

# 基于实体对象树 FOOT 模型和四层体系结构的 GIS 方法研究

刘仁义 刘南 谢炯 冯杭建 李伟 严志民  
何江 俞剑君 夏凯 阮淼钗

(浙江大学 GIS 重点实验室, 杭州 310028)

**摘要** 提出了面向地理空间实体对象树模型(FOOT),首先重点讨论了空间分区、空间聚簇、空间索引,然后分析了四层结构设计中的数据库服务层、应用接口层、应用层相关技术。基于所提出的空间数据组织方法和系统结构,介绍了具有海量空间数据管理的系统功能模块,并提供了详细的对象逻辑关系说明。同时在对 FOOT 模型和系统实现的关键技术进行设计的基础上,给出了一个基于 Linux 操作系统和 Oracle 数据库的分布式海量空间数据处理平台实例。采用 30 GB 的空间数据测试结果表明,系统实现了面向实体的空间数据组织、统一的海量空间数据管理以及分布式空间数据计算等主要目标。该系统用于对全国 1:400 万土地利用图、1:400 万市县行政图数据上载的试验表明,不仅显示速度较快,且图形裁剪、空间叠加、缓冲分析等效率较高,验证了 FOOT 模型的正确性、有效性、合理性。

**关键词** 实体对象树模型(FOOT) 海量空间数据 Linux Oracle

中图分类号: P208 文献标识码: A 文章编号: 1006-8961(2004)09-1135-10

## The Method of GIS Platform Based on Feature-oriented Object Tree(FOOT) and Four Tier Architecture

LIU Ren-yi, LIU Nan, XIE Jiong, FENG Hang-jian, LI Wei, YAN Zhi-min,  
HE Jiang, YU Jian-jun, XIA Kai, RUAN Miao-qian

(Key Laboratory of GIS, Zhejiang University, Hangzhou 310028)

**Abstract** A geo-spatial data model of feature-oriented object tree (FOOT) is presented. Spatial partitioning, spatial clustering and spatial indexing were emphatically discussed. Database services, application interface and application services in Four Tier Architecture are analyzed. By the spatial data organization and systematic architecture, function modules with massive spatial database management for the system are introduced. And also detailed object logic relation for the modules were explained. At last, on the base of design for the FOOT model and key technique of systematic implementation, an example of massive spatial database platform running in Linux and Oracle was given. The testing with 30 GB spatial data shows that the systematic main aims implement in feature-object spatial data organizing, integrative massive spatial data management and distributed spatial data computing. By the uploading national land use map and national county-administration map of 1:400, the display is rapid. the efficiency of map windowing, spatial overlaying and buffer analyzing are very high, which validates the correctness, validity and rationality of FOOT.

**Keywords** feature-oriented object tree(FOOT), massive spatial data, Linux, Oracle

## 1 引言

为了实现下一代(或称第四代)GIS平台提出的面向网络海量空间信息的高效管理及适应分布式存储与处理的要求,在国家863项目支持下,采用直接“面向实体对象树”模型(feature-oriented object tree, FOOT)的数据组织方法,对下一代GIS平台的若干核心技术进行了研究和试验,并开发出了一个基于Linux和Oracle的海量空间数据管理和多用户协同处理的分布式GIS平台原型。

## 2 基于实体对象树结构的空間数据组织

ArcInfo、MGE、GeoMedia、MapInfo及国产GIS软件平台在系统体系结构(C/S及B/S)、空间数据组织方式、开放性(组件式开发方法)、空间与属性的统一管理等方面已有很大改进,但仍存在许多共性问题无法解决<sup>[1,2]</sup>,而基于图层的空间数据组织模型是导致诸多其他问题产生的内在基本因素,如:以二维的地图图层作为地理空间处理基础,使得海量空间数据空间分析困难;空间数据比例尺粒度单一,无级数据表达难以调控;矢量栅格一体化操作受到制约;多用户环境下多图层空间数据同步处理机制无法实现。

### 2.1 几种常见的GIS空间数据模型及缺陷

地理空间数据模型,即CAD(computer aided design)数据模型、Coverage数据模型<sup>[3,4]</sup>以及近年来出现的以Geodatabase为代表的面向对象的空间数据模型。

#### 2.1.1 CAD数据模型

CAD数据模型起源于20世纪60~70年代,它是通过点、线、面将地理数据存贮在二进制文件中,由于其不仅缺乏属性信息<sup>[1]</sup>,且空间数据不是存储在数据库中,因此给GIS软件开发和GIS数据共享都带来了困难。

#### 2.1.2 Coverage数据模型

该模型源于1981年ESRI公司GIS软件——ArcInfo<sup>[3]</sup>。此模型比CAD数据模型有了较大的改进,其不仅支持属性数据,并可将空间数据与属性数据有机地结合在一起,同时存储了矢量数据的拓扑关系。Coverage数据模型不仅提高了地理空间信息的表达能力和数据的分析能力,也提高了数据录入

的准确性,由于该模型将空间数据和属性数据分开存储,仅用点、线、多边形等几何元素描述空间实体,因而缺乏对空间实体的显式定义及其基础关系的描述。由于Coverage数据模型是基于文件方式来管理空间数据,从而导致了—些其他问题,如:

(1)数据冗余度大。由于数据重复存储,且基本上与各自的应用程序相对应,致使数据不能以记录和数据项为单位共享,即使有部分数据相同,只要逻辑结构不同,那么用户也必须各自建立自己的文件。这不仅浪费存储空间,且不能统一修改,容易造成数据的不一致性。

(2)缺乏数据独立性。由于文件系统中的数据文件是为了满足特定业务的专门需要而设计的,因此只能服务于某一特定应用程序,且数据和程序相互依赖,如果改变数据的逻辑结构或文件的组织方法,则必须修改相应的应用程序;如果改用另一种程序设计语言来编写程序,也将影响数据文件的结构。

