

# GeoDB 空间数据库存储体系设计与实现

唐林燕

(广东轻工职业技术学院 广州510300)

## Design and Implementation of the Storage Architecture of GeoDB Spatial Database

TANG Lin-Yan

(Guangdong Light Industry Technical College, Guangzhou 510300)

**Abstract** The paper detailedly analyzes the storage architecture of GeoDB system, discusses the fundamental storage of GeoDB—Dataset, and in order to store great amount of spatial data, brings forward a mechanism of object-based compression storage for vect spatial data and design and implementation of the storage architecture of Spatial Database based on relation database.

**Keywords** Spatial database, Dataset, Vect spatial data

## 1 引言

GeoDB 的空间数据库方案基于多源空间数据无缝集成技术。为了适应不同层次存储和管理空间数据的需要,GeoDB 空间数据库支持多种实现方案。根据源数据库的不同,空间数据库主要分为两种:(1)源数据库需自定义数据结构;(2)源数据库已提供空间存储方案(如 Oracle、SDE)。本文主要探讨前一种空间数据库的存储体系的设计和实现。

## 2 GeoDB 存储体系结构

GeoDB 空间数据库由两部分组成:Datasets 和 Database Info。Datasets 是各数据集的集合,数据集是 GeoDB 空间数据库组织、存储和管理空间数据的基本单元;数据库属性信息(Infos),包括:数据库版本号(Version)、坐标体系(Coordinate System)、坐标单位(Coordinate Unit)、距离单位(Distance Unit)、投影(Projection)、描述信息(Description)等。同一空间数据库各数据集具有相同的坐标体系、坐标单位、距离单位和投影。

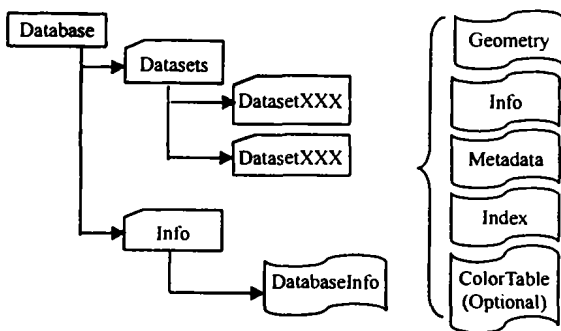


图1 GeoDB 空间数据库体系结构图

## 3 数据集设计

### 3.1 数据集定义

GeoDB 采用数据集(Dataset)作为数据库组织、存储和管

理空间数据的基本单元。数据集在概念上与 Arc/Info 的图幅工作单元(Coverage)或 Mapinfo 的表(Table)相似。Dataset 是对传统地图的一种模拟。它相当于一张地图或传统制图的单要素版,用以表示某一给定区域内地理要素的空间位置和属性信息,但不带显示属性。GeoDB 提出数据集(Dataset)概念,最主要的特点就是空间数据的独立存储。

### 3.2 数据集分类

出发角度不同,分类方法则不同,数据集的分类结果也不一样。

(1)从组成数据集的元素的空间数据类型角度出发,数据集可分为:GIS 型和 CAD 型。GIS 型数据集组成元素是具有同一空间数据类型的地理对象,一个数据集之内,存储同类型的空间数据,可以简化数据集的数据结构定义,以便实现高效的空数据存存储和管理。CAD 型数据集组成元素可包括多种不同数据类型的地理对象,此类型数据集是 CAD 软件在 GIS 系统中的扩展,此时可以将 CAD 型数据集设计为存储多种空间数据类型。

(2)从组成数据集的元素的空间数据结构类型角度出发,数据集可分为:矢量(Vector)型和栅格(Raster)型。矢量数据集由矢量数据结构的空对象构成,它在狭义上仅包括点(Point)、线(Line)、面(Region)、文本(Text)、网络(Network)、宗地(Parcel)和三角不规则网络(TIN)等类型的数据集;广义上还包括纯属性数据(Tabular)和 CAD 型数据集,由于双方数据访问风格近似,在空间数据库引擎中归类为矢量数据库引擎(详见另一论文《GeoDB 空间数据库引擎设计与实现》)。栅格型数据集由栅格数据结构的空对象构成,栅格型数据集包括:影像(Image)、格网(Grid)、数字地面高程(DEM)、影像外部连接(Image Outer Link)等类型的数据集。影像外部连接类型的数据集提供了 ECW 影像外部连接处理。

