


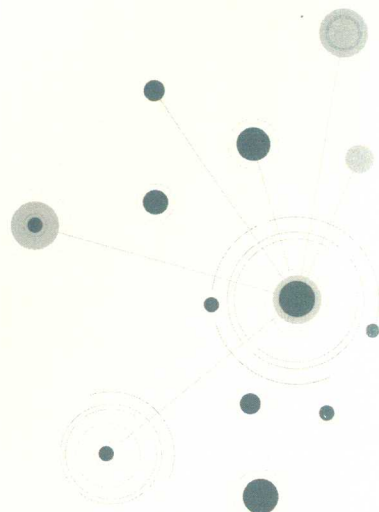


# GIS 并行处理中的 空间数据域分解技术研究

周 芹 黄耀欢 著




 气象出版社  
China Meteorological Press



# GIS 并行处理中的空间数据域 分解技术研究

周 芹 黄耀欢 著



 气象出版社  
China Meteorological Press

## 内容简介

GIS的功能主要体现在空间数据存储与管理、空间数据可视化和空间分析三个方面。空间分析作为GIS的核心模块,是区别于其他MIS系统(信息管理系统)的主要标志。GIS中的多数空间分析算法同时具备计算密集和数据密集两大特点,并行加速难度大。本书针对空间分析的并行化难题,从数据域分解的角度出发,探讨了GIS软件空间分析并行处理的框架,设计了多种基于空间索引的数据域分解方法,并实现了原型系统,选取了三个具有代表性的空间分析算法:R树的并行加载、线段求交并行处理和最短路径并行查找,对相关关键技术进行验证。本书适用于对GIS有一定了解的读者,尤其是在空间分析算法并行化研究方面,本书能起到入门指引的作用。

### 图书在版编目(CIP)数据

GIS并行处理中的空间数据域分解技术研究/周芹,  
黄耀欢著. —北京:气象出版社,2019.12  
ISBN 978-7-5029-7130-4

I. ①G… II. ①周… ②黄… III. ①空间信息技术—  
并行处理 IV. ①P208

中国版本图书馆CIP数据核字(2019)第285039号

GIS BINGXING CHULIZHONG DE KONGJIAN SHUJUYU FENJIE JISHU YANJIU

### GIS并行处理中的空间数据域分解技术研究

周芹 黄耀欢 著

出版发行:气象出版社

地址:北京市海淀区中关村南大街46号

邮政编码:100081

电话:010-68407112(总编室) 010-68408042(发行部)

网址:<http://www.qxpbs.com>

E-mail: [qxpbs@cma.gov.cn](mailto:qxpbs@cma.gov.cn)

责任编辑:简学东

终审:吴晓鹏

责任校对:王丽梅

责任技编:赵相宁

封面设计:博雅锦

印刷:北京建宏印刷有限公司

开本:787 mm×1092 mm 1/16

印张:7.25

字数:206千字

彩插:1

版次:2019年12月第1版

印次:2019年12月第1次印刷

定价:55.00元

本书如存在文字不清、漏印以及缺页、倒页、脱页等,请与本社发行部联系调换。

# 前 言

近年来,随着信息技术的发展和应用需求的拓展,地理信息系统(Geographic Information System, GIS)快速发展,地理信息科学和技术发生了巨大的变革。目前, GIS 发展的最大特点是更为紧密地与计算机技术、网络技术和通信技术结合,形成各类专业应用和地理信息服务。伴随着技术的变革与应用模式的更新,支撑海量数据管理和高速运行的大型系统是 GIS 领域的重要研究目标。基于空间数据域分解的空间分析并行计算技术的研究为大型 GIS 空间数据库管理和应用探索了一条新的途径。

## 1. 高性能 GIS 的发展

高性能 GIS 是当前地理信息科学最为重要的课题之一。而高性能 GIS 需要集中解决两个核心技术问题,一是海量数据处理,二是高性能计算。

大型空间数据库建设已经成为许多行业应用的基本要求。在我国,如国家电子政务系统的建设、国家空间基础设施的建设,空间数据库的数据量均达到 TB 甚至 PB 级;全球地理信息系统化搜索软件谷歌地球(Google Earth)上全球地貌影像的有效分辨率至少为 100 m,针对大城市、著名风景区、建筑物区域提供分辨率为 1 m 和 0.6 m 左右的高精度影像,Google Earth 部署在全球的数据服务器多达千万台。对海量空间数据进行合理组织、有效的管理并投入应用,充分发挥其作用,是 GIS 目前面临的重大挑战。

