

## GNETS: 一种新的 GML 数据存储和查询机制

张 龙, 关侗红, 王炜立, 钱志萍

(同济大学 计算机科学与技术系, 上海 201804)

**摘 要:** 随着 GML 的发展和 WebGIS 环境下地理空间信息的广泛使用, 越来越多的地理信息数据以 GML 格式表示. 然而, 以文本形式存储的 GML 数据给数据管理和查询带来不便. 根据 GML 文档的特点, 提出一种新的基于节点关系的 GML 数据存储和查询机制, 将 GML 数据存储在对象关系数据库中, 并根据用户提供的搜索关键词, 利用 GML 数据在对象关系数据库中的存储模式和节点关系, 对 GML 数据进行相关查询, 最终将查询结果以 SVG 格式显示给用户. 实验结果表明, 提出的查询方法可行有效.

**关键词:** GML; 对象关系数据库; 数据存储; GML 查询

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1000-7180(2011)08-0022-04

## A New Mechanism for GML Data Storing and Querying

ZHANG Long, GUAN Ji-hong, WANG Wei-li, QIAN Zhi-ping

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

**Abstract:** With the GML's development and widespread use of geospatial information under WebGIS environment, more and more geographical data expresses in GML format. However, it brings inconvenient for GML data storing and querying. In this paper, based on the GML document features, we propose a new mechanism for GML data storing and querying, which stores GML data in the object-relational database, then does queries corresponding to the key words that users provide for searching according to the storing patterns and nodes relation of GML data in the object-relational database, and last return the searching result as the SVG format to users. Experimental result shows that the method brought forward in this paper is feasible and efficient.

**Key words:** GML; the Object-Relational Database; data storing; GML querying.

### 1 引言

随着 Web 技术的广泛应用和 Internet 的迅猛发展, GIS (Geographic Information System) 已经在各个应用领域取得了惊人的成就. 而现有的 GIS 大多采用不同的数据结构和模型, 不易实现 GIS 集成, 以利于空间信息的共享和互操作. 1999 年开放式地理信息系统协会 (OGC)<sup>[1]</sup> 提出了地理标记语言 GML (Geography Markup Language)<sup>[2]</sup>, 为 GIS 空间数据建模、集成与共享提供了统一的标准和框

架. GML 是 XML (eXtensible Markup Language)<sup>[3]</sup> 的一种特殊表现形式, 用于地理空间信息的编码、传输和存储. 近年来, 随着大众化、分布式、网络化的海量信息 GIS 系统的发展, 越来越多的地理信息以 GML 格式表示, 同时也对如何有效地存储和查询 GML 数据提出了新的挑战. 一些学者<sup>[4-7]</sup> 提出利用传统数据库来存储和查询 GML 数据. 由于关系数据库不能很好地支持空间数据对象的存储, 本文采用了对象-关系数据库作为后台存储数据库.

由于任何一个 GML 文档都可以表示成一棵文

收稿日期: 2011-05-12; 修回日期: 2011-06-15

基金项目: 国家自然科学基金项目 (60873040); 国家“八六三”计划项目 (2009AA01Z13); 新世纪优秀人才支持计划 (NCET-06-0376); 上海市“曙光计划”项目 (09SG23)

档树,基于这种文档特性,本文提出了基于节点关系的 GML 数据存储和查询机制 GNETS (GML-Node-Edge-Text-Spatial),即将 GML 数据以节点信息、边(节点关系)信息、文本信息和空间信息进行数据对象划分,存储在对象关系数据库中,通过查询关键词,自动生成查询语句并解析,将查询语句的查询信息保存在自定义的数据对象中,以便 GML 数据管理.利用 GML 数据在对象关系数据库中的存储模式和节点关系,进行 GML 数据查询,通过构造算法将查询结果还原成 GML 片段,并以 SVG<sup>[8]</sup> 格式显示给用户.

## 2 相关工作

对于 GML 数据存储,Leonardo 和 Eliane<sup>[4]</sup>提出了一种把 GML 数据存储到关系数据库的机制,但存在存储粒度太小和不支持空间操作等问题. Corcoles 和 Gonzalez<sup>[5]</sup>选择了三个基于关系数据库的存储模型 LegoDB、Monet、Xparent 来存储 GML 数据,但不支持空间操作. Jeung 和 Park<sup>[6]</sup>, Li 和 Zhou<sup>[7]</sup>提出了通过扩展 PostgreSQL/SPE、Oracle Spatial 等空间数据库来存储 GML 数据,支持空间、非空间和两者混合的查询,但没有对查询结果数据集进行 GML 文档重新构建.