(3)从构成数据集的元素是否为固定长度的角度出发,数据集可分为:定长型(FixedLength)和变长型(Variable Length)。定长型数据集构成元素的长度固定(如点、三维点、影像块等),定长型数据集不支持压缩存储。变长型数据集构

唐林燕 副教授,主要研究方向为计算机应用、数据库技术等。

成元素的长度不定,而绝大多数地理对象长度可变。

### 3.3 GeoDB 支持的数据集类型

GeoDB 支持的数据集类型如图2所示。

	Dataset 类型	说 明	元素类型	元素长度
GIS	Tabular	只有属性表,没有几何对象	无	无
	Text	文本	Text	变长
	TextZ	三维文本	TextZ	变长
	Point	点	Point	定长
	PointZ	三维点	PointZ	定长
	Line	线	Line	变长
	LineZ	三维线	LineZ	变长
	Network	网络	Line + Point	变长
	Region	面	Region	变长
	RegionZ	三维面	RegionZ	变长
	Parcel	宗地	Region + Point	变长
	TIN	三角不规则网络	PointZ + PointGL	定长
	PointGL	带法向量的 Point3D	PointGL	定长
栅格	Image	影像	ImgBlock	定长
	Grid	格网	ImgBlock	定长
	DEM	数字地面高程	ImgBlock	定长
	ECW	ECW 压缩图像连接	ECW 压缩图像	无
CAD	CAD	CAD 型数据集	任意类型	变长

图2 GeoDB 数据集分类表

### 3.4 数据集组成

普通数据集由如下六部分组成:几何数据表(Geometry),存储空间数据的几何信息及其相关属性,如:类型、大小、边界、压缩标志以及空间分布位置等信息;信息表(Info),存储空间数据的常用信息,如:类型、选项、元素个数、数

据集边界(Dataset Bound)、空间索引边界、空间索引层数等;元数据表(Metadata),存储描述数据集的操作和其它信息,功能上相当于简单的数据字典,一条元数据记录包括一个元数据头(Title)和一个元数据内容(Content);索引表(Index),存储地理对象空间索引信息,如:对象 ID 和空间索引号;调色板表(Palette),存储影像数据集的调色板信息;属性数据表(Property),存储其它与几何信息无关的属性信息,其中系统定义的固定字段有 ID 与 UserID, ID 表示地理对象的标识符,UserID 的值由用户设定,在创建专题图中使用。数据集类型不同,包含的其它固定字段也不同。用户也可以在属性数据表中根据需要生成自定义字段。

在基于关系数据库的 GeoDB 空间数据库中,上述六个表分别存储于数据库的各对应表(Table)中。数据集类型不同,其组成部分也不同。图3给出每个 GeoDB 数据集的具体组成,"★"表示该部分为当前数据集的组成。

### 3.5 GeoDB 数据集设计要点

(1)带拓扑信息的数据集的设计 空间数据库应提供带拓扑信息的空间数据类型,主要包括 Network 和 Parcel。GeoDB 本身并不提供带拓扑信息的空间数据类型,而是通过定义一个子数据集(Child Dataset)来存储空间数据的拓扑信息。

Network Dataset 由一个 Line Dataset 和 Point Dataset 组成。其中 Line Dataset 存储 Network Dataset 的地理对象(线)的空间位置等属性;Point Dataset 描述 Network Dataset 的地理对象(线)之间的拓扑关系,存储相关节点(Node)信息。

Parcel Dataset 由一个 Region Dataset 和 Point Dataset 组成。其中 Region Dataset 存储 Parcel Dataset 的地理对象(面)的空间位置等属性;Point Dataset 描述 Region Dataset 的地理对象(面)之间的拓扑关系,存储相关节点(Node)信息。

选项 类型	Info	Metadata	Geometry	Index	Palette	Property		
						ID	UserID	Others...
Tabular	★	★				★	★	
Text	★	★	★	★		★	★	
TextZ	★	★	★	★		★	★	
Point	★	★	★	★		★	★	
PointZ	★	★	★	★		★	★	
Line	★	★	★	★		★	★	Length
LineZ	★	★	★	★		★	★	
Network	★	★	★	★		★	★	Length, NodeResistance
Region	★	★	★	★		★	★	Area, Perimeter
RegionZ	★	★	★	★		★	★	
Parcel	★	★	★	★		★	★	
TIN	★	★	★	★		★	★	
PointGL	★	★	★					
Image	★	★	★		★			
Grid	★	★	★					
Dem	★	★	★					
ECW	★	★						
CAD	★	★	★	★				