(3)缺乏集中管理。除了对记录的存取由文件系统承担以外,文件没有统一的管理机制,致使数据的维护任务仍然由应用程序来承担。

(4)缺乏安全性和完整性。由于其是用文件来存放空间数据,因此安全性难以得到保证;同时,由于空间数据的来源广泛,数据质量情况复杂,致使数据的完整性不能得到很好的保证。

(5)无法进行文件共享和网络操作。网络化的GIS和WebGIS是GIS发展的必然趋势,而用文件来存放空间数据,空间数据就不能够同时被多个用户共享,致使分布式计算和多用户并发操作不能实现。

#### 2.1.3 面向对象数据模型

随着面向对象的技术的成熟和广泛应用,以ArcInfo8中Geodatabase<sup>[1]</sup>为代表的面向对象的空间数据模型应运而生。这种面向对象的数据模型可以将物体的空间图形数据和属性数据集成在同一对象中处理,从而克服了地理关系数据模型的不足。由于面向对象的数据模型提供了更丰富的数据表达能力,因而能够更“自然”地表示客观世界,但是由于空间数据的复杂性及受当前技术水平的限制(如面向对象的数据库技术仍不成熟),其仍有不可避免的缺点,主要表现在以下几方面:

(1)没有突破图层的概念。由于当前面向对象数据模型仍以图层作为用户获取数据的主要手段,因此降低了数据请求的灵活性,例如,在海量空间数据库中,用户常常需要的不是整层的信息,而是层中的

部分信息;在时空数据库中,用户需要的某一时刻的信息可能是按照某种条件查到的空间对象的集合。可见这些应用要求用户获取数据的接口已不再是狭义的图层,而是满足某种条件的动态空间对象集;而以图层为条件来组织空间数据仅仅是动态对象集的一个实例<sup>[5]</sup>。

(2)当前的面向对象数据模型仅是一种逻辑模型,仅在代码级实现了面向对象。由于目前面向对象数据库技术只能将面向对象的空间实体存储于对象-关系型数据库中,且空间实体的存储需用中间件将其属性与规则分解后方能存储,而空间实体的组合也要通过中间件来处理,因此,面向对象数据模型仅是一种逻辑模型,它仅在代码级实现了面向对象。可见这种复杂、繁琐的分解与组合操作将降低系统的工作效率,同时也阻碍了空间实体规则的继承与派生。

(3)面向对象数据模型中,有关面状拓扑尚未实现,相关理论与技术尚需进一步研究。由于面状拓扑本身的复杂性,致使当前面向对象模型中虽然有关于面状拓扑的描述,但仅停留在理论上,直到 ArcInfor8.3 才提供了对面状拓扑的支持,因此,如何灵活、高效地描述面状拓扑关系,怎样实现面状拓扑的约束规则、实现拓扑关系的自动维护等都是值得进一步研究的问题。

## 2.2 面向实体对象树模型

空间数据模型与组织是第四代 GIS 软件平台所要解决的关键技术,由于传统的地理信息系统软件在数据组织方面采用图幅、工程、图层、几何图元的组织和存储模式<sup>[6]</sup>,而从面向对象的角度看,由于这种方法人为分割了地理对象在空间上的连续性,且缺乏对空间实体或现象的显式定义及其基础关系的描述和难以表达、处理复杂的空间实体,因而导致数据分析和处理复杂性提高,也无法实现空间数据面向问题处理、空间数据挖掘等深层次的应用。在本研究中,将 GIS 技术、数据库技术和空间认知理论相结合,通过分析现有空间数据建模利弊,设计了一套脱离图层概念、面向地理空间实体的数据模型,并从几何、语义、空间关系、属性和时态等多方面描述了地理现象和地理过程,体现了新一代 GIS 的设计思想和实现策略<sup>[7~9]</sup>。

面向地理空间实体对象树模型(FOOT)是将空间数据组织为面向实体具有分布式结构的数据对象集合,包括通用实体类(EntityClass)、要素类(FeatureClass,或称地物类、特征类)、特征数据集

(FeatureDataset);在特征数据集中还可包含实体类、要素类、拓扑类、关系类。一个实体类在空间数据库中为一个表,用于存储非空间数据;一个特征类是指具有相同集合类型与相同属性的要素的集合,其在数据库中对一个空间表,在本研究中,由于涉及的覆盖区域的地图一般为 GB~TB 级数据量,其对应的是一个包含数十万至数百万条记录的大表(地图基表);关系类用于定义两个不同的实体类或要素类之间的关联关系,例如,可以定义房主和房子之间的关系,房子和地块之间的关系等;拓扑类包括线性拓扑及面状拓扑信息,特别由于这种面状拓扑信息的支持,大大提高了诸如空间叠加、剪切等空间算子的效率。一个特征数据集是共享空间参考系统的特征类、关系类、拓扑类的集合,其数据组织模型如图 1 所示。

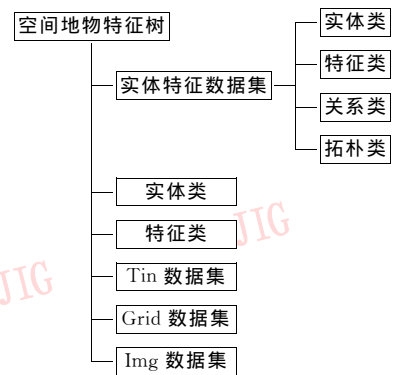


图 1 面向地理空间实体对象树数据组织

与传统的基于图层的空间数据模型相比,由于 FOOT 建模方法是一种在较高抽象层次的建模方法,因此具有较好的地理信息认知观,同时亦能很好地利用面向对象的原理与方法来定义和建立空间关系与非空间关系。FOOT 数据模型对地理现象的数字表示和空间描述更完备、更具有完整性,它突破了图层的概念,除了能描述空间几何目标之间的拓扑关系外,还能表达特征之间的非拓扑和属性语义关系,而这些关系在传统的基于图层的 GIS 模型中通常被遗漏掉。由于突破了图层的束缚,FOOT 提供的用于获取数据的接口已不再是狭义的图层,而是满足某种条件的动态空间对象集,这不仅使得空间分析的效率大大提高,而且使得用户可以按照需要来定义进行分析操作的数据,而不需要针对整个图层或者是将多个图幅拼接之后再行空间分析。另外,面向对象方法及其抽象机制的应用,使 FOOT 模型的语义更加丰富,从而使其能够很好地表示复杂地理现象和实体。