对于 GML 数据查询,主要应用 XML 查询的研究成果. XML 的研究目前主要集中在 XML 标准查询语言 Xquery<sup>[9]</sup>到数据库查询语言 SQL 的查询转换研究. Fernandez 等<sup>[10]</sup>提出的 SilkRoute 方法,以及 David Dehaan 等<sup>[11]</sup>提出的基于动态区间编码的转换方法等都是将 XQuery 查询转化为 SQL 查询. 孙宏等<sup>[12]</sup>提出了模式映射弱依赖的 XQuery 到 SQL 的转化算法. 曹莉等<sup>[13]</sup>提出了一种基于模式的 XML 查询及优化方法.

然而 GML 文档多层嵌套的文档特点不仅给 GML 数据在数据库中存储造成了极大的数据冗余和空间浪费,极大地影响了查询效率,而且不利于查询结果的重构. 本文针对上述问题,提出了一种新的 GML 数据存储和查询机制 GNETS,较大程度地减少了 GML 数据在对象关系数据库中的存储空间和数据冗余,并给出了一种高效的数据查询方法和查询结果重构算法.

## 3 GNETS 数据存储

本文通过对 GML 文档树中的节点类型划分将不同类型的节点数据保存在对应的数据模式中.

图 1 为 GML 数据存储模式.

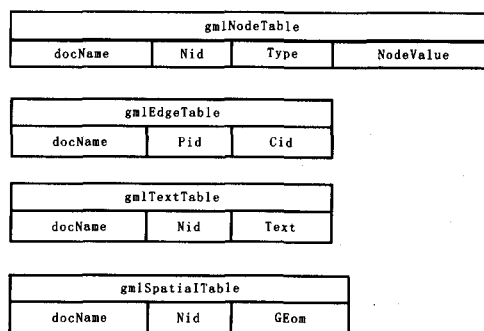


图 1 GML 数据存储模式

本文创建四种数据模式: gmlNodeTable、gmlEdgeTable、gmlTextTable 和 gmlSpatialTable,对应于数据库的四张表,用于保存所有 GML 文档数据,分别用 V、E、T、S 代表. v、e、t、s 分别对应四张表中的一条记录. gmlNodeTable 保存文档树中非叶节点的信息,其中 docName 保存 GML 文档的名称,其他三个模式相同, Nid 保存节点在文档树中的编号,文档树中的节点以先序遍历的方式进行编号, Type 用于保存节点的数据类型,节点类型有四种 E、A、T 和 S. E 表示子节点中包含元素节点的节点, A 表示属性节点, T 表示文本节点, S 表示几何节点. NodeValue 保存节点的名称; gmlEdgeTable 保存文档树中节点之间的父子关系, Pid 保存父节点的 Nid, Cid 保存子节点的 Nid; gmlTextTable 保存文档树叶节点的属性值或文本值; gmlSpatialTable 保存叶子节点中所有空间节点的几何信息(GEOM).

## 4 GNETS 数据查询

在 GNETS 数据存储的基础上,本文提出了 GNETS 数据查询,查询过程如图 2 所示. GNETS 查询处理主要包括: GQuery 语句解析、条件节点(CNodesList)查询、目标节点(RNodesList)查询、GML 数据重构.

### 4.1 GQuery 语句解析

本文通过解析 GQuery 语句,将查询信息保存在自定义的 GQLInfo(GML Query Information)对象中. GQLInfo 对象是由一个或多个 FWRInfo(FOR-WHERE\_RETURN Information)对象组成, FWRInfo 对象的数量由 GQuery 语句中涉及到的文档数量决定. FWRInfo 对象描述单个文档的查询信息,包括文档名称、路径表达式、查询条件、关系表和返回值. 因为在多文档查询过程中,某一文档的查询会以其他文档的查询结果作为条件,所以我们用

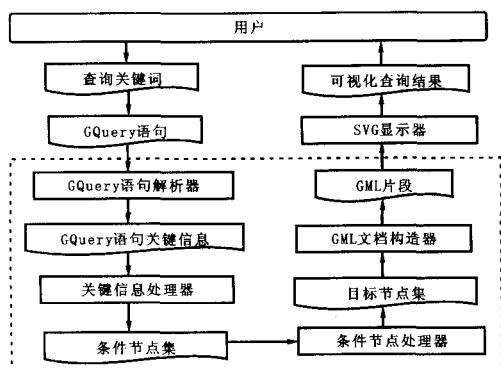


图2 GNETS数据查询过程图

关系表存储文档之间的查询关系。本文仅以 FLWOR(FOR-LET-WHERE-ORDER BY-RETURN)语句为例,建立 GQuery 语句的关键信息 GQLInfo 对象。