高性能计算用于复杂的 GIS 模型或算法造成处理过程中高密度的计算以及 I/O 操作。例如,对空气污染情况进行监测与预报,须将气象以及污染源的条件输入数学模型进行精确分析,完成诸如选择性排放措施的影响模拟、实时预报以及

确认新的污染源等功能。

应用模式的改变对 GIS 软件基础功能的实时性提出了更高的要求。在应用系统中,Web GIS 的广泛应用彻底改变了 GIS 传统的应用模式,使得目前构建地理信息系统在软件体系架构上从传统面向数据/应用的方式向面向服务的方式转变。Internet 技术与 GIS 相结合,GIS 通过万维网功能得以扩展,真正成为一种大众使用的工具,从万维网的任意一个节点,Internet 用户可以浏览 GIS 站点中的空间数据、制作专题图,以及进行各种空间检索和空间分析,从而使得普通网民容易访问和使用地理信息,GIS 有机会被大众所接受。面向海量数据的高效、实时的检索查询、显示、空间分析操作是这种面向服务的应用模式的基本要求。

由此可见,空间应用领域的不断扩大、空间数据的海量增长,特别是大型应用系统所需要的 TB 级甚至 PB 级海量存储空间,对 GIS 软件的基础功能提出了更高的要求,面向海量数据的存储、高效检索查询、显示、空间分析操作等,对现有的 GIS 软件在性能和扩展性方面提出了严峻的挑战。

## 2. 多核对并行计算技术的推动

多核是指在一枚处理器中集成两个或多个完整的计算引擎(内核),多核处理器即单芯片多处理器(Chip Multi Processors,CMP),是单枚芯片能够直接插入单一的处理器插槽中,操作系统会利用所有相关的资源,将它的每个执行内核作为分立的逻辑处理器,用以增强性能、降低能源消耗以及更有效的多任务并行处理。

CMP 最早是在 1996 年由美国斯坦福大学提出的,其思想是在一块芯片内实现对称多处理(Symmetrical Multi-Processing,SMP)架构,且并行执行不同的进程。早在 20 世纪末,惠普(HP)和 IBM 就已经提出双核处理器的可行性设计。

多核处理器技术具有普通单核、单线程处理器所没有的性能优势,目前已成为提高处理器性能的主要途径(刘近光等,2007)。多核心技术在应用上的优势有两个方面:为用户带来更强大的计算性能和满足用户同时进行多任务处理和多任务计算环境的要求。多核处理器也为很多的应用领域提供了新的解决方案,使得我们必须面对并发和并行操作这些通常是并行计算的专业人员和高端用户才需要面对的问题(张云泉,2008)。

当前各个主流处理器厂商都推出了自己的多核产品。IBM 在 2001 年就推出了基于双核心的 POWER4 处理器,随后是 Sun 和惠普公司,都先后推出了基于双核架构的 UltraSPARC 以及 PA-RISC 芯片,但此时双核心处理器架构还都是在高端的精简指令集计算机(Reduced Instruction Set Computing, RISC)领域,直到 2006 年 Intel 和 AMD 公司相继推出自己的双核心处理器,双核心才真正走入了主流的 X86 领域。Intel 公司 2006 年年底抢先推出了自己的四核产品,AMD 公司也在 2007 年推出自己的四核产品。八核甚至更多核的产品也已经或即将出现。

发挥出多核处理器硬件的优势,需要从编译器技术、操作系统及其他系统软件的架构上来加以发掘。多核处理器对操作系统和程序设计等都提出了很多挑战,需要思考怎么解决这些问题。应对多核处理器的软件开发,可以有几种解决思路,包括硬件隐藏、自动并行、多线程、优化、新并行语言等。一些新的高生产率和支持全局地址空间的并行程序语言已经出现,而且正在快速发展,对应对多核处理器的挑战提供了可能的最终解决途径。