本系统中,将现实地理世界抽象成空间参考系模型、地理几何对象模型、空间数据库模型、元数据库模型等 4 个基本实体模型。其中,空间参考系模型用于确定现实世界与实体世界转换的坐标系统,地理几何对象模型用于将现实世界抽象为地理基本几何对象,空间数据库模型用于描述实体间拓扑关系及其关联属性,元数据库模型用于描述实体自身属性。

### 2.3 FOOT 模型数据组织

FOOT 模型在数据库中的数据组织如图版 II 图 1 和图 2 所示。其中分幅索引表用于管理逻辑分幅,由于逻辑分幅是基于空间数据库(包括各个特征类)之上的一种分幅描述,其一个“图幅”对应空间数据库中一定区域范围的所有特征类数据,因此它是

一种分幅的描述,而不具体将空间数据进行分割。这些分幅描述信息记录在专门的一个分幅索引表中。分幅索引表用于记录各图幅的图幅号、图幅名、图幅边框、图幅描述、邻接图幅信息等。空间分区表用于描述实体类的空间分区信息,它记录有分区号、分区名、最大值、边界、二叉树深度等信息。分幅索引表和空间分区表不仅可通过分幅分区对应表进行关联,且一个空间分区可以对应多个逻辑分幅。

FOOT 模型采用 3 种措施来提高图形的操作速度和效率,这 3 种措施即,整体地图基表空间分区(spatial partitioning);各分区内空间聚簇(spatial clustering);分区表的空间索引(spatial index)。

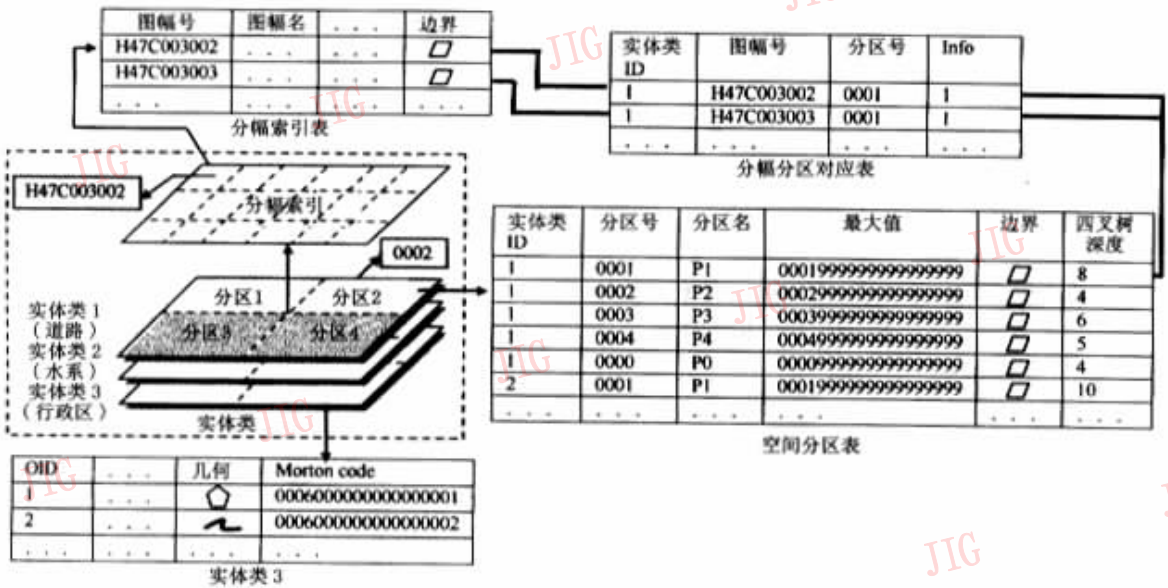


图 2 FOOT 模型数据组织

#### 2.3.1 空间分区

被处理的地理要素在空间上常常聚集于某一特定区域。空间分区就是将基于要素的空间位置放入数据库中相应的数据分区中,物理上虽可把一个表或索引分割为许多更小、更易于管理的区块,但在逻辑上仍可用一个表或一个索引管理,如图版 II 图 1 所示。空间分区后,在对大表进行处理时,只要将处理范围尽可能缩小在需要处理的数据分区内,就可大大提高海量空间表的处理速度。

##### (1) 空间分区表的创建

下面以图版 II 图 1 为例来建立整个浙江省行政区划图的空间分区表 counties。该分区空间表以 sheetcode 为分区键(partition key),而整个浙江省

行政区划图根据空间位置被分为如下 5 个分区: 0001 分区、0002 分区、0003 分区、0004 分区和 0005 分区(用来存储跨图幅要素),如,凡是空间位置落入 0004 分区空间范围内的要素都将放入分区空间表的 0004 分区中,其余依次类推;对于跨越两个或两个以上区域的要素则放入 0005 的数据分区内。空间表的分区在空间上可以不是规则图形(如图版 II 图 1 所示的矩形),其分区键可以是要素的 ID 等字段。

##### (2) 创建空间分区地图基表

以浙江省行政区划图(counties)为例来介绍其创建过程:首先将全部图幅上载后,合并成一个大表(地图基表),然后对该大表进行空间分区,如分为 4 个区域,其分别对应图版 II 图 1 所示的 0001、0002、0003

和 0004 等 4 个分区。建立分区空间表的流程如下:

① 将区域内以图幅为单位矢量数据通过空间数据库引擎 (spatial database engine, SDE) 上载到数据库。每个分区对应一个游离空间表,就添加一个与存储分区内图幅对应的 ID 字段;