#### 4.2 条件节点及目标节点查询

当得到关键信息 GQLInfo 后,根据条件节点查询器,查询满足条件的条件节点集。

对于单文档查询,在 GQLInfo 中只有 FWRInfo1。首先通过 FWRInfo1 的条件语句,获取查询文档的名称 SDocName,并根据条件语句获取条件节点的名称 CName,在节点表(N)中查询所有名称为 CName 并属于 SDocName 的节点,并根据边表(E)查询其属性值、文本值或者空间信息,并根据条件进行判断,如果满足,添加到条件节点集 CNodesList 中。

当 GQuery 语句为多文档条件查询时,涉及到多个 FWRInfo 对象。首先查找 FWRInfo 对象关系表为空的 FWRInfo 对象,并按照单文档进行处理,将得到的 CNodesList 作为临时条件节点集,将该 FWRInfo 对象设为已处理,然后在剩余的 FWRInfo 对象中查找那些关系表中涉及到的 FWRInfo 对象已经全部处理的对象,并将上一次临时条件节点集作为条件进行查询,依次循环,直到所有 FWRInfo 都已处理,返回最终的 CNodesList。

当得到条件节点集后,根据条件节点集进行目标节点集的查询。因为条件节点是目标节点的直接或者间接子节点,所以为了得到目标节点集,我们只需要遍历条件节点集,查找在查询文档中满足 FWRInfo1 查询路径的节点,保存在 RNodesList 中。

#### 4.3 GML 数据重构

当得到 RNodesList 后,就需要构造目标节点集 RNodesList 的 GML 片段。其算法如图 3 所示。

首先根据 FWRInfo1 的路径表达式,构造树的主杆,并对树主杆上的节点添加属性。然后将每一个

目标节点生成的子树(算法如图 4 所示)添加到主杆最后一个节点上。该树即为满足 GQuery 语句的 GML 片段。

Algorithm 1 Get the result Tree according to the RNodesList

Input: RNodesList and the FWRInfo1 of GQLInfo

Output: the result Tree

1. Create a Tree = null, ParentNode = Tree;
2. Let the ChildrenTreeList = {};
3. Let PnodeList = pathExpression.split("/");
4. For i: 0 to PnodeList.Length-2
5. Create Node v named PnodeList[i];
6. ParentNode.appendChild(v)
7. ParentNode = v;
8. End for
9. For each v in RNodesList
10. Create node TPNode named v.NodeValue;
11. ParentNode.append(TPNode);
12. CreatechildTree(TPNode, v.Nid);
13. End for each v
14. Return the result Tree;

图3 GML 数据重构算法

Algorithm 2 Create the child tree of parent node

Input: Parent Node; parentNode and Parent Node id; Id

Output: the child Tree of parent node

1. CreatechildTree(Element parentNode, int Id)
2. If (exists e. Pid = parentid && e ∈ E)
3. add all e. Cid in ChildrenIDList;
4. End if
5. For each Id in ChildrenIDList
6. If (v. Nid = Id && v ∈ V)
7. Switch (v. Type)
8. Case 'A': add node as attribute node to parent node
9. Case 'T': add node as text node to parent node
10. Case 'G': add node as geometry node to parent node
11. Case 'E':
12. Create node ev named v. NodeValue;
13. parentNode.append(ev);
14. CreatechildTree(ev, v. Nid);
15. End if
16. End for each Id

图4 目标节点子树生成算法

#### 4.4 GNETS 查询优化

对于 GML 数据的查询,GNETS 查询算法需要不断的将判断条件转换成对应的 SQL 语句,进行 SQL 查询,极大影响了查询效率。所以当文档不是很大的时候,可以将涉及到同一文档的四张表的内容一次性从数据库中读取出来存储于内存中,减少了访问数据库的次数,大大降低了查询所消耗的时间。但当文档比较大的时候,这种做法不可取。

## 5 实验及分析

对于实验,我们在 JAVA 平台上做了一个 GIS 原型系统,该系统中数据存储和数据查询部分采用 GNETS 机制。

### 5.1 实验数据

本文所用的实验数据如表 1 所示. 该组数据用于描述墨西哥的所有城市、河流以及州的信息. 其中文档 1、2、3 来自 ArcInfo<sup>[14]</sup>, 文档 4、5 是在 2、3 文档基础上人工合成的数据. 文档的大小从 24 kB 到 3 080 kB 不等,各文档的节点数统计如下。