图3 GeoDB 空间数据库组成

(2) TIN 数据集的设计 TIN Dataset 由一个 PointZ Dataset 和 PointGL Dataset 组成。其中 PointZ Dataset 存储

空间三角面顶点序号(与顺序无关), PointGL Dataset 存储 PointZ Dataset 中各序号对应的具体 PointGL 类型(三维空间点坐标+法向量)的空间数据。

TIN Dataset 设计的目标是支持基于 OpenGL 的三维 GIS 的显示, 此时可将 TIN Dataset 拆分为 PointZ Dataset 和 PointGL Dataset。其中 PointGL Dataset 存储各个顶点的具体三维坐标和法向量; PointZ Dataset 存储的并不是严格意义上的三维空间点, (X、Y、Z) 分别对应一个三角面的三个顶点在 PointGL Dataset 中的存储序号。

(3) 栅格数据集设计 栅格数据集集中的 Image、Grid、DEM 型数据集存储的是 Image Block 型空间数据。在 GeoDB 系统中, 系统设定影像块必须为正方形, 边长为 2 的幂积。优点有: 简化数据结构; 提供高效的检索机制。只要给定当前块的行号和列号, 就可以通过计算得出影像块的位置。

一幅影像图在横向和纵向上被拆分为多个影像块, 但由于影像图的大小往往不是 2 的幂积, 因此拆分时, 影像的横向或纵向的最后一块 Block 中可能存在无效数据, 将会导致少量的存储空间浪费。在 Block 中可定义一个 ValidSize 变量, 通常情况下 ValidSize.Width = ValidSize.Height = Block.Size, 在最后一块中, ValidSize.Width 和 ValidHeight.Y 分别等于数据块的有效长与宽, 满足如下公式(其中 Cols 表示横向方向影像块的数目, Rows 表示纵向方向影像块的数目):

$$\text{Image.Width} = \sum_{i=0}^{i < \text{Cols}} \text{Block}_i \cdot \text{ValidSize.Width}$$

$$\text{Image.Width} \leq \sum_{i=0}^{i < \text{Cols}} \text{Block}_i \cdot \text{Size}$$

$$\text{Image.Height} = \sum_{i=0}^{i < \text{Rows}} \text{Block}_i \cdot \text{ValidSize.Height}$$

$$\text{Image.Height} \leq \sum_{i=0}^{i < \text{Rows}} \text{Block}_i \cdot \text{Size}$$

#### 4 GeoDB 压缩存储实现

海量的空间数据存储一直是空间数据库研究的热点课题之一。目前空间数据压缩的研究大多集中在影像数据, 对于矢量数据压缩的研究并不多。矢量数据的压缩必须遵循无损压缩的原则, 矢量数据的每一位信息都具有特定的空间含义, 有损压缩将导致不可预知的数据错误。

常见的数据压缩机制是将整个数据文件压缩存储, 这将导致每次使用时需将整个数据文件解压, 耗费大量的 CPU 时间, 极大地降低了系统的效率。在 GeoDB 矢量数据集中, 提出矢量数据的对象级内嵌压缩存储机制如下:

1、矢量数据压缩仅限于变长的矢量数据集, 定长矢量数据由于结构比较简单, 额外的压缩信息的存储往往与压缩后节省的存储空间相抵消。

2、压缩数据数据量过小时, 数据压缩效率不高, 在地理对象的压缩中, 可确定如果地理对象数据量小于某阈值, 不予压缩。可设此阈值为 64 个字节。

3、由地理对象控制自身的压缩。地理对象压缩后计算压缩比, 如小于某一阈值则表明该压缩有实用价值, 在数据库中存储压缩数据; 反之, 存储原数据。经统计, 可设该阈值为 2/3, 压缩标志信息存储在该地理对象对应的记录头结构中。