面对多核系统,需要有并行编程的思想才有可能充分利用资源。对于一些线性模型,如 Google 的查找算法,对处理器系统的要求是多多益善;一些用于网络游戏的服务器,也能够从多核之中看到可支持用户数量的成倍增长;对于一些需要科学运算的石化、石油勘探等大量数据转换来说,不仅需要多核,还需要大量的显示芯片实现数据的可视化;在仿真领域,大量的多核可以充分模拟仿真的网格特性,因而可以在气象等高密度计算中充分利用。对于 GIS 软件来说,通过并行技术充分利用多核提供的计算能力是目前提高空间数据的处理能力行之有效的方法(汪少敏等,2008)。

### 3. 并行计算技术与 GIS 空间分析

GIS 的功能主要体现在三个方面:数据库功能、可视化功能和分析功能(胡鹏等,2005)。空间分析作为 GIS 的核心模块,是区别于其他 MIS 系统(信息管理系统)的主要标志。GIS 经过近 50 年的发展,常用的空间分析算法如缓冲区分析、路径分析、叠加分析等,都得到了深入研究。很多算法,如最短路径,在串行计算环

境下几乎已经达到理论上的时间复杂度的极限(Narsingh et al,1984)。

应用系统中的基本空间数据处理功能主要由商业 GIS 软件作为其特定模块提供, GIS 数据量的急剧膨胀和应用要求的提高导致 GIS 计算任务既是数据密集型又是计算密集型, 在空间数据获取能力空前提高的背景下, GIS 软件无法突破海量数据空间分析的瓶颈问题, 空间分析效率不尽如人意, 空间数据也得不到有效分析利用。随着计算机硬件的不断发展, 可被利用的计算资源变得相对廉价也容易获取, 并行计算不再是专为解决某一个特定问题而采用的技术。GIS 必须利用并行计算技术, 以充分利用计算资源提高空间数据处理的性能。GIS 作为一种 IT 技术, 很大程度上受计算机硬件、软件的影响, 计算机技术的发展推动 GIS 技术的进步, GIS 必须密切关注计算机领域的技术发展, 以顺应这种发展趋势。

在 GIS 的高性能计算领域, 学者们已经注意到并行技术带来的好处, 并进行了相关研究, 但大多集中于上层架构的设计, 更多的是关注通过集群、分布式, 甚至网格聚集的计算资源获取空间数据处理的高性能, 如 Rocks 集群下的并行 GIS (靳华中等, 2005; 王银娣, 2005)、基于 GIS 模型计算的网格 GIS (扈海波等, 2007), 这些系统通常需要 GIS 软件提供基本的空间分析功能, 而且这部分的性能决定了整个系统的加速上限。并行算法在数值计算领域的研究较丰富, 但是在空间分析领域中则亟待深入(扈海波等, 2007)。目前很少有 GIS 软件通过并行技术来获得高性能, 本书则在这方面尝试进行探讨。

# 目 录

## 前 言

|                              |        |
|------------------------------|--------|
| 第 1 章 并行处理中空间数据域分解研究进展 ..... | ( 1 )  |
| 1.1 并行计算与空间分析并行算法研究现状 .....  | ( 1 )  |
| 1.1.1 并行计算机和并行计算体系结构 .....   | ( 1 )  |
| 1.1.2 GIS 空间分析研究现状 .....     | ( 3 )  |
| 1.2 空间数据库引擎研究进展 .....        | ( 4 )  |
| 1.2.1 空间数据库模型 .....          | ( 4 )  |
| 1.2.2 空间数据库引擎的体系结构及对比 .....  | ( 5 )  |
| 1.2.3 空间数据库引擎关键技术 .....      | ( 8 )  |
| 1.3 空间索引技术研究进展 .....         | ( 10 ) |
| 1.3.1 空间索引结构的必要条件 .....      | ( 10 ) |
| 1.3.2 空间索引分类 .....           | ( 11 ) |
| 1.3.3 常用空间索引结构 .....         | ( 12 ) |
| 1.4 空间数据分解技术研究现状 .....       | ( 14 ) |
| 1.4.1 空间数据模型 .....           | ( 14 ) |
| 1.4.2 空间数据域分解 .....          | ( 15 ) |
| 1.4.3 空间数据的分解粒度 .....        | ( 16 ) |
| 1.4.4 空间数据的分解方法 .....        | ( 18 ) |
| 1.5 并行 I/O 策略研究现状 .....      | ( 19 ) |
| 1.5.1 并行 I/O 的目标 .....       | ( 19 ) |
| 1.5.2 冗余磁盘阵列方法 .....         | ( 20 ) |
| 1.5.3 并行文件系统 .....           | ( 20 ) |
| 1.6 小结 .....                 | ( 21 ) |