表 1 实验数据

| 编号 | 文件名称              | 大小    | 节点数/个 | 几何体数/个 |
|----|-------------------|-------|-------|--------|
| 1  | MEX_CITIES.xml    | 24    | 408   | 36     |
| 2  | MEX_RIVERS.xml    | 120   | 274   | 30     |
| 3  | MEX-STATES.xml    | 308   | 528   | 33     |
| 4  | Modify_RIVERS.xml | 604   | 1 241 | 178    |
| 5  | Modify_STATES.xml | 1 589 | 2 647 | 163    |

### 5.2 性能分析

本文设计的实验分别对基于文档<sup>[15]</sup>、基于 Native XML、基于 GNETS 以及基于 GNETS \* (按照 4.4 节中优化后的 GNETS) 的存储方法进行实验对比. 基于 Native XML<sup>[16]</sup> 的存储方案是将 GML 整个文档存储在 DB2 Pure XML 数据库. 实验 1 分析比较不同查询方法在相同条件下查询所消耗的时间, 实验 2 分析比较不同文档尺寸下不同查询方法所消耗的时间。

实验 1 结果如图 5 所示, 基于文本存储的所消耗的时间最多; Native XML 方法在查询时不需要遍历整个文档而加快查询, 所以较快; 对于 GNETS 方法, 由于事先将 GML 文档存储在对象关系数据库中, 在查询过程中只需要去查询与条件表达式或者查询路径相关的节点, 不需要对所有节点进行判断, 提升了查询效率. GNETS \* 方法将所有数据载入内存, 不涉及数据库和 GQuery 到 SQL 的转化, 所以查询效率最高。

实验 2 结果如图 6 所示. 随着文档的增大, 所有查询方法的查询时间都在增长. 其中文本查询方法时间消耗增长最快, GNETS 和 GNETS \* 最慢. 这是因为 GNETS 和 GNETS \* 是以固定的数据模式存储在对象关系数据库中, 其查询效率不受文档嵌套层次的影响. 当文档嵌套层次越多, GNETS 和

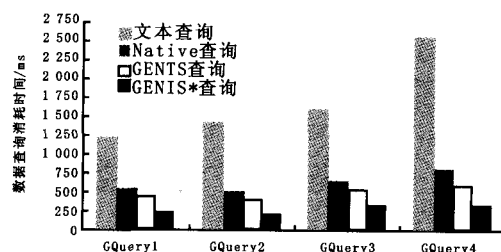


图 5 不同 GQuery 语句和存储方法下查询时间对比

GNETS \* 查询效率就越高。

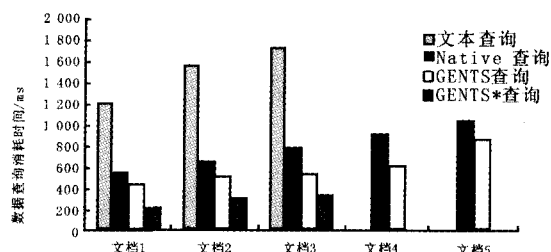


图 6 不同文档大小下查询时间对比

## 6 结束语

本文提出一种 GML 数据存储和查询的新机制 GNETS, 该机制通过对 GML 数据的划分, 将 GML 数据存储在对对象关系数据库中, 然后通过 GML 数据查询器依次查询条件节点和目标节点, 通过 GML 数据重构算法, 生成 GML 片段, 以 SVG 的格式显示给用户. 实验证明, 本文提出的 GML 数据存储和查询机制 GNETS 能高效地存储、查询 GML 文档数据。

### 参考文献:

- [1] Open GIS Consortium - OGC [EB/OL]. [2011-6-21]. <http://www.opengis.org>.
- [2] Geography Markup Language (GML) 2.0 [EB/OL]. [2011-6-21]. <http://www.opengis.net>.
- [3] Extensible Markup Language (XML) 1.0 [EB/OL]. [2011-6-21]. <http://www.w3.org/TR/REC-xml/>.
- [4] Reinehr L, Dias E. A Mechanism to Import GML Data into a Relational Database [EB/OL]. [2011-6-20]. <http://www.venuewest.com/gmldev03/absbios/mechimp.html>.
- [5] Corcoles J E, Gonzalez P. Analysis of different approaches for storing GML documents [C]//Proceeding of the Tenth ACM International Symposium on Advances in Geographic Information Systems. New York: ACM, 2002:11-16.