4、鉴于无损压缩的技术相对比较成熟, 系统的压缩算法直接下载自网上源程序。

5、压缩和解压过程占用一定的 CPU 时间, 但同时也节省

了数据读写盘时间。

图 4 给出矢量数据压缩存储流程, 图 5 给出矢量压缩数据访问流程。

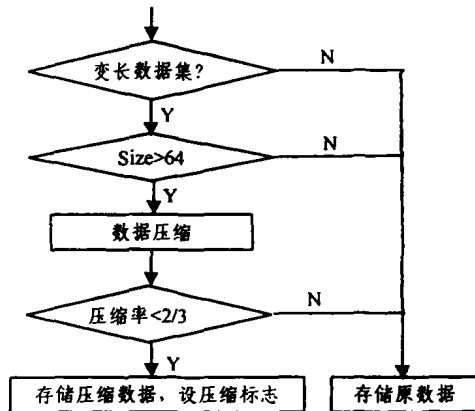


图 4 矢量数据压缩存储流程图

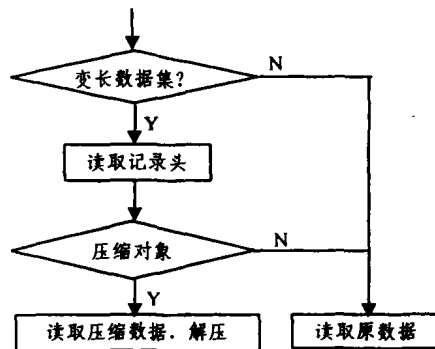


图 5 矢量压缩数据访问流程

### 5 基于关系数据库的空间数据库存储体系的设计与实现

目前, 可预计基于关系数据库的空间数据库将成为 GIS 基础平台的空间数据库首选方案。可采用 RDBMS 作为空间数据存储方案, 最大程度地利用 RDBMS 发展所取得的最新科技成果。

#### 5.1 空间数据模型与 RDBMS 的集成

在 RDBMS 中实现空间数据库的关键是如何将面向对象的空间数据模型集成到 RDBMS 的关系数据模型中。现在通用的做法是将空间数据作为整体用 RDBMS 提供的二进制的对象数据块 BLOBS 格式存储, GeoDB 也采用这种机制。GeoDB 提供多种 RDBMS 的空间数据库解决方案, 小型的有 Access, 大型的有 SqlServer 等。

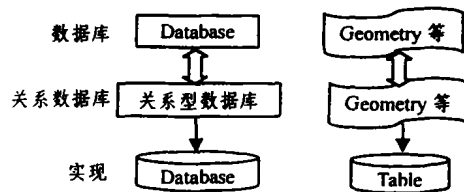


图 6 基于 RDBMS 的空间数据库存储体系对应图

#### 5.2 存储体系设计实现

基于 RDBMS 的空间数据库存储体系是建立在 GeoDB

(下转第 107 页)

SDL 进程  $P_i$  一个信号量  $S_i$ ; 当一个 SDL 进程  $P_1$  要向另一个进程  $P_2$  发送信号, 先将信号存放在这个公共区域中, 释放进程  $P_2$  对应的信号量  $S_2$ ; 进程  $P_2$  获得信号量  $S_2$  即知道有本进程的信号到达, 然后到公共区域中取回信号。

### 3.4 定时器的映射

与 SDL 信号一样, SDL 定时器的映射同样与嵌入式系统软硬件平台相关, 对于单机系统具体可采用以下两种映射方式。

• 映射为操作系统的定时器 如嵌入式操作系统内核提供定时器支持, 可以使用操作系统内核定时器来实现 SDL 定时器, 即当一个 SDL 进程创建 SDL 定时器, 相应地在操作系统内核中创建一个操作系统内核定时器, 当时间到达指定时刻, 该操作系统内核定时器向对应 SDL 进程发送定时器到信号。本方法比较直观, 实现也比较简单, 但操作系统内核定时器存在消耗系统资源多的缺点。

• 使用定时器管理进程实现 嵌入式操作系统如没有提供定时器支持, SDL 定时器可以使用专门的 SDL 定时器管理任务来实现, 即系统初始化时操作系统内核中创建一个专门的定时器管理任务 (ProcessTimer), 当 SDL 进程创建 SDL 定时器, 相应地向 ProcessTimer 注册一个定时器, ProcessTimer 定时检测其管理的 SDL 定时器, 当某一 SDL 定时器到达设定时间, ProcessTimer 则向对应 SDL 进程发送一个定时器到信号。本方式实质上是在操作系统内核外实现原来的系统定时器机制, 其缺点是定时器计时精度低于操作系统内核定时器。

表1 SDL 预定义数据类型

类型	值域	操作
Boolean	True, False	not; and; or; xor; =
Char	Character enclosed by ''	(; (=; ); ); =; Num; Chr
Integer	0; 1; ...	-; +; -; *; /; (; (=; ); ); =; Float; Fix
Real	0; ... 1; ...; 107. 86; ...	-; +; -; *; /; (; (=; ); ); =
PID	Null	none
Duration	as Real	+; -; ); *; /
Time	as Real	-; +; -; (; (=; ); ); =
Charstring	Characters enclosed by ''	Mkstring; Length; First; List; //; (index); Substring ( string; start position; length)