② 创建空间分区表(counties)。建立 5 个分区,其中 4 个分区对应原有 4 个图幅,第 5 分区存储跨区地物要素,该表结构定义与游离空间表类似;

③ 将游离空间表数据导入空间分区表内。虽根据分区键,就可将图幅数据导入对应的空间分区中,但跨图幅地物要素必须进行接边处理后,方可存储到跨空间分区中,即一个完整的跨空间分区实体(如一条河流、道路或一个封闭自然保护区)存储为一条独立记录。

空间分区表建立后,多个用户就可以同时在同一或不同分区上进行空间数据操作,由此就可实现系统的并发处理,以提高处理速度。在空间数据库中,空间分区元数据信息由空间分区表维护,空间分区表的结构如表 1 所示。

表 1 空间分区表结构

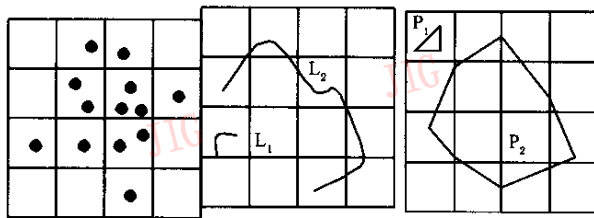
字段	说明
itemID	特征类 ID,注意不同的特征类具有不同的分区及空间聚簇参数
PartitionName	为分区名,对应 Oracle 的分区名
PartitionCode	分区号,Morton Code 字段的前 4 位
MaxValue	为分区上限,比如 0005 分区的上限为 0005999999999999999
Boundary	为一个空间分区的外包框
QUATREELEVEL	四叉树深度

### 2.3.2 空间聚簇

把空间上相近或相关联的地物要素存储在相近的物理页面上称为空间聚簇 (spatial clustering)。由于空间上相近的地物在物理页面上相近,可以减少往返搜寻的时间,因此可提高效率。但空间聚簇需要在空间表上加聚簇键 (cluster key, 如 Morton 码) 来对聚簇键值进行排序,同时将空间数据导入到另一具有同样分区的表中,以便实现空间聚簇。

在本研究中,借鉴 Morton 码的思想来实现对空间实体的编码,聚簇键值即为该编码值。由于在 GIS 中,Morton 码是对空间四叉树的格网做编码,而在此是要对地理实体做 Morton 编码,所以需要通过对生成格网 Morton 码的算法做演变来形成生成空间实体 Morton 码的算法。其演变的基本步骤分为以下两步:首先根据 FTLeve (四叉树的深度) 生成格网的

Morton 码;然后,对于点状特征,若其落入某一个 Morton 码格网内(如图 3(a)所示),则其编码为此格网的 Morton 码;若其落在格网的边线上,则取任意一个格网的 Morton 码即可;对于线状、面状特征,若其完全落入某一个 Morton 码格网内(如图 3(b)中的  $L_1$ 、图 3(c)中的  $P_1$ ),则其编码为此格网的 Morton 码,若其占据了  $n$  个格网(图 3(b)中的  $L_2$ 、图 3(c)中的  $P_2$ ),则取所占格网数最多、最临近树底层的 Morton 码,并在其后补 0,以便形成与叶子节点位数相同的编码后,再将其作为此特征的编码。



(a) Morton 码的格网 (b) 线特征 (c) 面特征

图 3 Morton 码的格网与地理特征

由于索引组织表 (index organization table, IOT) 是根据主关键字索引来存储数据,所以在此要求保证特征的编码唯一。本文拟在基于第 2 步形成的特征编码的基础上追加码段,以此保证编码的唯一性,追加的码段用于代表此格网内的第几个特征。

在空间分区表中需增加一个存储 Morton 码字的字段,该字段由两部分编码构成(如图 4 所示)。图中,分区号表示该要素对应的分区编码(编码方式类似 Morton 编码)。分区号采用 4 位表示,其最多可表示 9999 个分区(0000 号分区留作它用)。假定每个分区大小为 1.0 GB,则可以存储 10 TB 左右的单表。若 Morton 码采用 15 位表示,则可以表示的四叉树深度为 24。

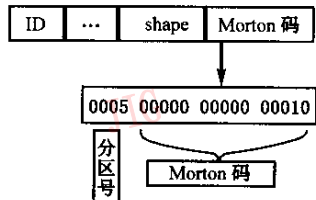


图 4 分区表 Morton 码字段及编码

在插入或更新记录时,系统内部采用两种分配方式维护 Morton 码的值。一种方法是在执行 insert 或 update 语句时,不指定 Morton 码值,由触发器生成该值;另一种方法是关掉触发器,在执行 insert 或 update 语句时,把 Morton 码值绑定上去。

以图版Ⅱ图1为例,对于分区0004中的所有要素,首先依据分区空间范围和其中的实体特征数量来确定对其进行四分的次数(四叉树的深度),该值可由用户输入,也可由系统自动确定;然后根据实体落入的网格在分区0004中的位置,计算出该网格Morton编码,并将该编码赋予该要素的MortonCode列(如图2中的实体类3表所示);接着计算各个分区的所有实体的Morton编码;最后将各个分区的实体要素依据其Morton编码进行排序后,导入另外一个具有同样分区的表中,这样就完成了整个分区空间表的空间聚簇。对一个经过Morton编码的分区空间表的处理过程可描述为:

```
create table B-temp.....as select * from A order by
MortonCode;
```

```
drop A;
```

```
Alter table B-temp rename to A;
```

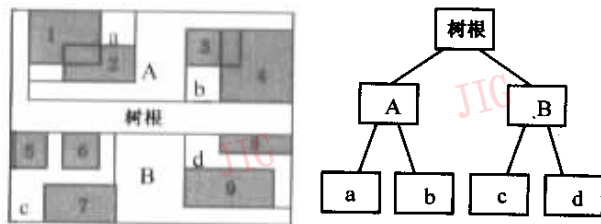
在空间数据库中,空间聚簇元数据信息由空间分区表维护,空间分区表表结构如表1所示。