- [7] Martin J. Mohlenkamp. A fast transform for spherical harmonics[J]. The Journal of Fourier Analysis and Applications, 1999, 2(2/3): 159—184.
- [8] Blais J A R, Provins D A, Soofi M A. Optimization of spherical harmonic transform computations[J]. Computational Science, 2005( 3514): 393—401.
- [9] Marcia A Inda, Rob H Bisseling, David K Maslen. On the efficient parallel computation of legendre transforms [J]. SIAM J SCI Comput, 2001, 23(1): 271—303.
- [10] Xiao Huadong, Lu Yang. Parallel computation for spherical harmonic synthesis and analysis[J]. Computers & Geosciences, 2007 (33): 311—317.
- [11] Matthew G Reuter, Mark A Ratner, Tamar Seideman. A fast method for solving both time—dependent Schrodinger equation in angular coordinates and its associated “m—mixing” problem[J]. The Journal of Chemical Physics, 2009(131): 94-108.
- [12] 赵廷刚, 张刚. 谱方法的 Chebyshev—Legendre 变换 [J]. 甘肃高师学报, 2006, 11(2): 1—3.
- [13] B Alpert, V Rokhlin. A fast algorithm for the evaluation of Legendre expansions[J]. SIAM J Sci Statist Comput, 1991(12): 158—179.
- [14] S Chandrasekaran, M Gu. A divide—and—conquer algorithm for the eigendecomposition of symmetric block—diagonal plus semiseparable matrices [J]. Numer Math, 2004, 96: 723—731.
- [15] Healy D M, Kostelec P J, Rockmore D. Towards safe and effective high—order Legendre transforms with applications to FFTs for the 2—sphere[J]. Adv Comput Math, 2004(21): 59—105.
- [16] Press W H, Teukolsky S A, Vetterling W T, et al. Numerical recipes in fortran 77[M]. 2nd. Cambridge, UK: Cambridge University Press, 1992.
- [17] 王鹏, 吕爽, 聂治, 等. 并行计算应用及实战[M]. 北京: 机械工业出版社, 2009.
- [18] Calvin Lin, Lawrence Snyder. Principles of parallel programming[M]. USA: Pearson Education, 2009.

### 作者简介:

王翔男, (1986—), 硕士研究生, 研究方向为大规模科学与工程计算研究.

宋君强男, (1962—), 博士生导师, 研究员, 研究方向为高性能计算和数值天气预报研究.

卢风顺男, (1982—), 博士研究生, 研究方向为通用 GPU 并行计算研究.

杨锦辉男, (1986—), 博士研究生, 研究方向为云计算研究.

(上接第 25 页)

- [6] Jeung H, Park S. A GML data storage method for spatial databases [J]. Journal of GIS Association of Korea, 2004, 12(4): 307—319.
- [7] Li Y, Zhou S, Li J. GML Storage: A Spatial Database Approach [C]// Proceeding of ER'04 Workshops 2004. Berlin: Springer, 2004: 55—66.
- [8] W3C XQuery 1.0 and XPath 2.0 formal semantics[EB/OL]. [2011—6—21]. <http://www.w3.org/TR/2005/WD-xquery-semantics-20050211>
- [9] Fernandez M F, Kadiyska Y, Morishima A, et al. SilkRout: A framework for publishing relational data in XML[J]. ACM TODS, 2002, 27(4): 438—493
- [10] Dehaan D, Toman D, Consens, M P et al. A comprehensive XQuery to SQL translation using dynamic Interval encoding[C]// Proceedings of the ACM SIGMOD international conference on management of data, New York, NY, USA: ACM, 2003, 623—634
- [11] Sun H W, Zhang S S, Zhou J T. XQuery—to—SQL translating algorithm with little on schema mapping between XML and RBB. [C]// proceedings of the 8th International conference on Computer Supported Cooperative Work in Design[s. l.]: IEEE, 2004.
- [12] 曹莉, 李新宇, 许钟. XML 数据查询及优化方法研究 [J]. 微电子学与计算机, 2006, 23(8): 125—128.
- [13] W3C. SVG1. 1 Specification. [EB/OL]. [2011—6—21]. <http://www.w3.org/TR/2004/WD-SVG11-200215/>. 2002.
- [14] ArcInfo. [EB/OL]. [2011—6—21]. <http://www.esri.com/software/arcgis/arcinfo/index.html>
- [15] GML 文档[EB/OL]. [2011—6—21]. <http://schemas.opengis.net/sampling/1.0.0/Examples/>
- [16] Nicola M, Linden B. Native XML Support in DB2 Universal Database [C]// Proceedings of the 31st international conference on very large data bases, 2005. Norway: IEEE, 2005: 1164—1174

### 作者简介:

张龙男, (1968—), 硕士, 研究方向为空间数据库管理.