想就是在不改变 XML 的存储方式和结构的条件下, 通过各种索引和组织方式使之能够提供符合数据库标准的查询模式和效率。原生 XML 的提出引发了很多新的 XML 概念, 如 XML DB[3]、XQuery[4] 等。本文基于原生 XML 的概念, 结合 XML 对空间数据的表达, 提出了一种基于 XML 和倒排表的空间索引。

1　基于 XML 的 R^+ 树空间索引

空间索引[5] 对于所有地理空间数据都是至关重要的, 和普通索引不同的是, 空间索引不仅包含特征的某些属性值, 而且包含特征的空间位置。对空间数据库的研究产生了大量的空间索引方法, 其综合性能比较突出的是 R^* 树、R^+ 树、Hilbert R 树、PMR 四叉树。本文建立的空间索引是基于 R^+ 树的, 因为它具有半结构化的特征, 结点广度和深度都不固定, 非常适合用 XML 文件来表示。

R^+ 树采取重叠区域技术进行空间存取, 它是一棵高度均衡的多叉树, 所有的叶结点处于树的同一层。树中的非叶结点又称为内部结点或目录结点, 包含形如(ref, rect)的若干数据项, 其中, ref 是指向某个子结点的指针, rect 是此子结点中包含的所有数据项的 MBR(最小外接矩形)。MBR 是所有空间索引的基础, 进行快速的初级过滤, 过滤的结果再进行空间运算以得到最终的结果, R^+ 树的每个结点正好和一个磁盘块对应。因此, 结点中能容纳数据项的最大数目是确定的。假设这个值为 B, 图 1 是一个 $B=3$ 的 R^+ 树示意图。由于 R^+ 树的深度和广度都不固定, 所以无法用 DTD 或者 Schema 来准确定义。图 1 并不代表标准的 XML 模式, 虚线部分代表一个或者多个 Index 嵌套。

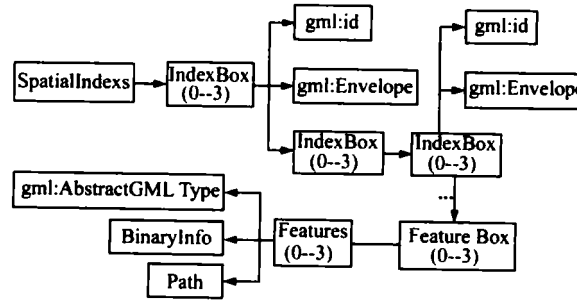

图 1　基于 XML 的 R^+ 树空间索引结构示意图

Fig. 1　R^+ Tree Spatial Index Structure Based on XML

为了加快空间运算速度, 可以将特征的空间信息以二进制的形式记录在索引文件中, 见图 1 中的 BinaryInfo 节点。该方法有两个优点: ① 在进行路径连接查询之前, 就可以进行精确过滤, 减少连接

收稿日期: 2006-07-29。

项目来源: 国家测绘局测绘科技发展基金资助项目(14601402024-04-04)。

次数, 提高整个查询效率; ② 二进制的数据存取速度比文本速度快得多, 而且二进制数据按照存储方式可以直接映射到内存对象。文本需要解析之后转化为内存对象。在这点上, 二进制数据的综合效率远高于文本。但是该方法中索引文件占用的空间较大, 是一种典型的空间换时间的方法。文献[6]详细讲述了如何将二进制数据嵌入 XML 文档。

2　原生 XML 数据查询语言及空间扩展

原生 XML 数据库需要提供一种抽象的查询描述语言, 能够有良好的结构, 并且具有数据库的特征。文献[7]提出了一种半结构化数据的查询方法, 但不支持空间特性。本文提出一种新的查询语言 XML-GSQL, 该语言以 SQL 为基础, 针对原生 XML 的特点进行了扩展。该语言具有以下几个主要特点: ① 扩展 SQL 语法。支持 SELECT、FROM、WHERE、DISTINCT、ORDER BY 等语法, 同时针对空间特性和空间关系进行扩展, 使之能适应空间查询。② 支持 XML 路径表达式。为了方便理解, 使用类似面向对象的表示法, 用点操作符“·”表示层次关系。③ 支持空间对象。支持如 Point、LineString、Polygon、Envelope 等空间对象, 并且支持相应的方法, 如长度 Length、多边形面积 Area。④ 查询结果至少可以以两种方式返回: 纯文本字段和 XML 文档。⑤ 支持空间运算符和空间关系符。类似于关系数据库提供的函数, 查询语言也应该支持相应的空间函数。