### 2.3.3 空间索引

在数百万条记录的大表中,为了有效地存取空间数据,必须创建空间索引(spatial indexing)。由于数据库系统SDE的某些空间操作需要对空间表建立空间索引,其作为一种辅助性的空间数据结构,空间索引介于空间操作算法和空间对象之间,通过它的筛选,大量与特定空间操作无关的空间对象被排除,从而可提高空间操作的效率。对空间数据的查询与获取经常需要执行快速的几何搜索运算,例如:点查询、域查询,而所有这些运算均需要快速存取空间数据对象。由此可见,要支持这些搜索操作就需要特殊的空间存取方法,即必须引进索引机制。

#### (1) R 树索引

R树(R-tree)索引是目前空间数据库应用最广泛的一种空间索引结构(图5),由于它实际是对B树的扩充,因而它不仅具有B树特有的动态平衡性(所有叶节点都在同一层上),而且能进行多维索引。



(a) R 树索引结构

(b) R 树

图5 R 树通过 MBR 分等级索引

由于R树具有动态平衡的特性,因此处理空间对象时,具有很大的灵活性和较高的效率。此外,R树有一个重要的特点就是与兄弟节点对应的空间区域可以互相重叠,这样的特性使R树比较容易进行删除和插入操作。

在本研究所生成的R树中,每个非叶子节点都由若干个由MBR(minimum bounding rectangle)和PointToChild构成的单元组成,MBR为包含其对应的空间对象的最小矩形,这个最小矩形是个广义上的概念,其二维上是矩形,三维空间上就是长方体,以此类推到高维空间;PointToChild是指向其对应子节点的指针,而叶子节点则是由若干个由MBR和GeoObjectID构成的单元组成。其中GeoObjectID是空间对象的标号,通过该标号可以得到对应空间对象的详细信息。

本研究所建立的索引全部为R树索引,即对地图空间分区中的几何地物实体,其创建的MBR层次化索引如图5(a)所示。

图5中,1至9为地图空间分区中相应几何体的MBR;a、b、c、d是R树的叶节点,含有所包括的几何体的MBR和指向该几何体的指针;即a含1-MBR、1-ID、2-MBR、2-ID;b含3-MBR、3-ID、4-MBR、4-ID;c含5-MBR、5-ID、6-MBR、6-ID、7-MBR、7-ID;d含8-MBR、8-ID、9-MBR、9-ID;A含a-MBR、a-ID和b-MBR、b-ID;B含c-MBR、c-ID和d-MBR、d-ID;根节点含A和B的MBR。

创建R树时,若回滚段不足,将导致失败,而用于排序操作的空间大小也影响创建索引所需的时间,可通过修改参数来调整。一般存放索引数据及创建索引时产生的临时数据(索引创建后自动释放)分别应为 $70 \times n$ 和 $200 \times n$  Bytes( $n$ 为记录行数),用于排序操作的临时表空间一般应为 $100 \times n$  Bytes( $n$ 为记录行数)。

#### (2) 分区空间索引

在分区表上创建分区空间索引,不仅易于维护,也可减少响应时间,但对分区类型(range、hash、composite)及操作(merging、exchanging、splitting)、分区键类型应制定相应限制规则。

## 3 系统结构设计

针对三级客户/服务器模式的不足,本研究采用了分布式四层(four-tier architecture)结构模型,其

包括:数据库服务层、应用接口层、应用层和用户界面层,分别对应空间数据服务器(Oracle 数据库平台)、数据库 PL/SQL 程序包、LibraSDP 应用服务器(包括 SDP 监听进程和 SDP 服务)和客户应用程序。为了使系统能够跨平台在 Internet/Intranet 上运作,应用服务器可采用套接字(socket)技术来实现与客户应用程序间的访问,并可与采用 Java 技术开发的 Web 应用进行通信,以便实现空间信息在网络上的发布。应用服务器与数据库服务器之间的通信采用 OCI(oracle calling interface)技术,这样数据库服务器无论运行在何种操作系统下,都能实现不同操作系统下数据库服务器的跨平台运行。

### 3.1 数据库服务及应用接口层

为了不造成胖客户,同时也能减少数据服务器的负担,在四层结构模型中,空间数据库服务器主要用来存储空间数据和提供基本的数据库服务,如锁定机制、安全机制等,同时负责空间数据的索引、处理、查询和调度,并通过数据库引擎的空间操作对象来完成空间分析等功能<sup>[6]</sup>。

应用接口层用于封装各种函数、存储过程和触发器等数据库对象,为空间表创建空间索引、进行小表接边运算等。为了减少网络流量,提高处理速度,可以将特定的功能经逻辑处理和编程,封装成 PL/SQL 函数或者存储过程,便可直接调用完成具体运算,这就避免了海量数据在网络上的传输,并提高了效率。

### 3.2 应用层

应用服务器(LibraSDP)是整个空间数据处理平台的核心,它是通过创建两种后台进程(监听进程和服务进程)对用户提交的访问请求和用户权限进行管理。系统服务流程如下:

(1)启动 LibraSDP 应用服务器,该服务器用于创建后台的监听进程(listening process);

(2)监听进程在 TCP/IP 端口(默认 3495,系统管理员设置)监听客户端程序的连接请求;

(3)客户端提交用户名、密码、服务名等连接信息,发出连接请求;

(4)如果连接成功,监听进程将激发一个专用的服务进程(server process);

(5)服务进程初始化,并获取共享内存;

(6)服务进程根据用户提交的用户名、密码、数据库服务名等信息连接到 Oracle 数据库,而后续用户提交的所有的数据处理请求将全部交给该服务进

程处理并实现,与监听进程无任何关系;

(7)监听进程把客户应用请求交给服务进程进行管理之后,继续监听新的连接请求,并执行其他的管理任务。

#### 3.2.1 监听进程

监听进程(listening process, LP)用于一直监听客户的连接请求,直到 LibraSDP 关闭,服务进程则用于保持与数据库的连接,直到客户应用程序释放连接或客户程序关闭。当 LibraSDP 启动时,监听进程同时启动,并开始工作。监听进程的任务是监听客户端发出的连接请求,并激发服务进程(客户端和数据库间的专用连接),而所有由 LP 产生的服务进程都由 LP 维护。