|                                    |        |
|------------------------------------|--------|
| <b>第 2 章 基于空间数据库的空间分析并行处理框架设计</b>  | ( 22 ) |
| 2.1 空间分析并行处理框架设计的基础                | ( 22 ) |
| 2.1.1 共相式 GIS                      | ( 22 ) |
| 2.1.2 并行编程环境                       | ( 25 ) |
| 2.1.3 并行算法的性能指标                    | ( 29 ) |
| 2.2 GIS 空间分析算法分类                   | ( 31 ) |
| 2.2.1 空间分析理论体系                     | ( 31 ) |
| 2.2.2 GIS 软件与空间分析                  | ( 34 ) |
| 2.2.3 常用并行空间分析算法及分类                | ( 35 ) |
| 2.3 空间分析并行处理框架设计                   | ( 38 ) |
| 2.3.1 空间分析并行处理框架结构                 | ( 38 ) |
| 2.3.2 空间数据模型                       | ( 41 ) |
| 2.3.3 并行处理框架与空间数据库引擎               | ( 42 ) |
| 2.3.4 并行处理模式                       | ( 45 ) |
| 2.4 小结                             | ( 48 ) |
| <b>第 3 章 空间数据域分解和原型系统关键技术及解决方案</b> | ( 49 ) |
| 3.1 四种外挂式空间索引的特点                   | ( 49 ) |
| 3.2 空间数据域分解                        | ( 50 ) |
| 3.2.1 基于 R 树的数据分解                  | ( 51 ) |
| 3.2.2 基于四叉树的数据分解                   | ( 57 ) |
| 3.2.3 基于图幅索引的数据分解                  | ( 59 ) |
| 3.2.4 基于多级网格索引的数据分解                | ( 61 ) |
| 3.3 空间索引的并发编辑                      | ( 63 ) |
| 3.3.1 基于锁定的并发编辑                    | ( 63 ) |
| 3.3.2 CQR 树的提出                     | ( 63 ) |
| 3.3.3 CQR 树算法描述                    | ( 66 ) |
| 3.3.4 CQR 的空间查询策略及性能分析             | ( 67 ) |
| 3.4 并行 I/O 策略                      | ( 70 ) |
| 3.4.1 空间数据并行 I/O 的目标和原则            | ( 70 ) |
| 3.4.2 空间数据的分区存储                    | ( 70 ) |

|              |                          |         |
|--------------|--------------------------|---------|
| 3.4.3        | 空间数据缓存和预取策略 .....        | ( 72 )  |
| 3.5          | 负载均衡 .....               | ( 76 )  |
| 3.6          | 小结 .....                 | ( 76 )  |
| <b>第 4 章</b> | <b>实验研究</b> .....        | ( 77 )  |
| 4.1          | 空间索引并行加载算法 .....         | ( 77 )  |
| 4.1.1        | 问题的提出 .....              | ( 78 )  |
| 4.1.2        | 基于 TGS 算法的 R 树并行构建 ..... | ( 78 )  |
| 4.1.3        | 查询效率分析 .....             | ( 80 )  |
| 4.1.4        | 实验分析 .....               | ( 81 )  |
| 4.2          | 线段求交算法的并行化 .....         | ( 85 )  |
| 4.2.1        | 相关研究 .....               | ( 86 )  |
| 4.2.2        | 基于规则格网的线段求交算法 .....      | ( 87 )  |
| 4.2.3        | 实验分析 .....               | ( 90 )  |
| 4.3          | 最短路径查找的并行化 .....         | ( 92 )  |
| 4.3.1        | 相关研究 .....               | ( 92 )  |
| 4.3.2        | 最短路径搜索串行算法 .....         | ( 93 )  |
| 4.3.3        | 网络分割最短路径搜索并行算法 .....     | ( 94 )  |
| 4.3.4        | 静态网络最短路径搜索并行算法 .....     | ( 95 )  |
| 4.3.5        | 实验分析 .....               | ( 95 )  |
| 4.4          | 小结 .....                 | ( 97 )  |
| <b>第 5 章</b> | <b>结论与展望</b> .....       | ( 98 )  |
| <b>参考文献</b>  | .....                    | ( 100 ) |