基于上文提到的索引结构和查询语言, 整个 XML 空间查询过滤过程可以分为以下几个步骤。图 2 表达了整个过程的流程和各个模块之间的关系。

1) XML-GSQL 查询语言语义分析。对查询语句进行语义分析, 提取出其中的查询谓词(SELECT, FROM, Order, Sum, 包括空间操作谓词)、条件子句(WHERE)、查询对象(XML 文件, XML 元素, XML 字段和属性等)、空间运算函数(Crosses, ReturnDistance, …)。分析结果可以表示为 $R(C) = V(O, W, F)$, 其中, R 表示查询结果, C 表示查询语言, V 表示谓词, O 代表所操作的对象, W 代表条件, F 代表空间操作。

2) 空间、非空间条件分解及过滤方案选择。分离查询条件中的空间和非空间条件, 有三种分解方法: ① $R(C) = R(P) \cap R(S) = Vp(O, Wp) \cap Vs(O, Ws, F)$, 其中, $R(P)$ 表示属性查询结果, $R(S)$ 表示空间查询结果, Wp 为属性查询条件, Ws 为空间查询条件。该公式表示分别对空间和非空间进行查询, 并将查询结果进行连接。② $R(C) = Vs(R(P)) = Vs(Vp(O, Wp), Ws, F)$, 该公式表示首先对属性进行过滤, 在属性过滤结果中再进行空间过滤。③ $R(C) = Vp(R(S)) = Vp(Vs(O, Ws, F), Wp)$, 表示首先进行空间过滤, 在空间过滤的基础上再进行属性过滤。

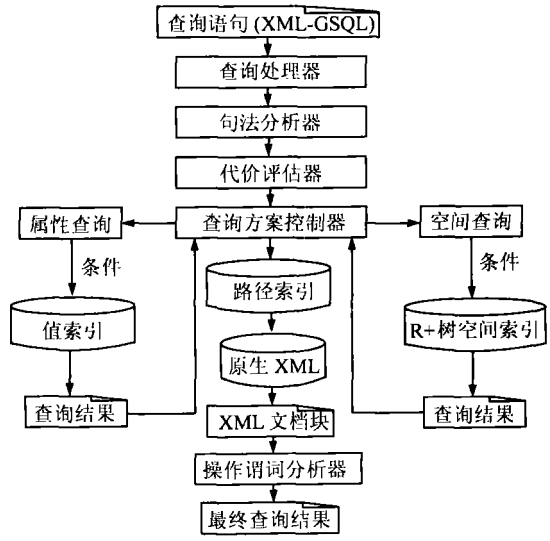

图 2　XML-GSQL 查询过滤过程

Fig. 2　Flow Chart of Spatial and Property Query

3) 过滤查询条件。根据步骤 2) 中的过滤方案, 分别进行空间查询或者属性查询。空间查询依赖于 XML 空间索引文件, 属性查询依赖于值索引, 查询结果以元素路径的形式返回。

4) 路径连接及结果输出。根据步骤 3) 中的查询结果, 利用路径索引在 XML 文件中查询到相应的元素, 根据用户传入的操作谓词和对象(SELECT 和 FROM 之间的子句), 返回最终查询结果。

根据以上设计, 以道路数据为例, 说明使用该查询语言进行常用的空间查询语法。

2.1　图文互访

首先分析从属性查询空间信息, 在实现时用到了值索引。假设要查询一条道路名称为“318 国道”的道路, 查询语法如下:

```
SELECT R.Road FROM [http://www.triside.com/samples/road.xml].Roads AS R _
WHERE R.Road.gml:id = '318国道'
```

该查询返回符合条件的 Road 结点 XML 文档, 包括其中的所有属性和空间数据。查询语言还支持模糊查询, 即 LIKE 操作符。如上述查询

条件需要查询所有国道,则 WHERE 子句修改为: WHERE R. Road. gml: id LIKE '%国道'。

2.2 点线面拓扑查询

拓扑关系按主体分为面-面、面-线、面-点、线-面、线-线、线-点、点-面、点-线。如需要查询一条道路经过的所有公共汽车站的信息,公共汽车站数据文件为 station. xml,汽车站以 Point 表示。假如站点离该道路的距离小于 20 m,则可以认为该车站在道路上。此类查询还用到了类似 SQL 中多表联合查询的多 XML 文件联合查询。省略上面的 URL 表示,实现语法如下:

```
@roadLine Polyline
SELECT @roadLine = newLineByString(221, 132, 422, 241, ….)
SELECT S.Staion FROM [road.xml].Roads AS R, [station.xml].Staions AS S WHERE _
    ReturnDistance(R.Road.CenterLine.LineString, S.Station.Point) <= 20
```