对于每个服务进程,并不单独地维护一个数据库连接对象,它还可通过连接池对象(connection pooling feature, CPF)技术解决连接效率及资源占用问题。在启动 LP 时,创建一个 CPF,并创建少量打开的连接。在需要执行语句时,CPF 动态地通过选择一个空闲的连接来执行语句,在执行完语句之后,便释放该连接,这样就可以大大优化应用程序的性能。监听进程和客户端的通信采用流式套接字(SOCK\_STREAM)实现。由于套接字采用消息传递的方式可实现一个节点到另一个节点的信息交流,因此需要自定义消息格式。LibraSDP 根据具体的应用需求,首先定义了一套规范的消息协议,而客户则必须严格遵守该协议才能顺利地实现与 LibraSDP 的通信,另外 LP 又通过其维护的消息来解析引擎和用户消息,并执行相应的操作。

#### 3.2.2 服务进程

服务进程(service process, SP)由监听进程激发产生,一个客户对应一个专用的 SP,SP 的个数即客户端的连接数。每个 SP 根据连接的用户名和密码与客户端的应用程序通信,而数据库则根据用户名和密码确定用户的权限和可以操作的要素类。

SP 内置一个空间数据库管理(spatial database manager, LibraSDM)内核,并以此构建具体的应用逻辑,以完成空间坐标转换、投影变换、图形管理和空间分析等功能(如图 6 所示)。LibraSDM 是 SP 的核心,也是对 OCI 依照 Recordset 记录集获取模型的一个封装,其能完成的功能有:数据库连接;数据获取(数据上下载);图表管理(表、视图创建、删除、更新操作);地理几何实体管理(几何实体空间操作、空间关系判断、空间联合);锁定机制(表锁定、列锁

定、行锁定功能);元数据管理(注册、身份认证等)。

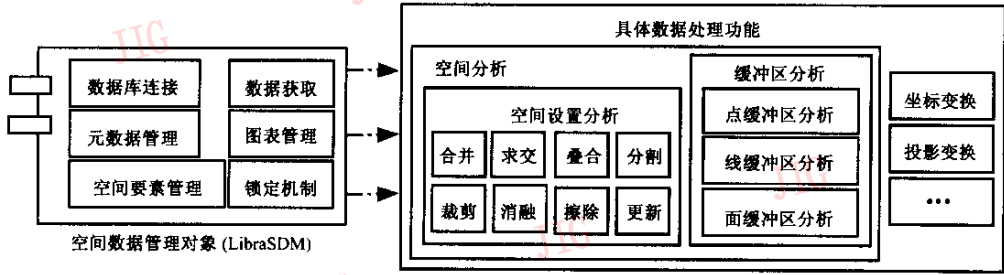


图 6 服务进程逻辑架构

### 4 功能模块设计

功能模块 (LibraSDM) 基础模块有 7 个 (如图 7 所示), 分别用于对外提供编程接口 (API)。这 7 个模块是: Common、GeoObjects、GeoDB、GeoManager、GeoOperator、SpatialReference 和 GeoDefine。下面重点介绍 Common、GeoObjects、GeoDB 和 GeoOperator 等 4 个模块。

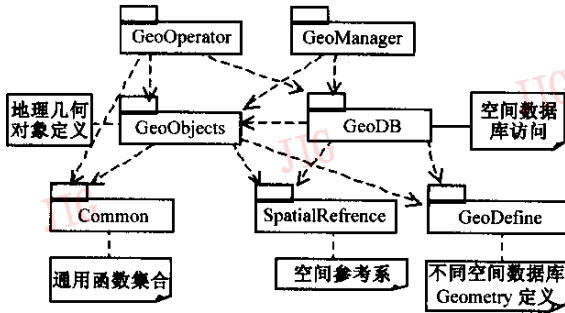


图 7 LibraSDM 架构

Common 模块主要用于描述面向地理几何对象模型、空间查询操作、数据库管理等过程中调用的一些公共类和函数,其也是空间数据管理对象中其他核心模块调用的公共部分所要实现的内容。

GeoObjects 模块用于提供一个地理几何对象模型,并使每个地理几何对象都属于几何模型中的一个类,在该模型中,其作用是将地理几何对象抽象为 6 个可实例化的子类型:点 (GeoPoint)、曲线 (GeoCurve)、多边形 (GeoPolygon)、多点 (GeoMultiPoint)、多线 (GeoMultiCurve)、复合多边形 (GeoMultiPolygon)、集合 (GeoCollection, 由点、线、面基本元素混合组成),而每个地理几何对象类又由一个基本几何元素或多个基本几何元素构成,共有 7

个原子级基本元素类型,分别是点 (ElementPoint)、连续直线 (Line)、圆弧 (CircleArc)、贝塞尔曲线 (BezierLine)、组合线 (CurveString, 由圆弧和连续直线混合组成)、多边形 (Polygon)、矩形 (Rect)。图 8 为地理几何对象模型的 UML 图。

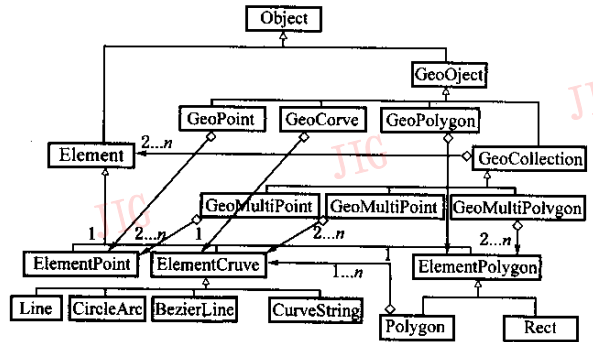


图 8 几何对象模型

GeoDB 模块主要用于实现数据库连接、图表管理、数据传输、元数据管理等基本功能。其数据库空间表中的地理实体访问和处理基于数据的游标模型 (cursor model),而使用游标模型则必须提供对空间数据库的命名,因为只有通过命名直接生成游标类,然后才能实现对实体数据的访问,游标处理模型如图 9 所示。“查询”定义了希望通过数据库连接获得的数据集,“游标”则提供反复访问空间对象集的