(上接第94页)

空间数据库的整体存储体系设计的基础上, 是其在纯关系数据库管理系统 (RDBMS) 上实现的空间数据库解决方案。两者对应关系如下: 空间数据库对应于一种源关系数据库; 在 RDBMS 存储中, 空间数据库整体设计时的 Datasets、Database Info 和 Dataset 仅仅是个概念, 而不存在具体实现; Geometry、Dataset Info、Metadata 等存储单元由 RDBMS's Table 实现。同一 Dataset 的各相关表遵循命名规则: 表名加“-”加 Dataset 名。图6给出了基于 RDBMS 的空间数据库存储体系与 GeoDB 空间数据库整体存储体系的对应关系。

总结 空间数据库是一种面向地理信息系统的工程数据库, 主要负责存储和管理空间信息。本文以空间数据库原型系统 GeoDB 为例, 详细阐述了空间数据库的存储体系结构原

对于多机系统, 定时器的映射还将面临两种选择, 即每个节点都创建一个定时器管理进程还是整个系统仅创建一个定时器管理进程。前者, 定时器到时信号传输不存在网络延迟, 但不同节点的时钟存在误差。而后者, 虽然消除了不同节点的时钟误差, 但不可避免定时器信号在网络上传输存在延迟。如何选择, 这是硬实时的分布式系统研究的难点。

### 3.5 预定义数据类型的映射

表1给出了 SDL 语言的预定义数据类型及其可能取值和运算操作。它们与 C 语言的基本数据类型相似, 映射转换比较简单, 与嵌入式系统软硬件平台基本无关, 限于篇幅本文不作详细描述。

结束语 本文简单介绍了自动代码生成技术的基本概念, 详细分析了基于 SDL 语言自动代码生成技术的主要环节, 剖析了影响最终代码性能的主要因素: 进程的并行性、数据拷贝、定时器精度等, 并提出了针对不同软硬件平台和应用特性的改进办法。

## 参考文献

- Manas J A, de Miguel-More T. From LOTOS to C. In: K. J. Turner, ed. Formal Description Techniques. North-Holland, 1989. 79~84
- van Bokhoven L J, Voeten J P M, Geilen M C W. Software Synthesis for System Level Design Using Process Execution Trees. In: Proc. of the 25th EUROMICRO Conf. Sept. 8-10, Milan, Italy, Vol. I, 1999. 463~467
- Gotzhein R, et al. Improving the efficiency of automated protocol implementation using Estelle. Computer Communications, 1996. 19 (14): 1226~1235
- Jia X, Skevoulis S. Code Synthesis based on Object-Oriented Design models and Formal Specifications. In: Proc. of the Twenty-Second Annual Intl. Computer Software and Applications Conf. 1998. 393~398
- Passerone C, et al. Modeling Reactive Systems in Java. In: Proc. of the Sixth Intl. Workshop on Hardware/Software Codesign, Seattle, Washington, USA, 1998
- Hsiung P A. Formal Synthesis and Code Generation of Embedded Real-Time Software. In: Proc. of the ACM/IEEE 9th Intl. Symposium on Hardware/Software Codesign, New York, USA, 2001
- ITU-T Recommendation Z. 100 Annex F: SDL Formal Semantics Definition. International Telecommunication Union, Geneva, 2000
- Fernandes, Joo Machado R, Santos H. Modeling Industrial Embedded Systems with UML. 8th IEEE/IFIP/ACM Int. Workshop on Hardware/Software Co-Design, E. U. A., May, 2000
- Spivey J M. The Z Notation: A Reference Manual. Prentice Hall, 2nd edition, 1992
- Heitmeyer, Constance L. Software Cost Reduction. Encyclopedia of Software Engineering, Vol. I, J. J. Marciniak, ed. Jan. 2002

理、设计和实现; 讨论了空间数据库的基本存储单元——数据集; 针对空间数据库的海量数据存储问题, 本文提出了矢量空间数据的对象级压缩存储机制及基于 RDBMS 的空间数据库存储体系的设计实现。

## 参考文献

- 柏廷臣, 李新, 冯学智. 空间数据分析与空间模型. 地理研究, 1999 (2)
- 吴信才. 地理信息系统的基本技术与发展动态. 地球科学, 1998 (4)
- OpenGIS Consortium, The OpenGIS Specification Mode Topic 1: Feature Geometry Version 3, March, 1998
- Gtting R H. An Introduction to Spatial Database Systems. Spatial Database Systems of the VLDB Journal, 1994, 3 (4)