2.3 量算和统计查询

SQL 语法提供了用于汇总的聚合函数 SUM 和统计函数 COUNT,在 XML-GSQL 查询中,对聚合进行了扩展,使之能支持空间几何特征的聚合。下面对两类聚合举例进行说明。假设道路 XML 文档中将道路划分为道路段,每段由国道名称加上序号,形如"318 国道-001",那么一个可能的操作是求所有国道的总长度,用 XML-GSQL 表示如下:

```
SELECT SUM_LENGTH(R.Road.CenterLine) FROM [road.xml].Roads AS R _
    WHERE R.Road.gml:id LIKE "318 国道%"
```

对于一个森林分布的 GML 文件,以林斑为主体记录元素,林斑的基本属性有林斑号(gml:id)、所属乡村(Village)、林斑空间形状(gml:Polygon)等。需要统计某个乡村所有的林斑面积,用 XML-GSQL 表示如下:

```
SELECT SUM_AREA(T.Tree..gml:Polygon) FROM [Tree.xml].Trees AS T _
    WHERE T.Tree.Village = '东村'
```

2.4 空间分析

空间分析是 GIS 中的一个基本功能,其中缓冲区分析是空间分析中最常用的一种分析方法,就上述的林斑数据为例,假如在一场火灾之后,要统计查询被该火灾烧毁的总林斑面积,已经取得火灾的过火区图形,其查询过程如下:

```
@fireArea Polygon
SELECT @fireArea = newPolygonByLineString(101, 342 332, 442, …)
SELECT SUM_AREA(Intersect(T.Tree.gml:Polygon, @fireArea)) FROM _
    [Tree.xml].Trees AS T WHERE Crosses(T.Tree.gml:Polygon, @fireArea) = true
```

上述表达式中同样用到了面积聚合函数 SUM_AREA,所不同的是统计的量不再是单一的属性值,而是经过空间运算后的结果,查询条件也由简单的属性条件变化为空间关系。

3 实 验

实验系统使用 Java 语言开发,利用 ArcGIS Java API 进行空间运算和空间对象持久化,XML 解析器使用 Apache Xerces, XQuery API 使用 Saxon,分别对 20 M、50 M、80 M 和 120 M 的 XML 数据进行了测试,图 3 显示了空间查询结果。实验表明,路径索引和值索引对于大容量的 XML 数据优势更明显,XML-GSQL 能够提供完善的语义,较好地满足了现有的 GIS 应用需求。同时,本文还有许多问题值得进一步研究,如更优化的索引存储机制,索引维护代价的考虑,以及如何将该索引系统和现有的 XML DB 标准结合起来。

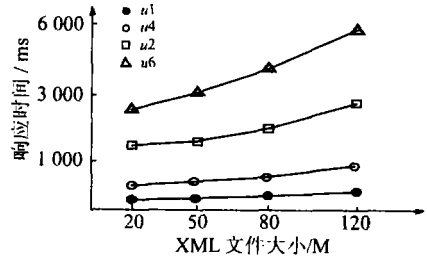

图 3 空间索引效率测试结果

Fig. 3 Result of Efficiency-Test of Spatial Index

参 考 文 献

[1] 关佶红, 虞为, 安扬, 等. GML 模式匹配算法[J]. 武汉大学学报·信息科学版, 2004, 29(2), 169-174

[2] Kimbro S. Introduction to Native XML Databases [OL]. http://www.xml.com/pub/a/2001/10/31/nativexmldb.html, 2001

[3] Tamino H. Schoning-A DBMS Designed for XML [C]. The 17th International Conference on Data Engineering, Washington D C, USA, 2001

[4] Robie J, Lapp J, Schach D. XML Query Language (XQL)[C]. The Query Language Workshop, Boston, Massachusetts, 1998

[5] 阎超德, 赵学胜. GIS 空间索引方法述评[J]. 地理

与地理信息科学, 2004, 20(4): 26-39
[6] 徐德智, 何芳, 吴敏. 二进制数据的 XML 集成方法研究与实现[J]. 计算机应用研究, 2004(9): 37-39
[7] Abiteboul S, Quass D, McHugh J, et al. The Lorel Query Language for Semistructured Data[J]. Journal of Digital Libraries, 1996, 1(1): 68-88

第一作者简介: 余亮, 博士生。现主要从事 GIS 应用、人工智能研究。
E-mail: yul26@163.com

A Spatial Index and Query Language Based on Native XML

YU Liang[1]　*BIAN Fuling*[1]
(1 Research Center of Spatial Information and Digital Engineering, International School of Software, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

Abstract: A method for constructing spatial index over the native XML is proposed. It also includes the path-index and the value-index, which are the base of spatial index. A new query language is introduced as XML-GSQL, which is designed to suit native XML and characterized by features of both XML document and database. And an examples of this language is given to evaluate the total efficiency of spatial query.

Key words: native XML; spatial index; R^+ tree; XML-GSQL; XQuery

About the first author: YU Liang, Ph. D candidate, majors in application of GIS and artificial intelligence.
E-mail: yul26@163.com