# 第1章 并行处理中空间数据域分解研究进展

经过近50年的发展, GIS 软件在 GIS 体系中一直占据着重要的位置。从 GIS 内涵上分析, GIS 可以从科学(Geographic Information Science)、软件(Geographic Information Software)和服务(Geographic Information Service)三个层次来加以考量, 其中, 服务是目标, 软件是手段, 科学是基础(方裕等, 2001)。从 GIS 的组成部分上分析, GIS 由硬件、软件、数据和用户组成(龚健雅等, 2008), 其中, 硬件是骨架, 软件是精髓, 数据是基础, 用户是灵魂(闵安成等, 2004)。

GIS 软件技术体系的发展经历了六个阶段: GIS 模块、集成式 GIS、模块化 GIS、核心式 GIS、组件式 GIS 和万维网 GIS。无论在哪个阶段, GIS 软件的功能都脱离不了空间数据存储、空间分析和可视化这三方面。计算机技术的发展是推动地理信息技术飞速发展的主要动力, 并行计算技术以求解大规模问题为目标。自 20 世纪 90 年代以来, 并行计算机的体系结构趋于成熟, 并行计算技术得以飞速发展; 而近年来多核多处理器的普及使得在普通计算机上通过并行计算提高性能成为可能, 这无疑对并行技术又是一次极大的促进。GIS 利用并行技术获取高性能是一种行之有效的方法, 国内外相关学者对并行计算技术在 GIS 中的应用做了大量研究和实验, 而在 GIS 软件中对并行技术的支持却很少有学者进行研究。本章通过分析空间分析相关并行算法的研究现状, 对 GIS 软件中空间分析的基础——空间数据存储相关技术, 即空间数据库引擎和空间索引进行阐述。

## 1.1 并行计算与空间分析并行算法研究现状

### 1.1.1 并行计算机和并行计算体系结构

并行计算技术是利用多计算机/多 CPU 来处理一个普通的计算任务的技术,

这样可以充分利用总体的内存单元与性能(Boisseau, 2001)。并行计算(parallel computing)是指在并行机上,将一个应用分解成多个子任务,分配给不同的处理器,各个处理器之间相互协同,并行地执行子任务,从而达到加速求解速度,或者扩大求解应用问题规模的目的(张林波等, 2006)。

对于信息技术来说,并行机的发展经历了较长时间。20世纪60年代初期,美国就出现了全世界最早的并行计算机,70年代之后还出现过向量机、并行向量机(以Cray公司产品为代表,中国的则是银河系列),以及90年代初的并行机,其代表作就是Think Machine公司的CM系列并行机。Think Machine公司通过将大量体系结构简单、功能较弱的处理器用网络连接实现大规模的并行计算系统,但由于应用领域有限等原因,这类系统很快就消亡了。曾经一度流行且目前还占有一席之地并行计算机是基于MPP的非共享存储集群和基于SMP的共享存储小型机。但目前市场上占据主流位置的是Cluster集群系统。

并行计算机的研究已经有多年的历史,1958年,Gill发表《并行编程》的论文,1959年Hooland发表《能同时执行任意多子程序的计算机》,开创了并行计算的先河,在其后的40年内,计算机科学的大量研究工作集中在并行计算机的研究上,Flynn和Rudd在1996年提出,“对高性能系统的不断增长的需求……将向我们指明一个简单的结论:并行是计算机的未来”。

根据数据(Data)和指令(Instruction),Flynn计算机分类法把计算机分为四类:MIMD, SIMD, SISD, MISD。传统的单处理器计算机称为单指令流单数据流(SISD);而在通用的多处理机系统中,每个处理器执行一个指令流对不同的数据流进行操作,Flynn称之为多指令流多数据流(MIMD),如果将程序设计为单一的指令流,在多个处理器上同时执行,进行不同数据流的操作,这类特殊的计算机系统称之为单指令流多数据流(SIMD)计算机系统。