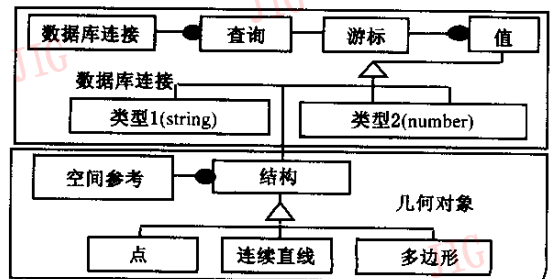


图 9 数据获取模型

机制,通过连接状态显示在集合中的当前项,提供访问集合中所有项的函数;游标返回的“值”表示结果集中元素的状态,其类型有简单类型(数字、字符串)和复杂类型(几何图元)。

GeoDB 的对象关系如图 10 所示,用于建立和维护与数据库(或数据库引擎)的连接。应用程序提供数据提供者、主机名、服务名、数据库名、用户名、用户密码、连接模式等信息。

CG\_Command 类用于返回结果集的查询声明或更新条目数量的非查询声明;CG\_Recordset 类用于获取执行 SQL(structured query language) 语句所得到的记录,由于可回滚的游标查询将加重服务器端的内存消耗而影响性能,故应提供前向  $n$  行顺序查询,在行内,可按列索引号(推荐)、列名查询;CG\_Table 类用于创建、修改、删除、查询数据库表的对象;CG\_Fields 类是字段属性定义集合,其由一系列的 field 组成。

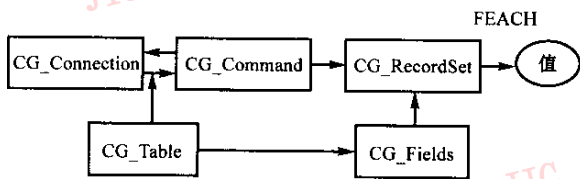


图 10 GeoDB 对象关系图

GeoDB 用于屏蔽底层各类数据库操作细节,以便为用户提供统一的数据库访问方法,并能针对不同数据库实现同一功能,其操作的差异在系统内部解决,而对外接口一致。

GeoOperator 模块严格遵循 OpenGIS 标准,结合 LibraGeometry 模型规格说明书的定义和说明来实现空间算子或操作函数等功能,如:包括叠置分析和缓冲区分析等。叠置分析包括并(Union)、交(Intersect)、叠和(Identity)、裁剪(Clip)、消融(Dissolve)、擦除(Erase)、分割(Split)、更新(Update)等。缓冲区分析包括点(集)、线(集)和面(集)的缓冲区分析;同时利用 SQL 编写或封装必要的空间查询和分析操作的类与方法,以便为客户提供屏蔽 SQL 语句操作的友好接口。

## 5 系统实验及运行实例

系统采用国家基础地理信息中心提供的地形图数据,建立了 30G 的海量空间数据库。系统具有如

下特点:①在客户端较好地实现了空间数据的放大、缩小、漫游、图层管理等 GIS 基本功能;②实现了多用户协同编辑(从海量空间数据中剪切一块图形或合并两幅图形,并对其进行编辑);③系统能对海量空间数据进行空间分析(缓冲区、图层叠置)操作,如在 30G 海量空间数据中能较快地查询出北京至深圳铁路线穿越的县市乡镇,并统计出沿线各市县的土地利用状况。

系统开发语言为 C/C++、PL/SQL,集成开发环境为 Kylix,配置管理工具为 CVS(concurrent versions system),数据库访问采用 OCI 方式,分布式计算环境构建为 BSD Socket。效率测试数据采用 1:25 万土地利用图(图幅号 gh4804),全国 1:400 万土地利用图,全国 1:400 万市县图。客户端硬件配置 CPU 为 Intel 2GHz、内存 256M,操作系统为 Red Hat AS 2.1;数据库服务器为 DELL2600,软件为 Oracle 9.2.0.4 for Linux,图形上载用时 41 s,边界跟踪(显示)用时 18 s,总共用时 59 s。其中值得注意的是,上载数据的用时,由于直接采用 GOMETRY 模型进行上载,没有使用 SQL Loader,上载 17 845 条记录用时 41 s,可见速度比较快。在上载数据过程中,没有发现内存泄漏。图版 II 图 2 为国家 863 重大攻关项目《面向网络海量空间信息的 GIS》2003 年 GIS 软件测评的测试结果。

## 6 结论

本研究试图放弃以图层为基础的传统 GIS 软件平台处理模式,尝试采用全新的面向实体对象树的空间数据模型。本研究设计了服务层、应用接口层、应用层和用户界面层 4 层体系结构,从而不仅避免了第 3 代 GIS 平台的三级客户/服务器模式的不足,而且实现了多用户同步处理机制;同时通过索引分区、空间聚类和空间索引的方式优化海量空间表,实现了海量空间数据的快速处理;整个系统基于 Linux 环境及 Oracle 数据库,客户端和数据库服务器可以跨平台操作,对于 GIS 软件在 Linux 环境下运行也作了有意义的探索。

## 参考文献

- 1 李德仁.论“GEOMATICS”的中译名[J].测绘学报,1998,27(2):1~5.
- 2 方裕,周成虎.第四代 GIS 软件研究[J].中国图象图形学报,

2001, **6A**(9):817~823.

- 3 Michael Zeiler. Modeling our world[M]. Redlands, CA, USA: ESRI Press, 1999, **7**:6~10.
- 4 李国标, 庄雅平, 王珏华. 面向对象的 GIS 数据模型——地理数据库[J]. 测绘通报, 2001, **30**(6):37~39.
- 5 程昌秀, 周成虎, 陆锋. ArcInfo8 中面向对象空间数据模型的应用[J]. 地球信息科学, 2002, **3**(4):86~90.
- 6 刘南, 刘仁义. 地理信息系统[M]. 北京: 高等教育出版社, 2002 年 7 月第一版.
- 7 刘仁义, 刘南. 一种基于 RDBMS 的空间数据管理模型及应用[J]. 中国图象图形学报, 2000, **5A**(10):825~829.
- 8 边学工. 分布式 GIS 分层体系结构模型的研究[J]. 武汉测绘科技大学学报, 2000, **25**(5):443~448.
- 9 陈斌, 方裕. 大型分布式地理信息系统的技术与发展[J]. 中国图象图形学报, 2001, **6A**(9):861~864.



刘南 1944 年生, 教授, 博士生导师。中国地理学会常务理事, 浙江省地理学会理事长。主要研究领域为面向对象的空间数据库理论。

谢炯 1977 年生, 博士研究生。研究方向为海量空间数据库理论及新一代 GIS 平台开发。

冯杭建 1979 年生, 硕士研究生。研究方向为新一代 GIS 平台开发技术。

李伟 1976 年生, 博士研究生。研究方向为海量空间数据库理论及新一代 GIS 平台开发。

严志民 1971 年生, 博士研究生。研究方向为海量空间数据库理论及新一代 GIS 平台开发。



刘仁义 1960 年生, 教授, GIS 博士。主要研究领域为面向对象的空间数据库理论、时空数据模型及 WebGIS 平台。

E-mail: liurenyi@mail. hz. zj. cn

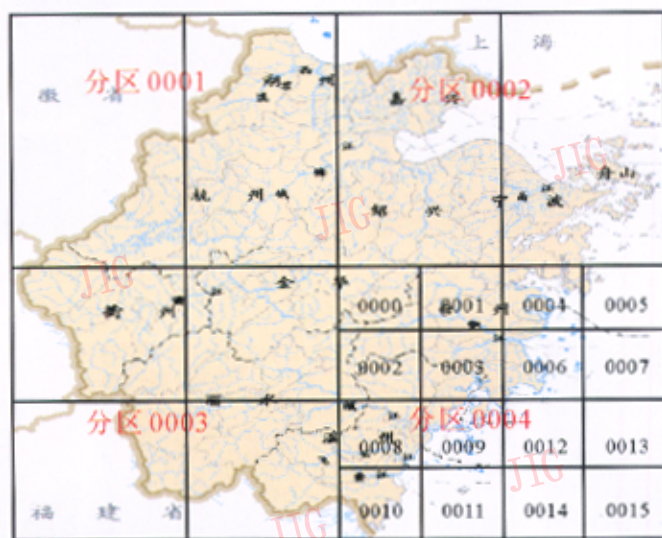
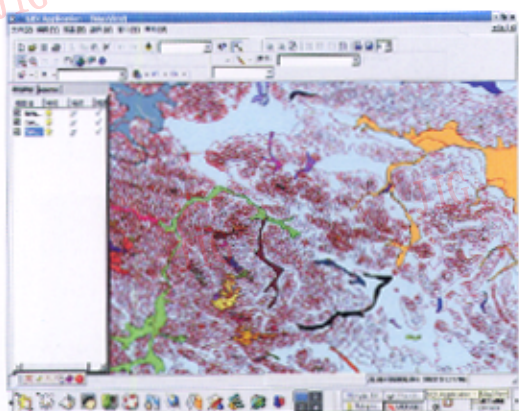
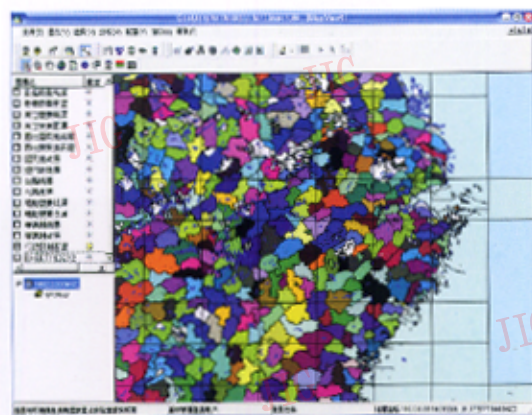


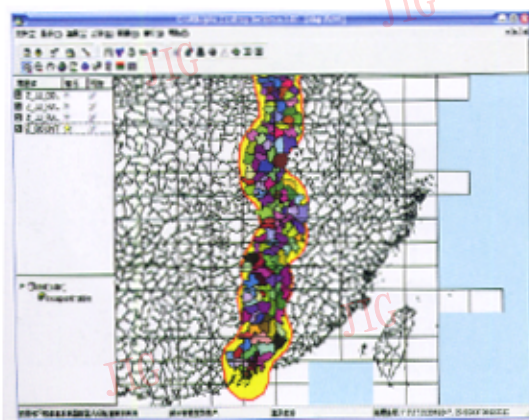
图1 浙江省地理空间分区和空间聚簇



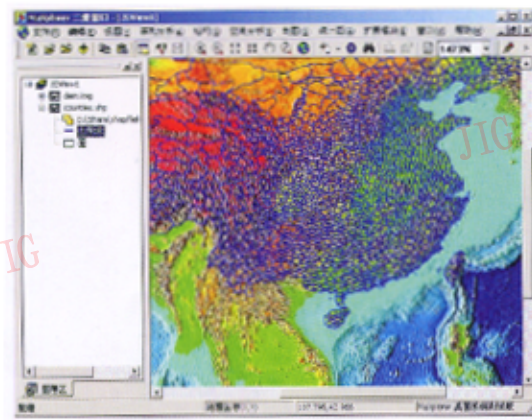
(a) 30GB海量地形数据显示



(b) 20GB空间数据查询分析



(c) 30GB海量空间数据缓冲区分析



(d) 30GB海量矢量栅格数据叠加

图2 测评软件对本系统的测试结果