SIMD结构是最先被研究的一类并行计算系统体系结构。其基本原理是有多个处理机构成并行处理系统,分别对不同的数据执行相同的操作。采用SIMD体系结构的并行处理系统具有如下几个特点:多个处理机紧密耦合、多个处理机在任务执行上是同步的、整个系统中存在一个集中的控制部件、多个处理机分别对不同的数据执行相同的操作。这些特点使SIMD结构系统在处理上具有向量处理特性,因此,它是大型向量或阵列并行处理所采用的主要体系结构(郑江, 2004)。

MIMD 并行处理系统同样也由多个中央处理器组成。但不同的是,这些处理器各自控制自己任务的执行,采用的是异步的控制方式。

对应于 Flynn 的计算机分类思想,产生了两种并行计算任务分解方法:任务并行(Task Parallelism)和数据并行(Data Parallelism)。任务并行机制要求计算任务能够分解成相对独立的处理过程,把这些独立的子任务分配给不同的处理器。不同处理器同时执行不同的任务以加速计算过程。GIS 空间分析计算中的任务并行已经有很多学者研究。与任务并行机制不同,数据并行机制则将数据进行分解,不同的处理器针对一个或者多个子数据处理,各个处理器上执行的是同一套操作。

### 1.1.2 GIS 空间分析研究现状

面对海量空间数据分析、复杂的计算模型、苛刻的计算时间以及高性能 GIS 服务要求,传统的空间分析算法已经不能满足应用的需要。将并行计算应用在 GIS 领域中是否有必要,这个问题经过了一个长期的讨论,最后由 Armstrong(2000)结束了争论,他指出:随着空间数据的增加,以及分析算法的复杂化,当单处理器无法满足这种对处理能力增大的需求时,必须用多处理器才能达到合理的运行时间。

随着对并行硬件的投入研究(如 Cray 公司)和开发软件(如 MPI)的发展,以及需要处理大容量数据的刺激,人们开始运用并行计算技术对 GIS 各个方面展开研究。由于 GIS 的空间分析算法较为复杂,其处理往往非常耗时,因此在并行环境下的 GIS 软件研究主要集中在空间分析算法的并行化上。GIS 经典的空间分析算法,如缓冲区分析、线段求交、叠加分析、最短路径、旅行商等问题,都有人对此在并行环境中展开相关研究,并都取得了较好的效果(Turton, et al, 1997; Densham, et al, 1998; Cramer, et al, 1999; 姚艺强 等, 2007; Franklin, et al, 1998; 邵回祖, 2007),他们都对 GIS 的空间分析算法的并行进行了深入探讨。但是这些研究和应用成果功能分散,都是针对某一类具体问题求解,有特定的输入、输出条件,经典算法也难以进行复用。

在并行技术应用用于 GIS 领域之前,学者们都致力于串行算法的不断优化以提高计算效率,但这些被优化的串行算法并不适合并行执行。在 GIS 大规模数据的处理中,数据的并行比任务的并行更易于实现,基于数据的并行可以有效地将数据映射到各个处理器上。而在集群计算中,基于数据的分解计算也易于实现集群

规模的动态变化,数据可以根据集群节点的变化实现动态分解,但任务的动态分解则比较难以实现。

## 1.2 空间数据库引擎研究进展

### 1.2.1 空间数据库模型

空间数据是 GIS 的基本构成之一,GIS 应用领域的推广、应用系统中数据量的急速增长及数据来源的多样性,使得依赖于具体 GIS 软件平台的文件型数据库不能满足实际需要,越来越多的 GIS 应用系统采用成熟的商业数据库来存储空间数据。

空间数据库引擎是一种处于应用程序和数据库管理系统之间的中间件技术,在用户和异构空间数据库之间提供开放接口。用户通过空间数据库引擎将不同形式的空间数据提交给数据库管理系统(DBMS),由 DBMS 统一管理;也可以通过空间数据库引擎从 DBMS 中获取空间类型的数据,并转化为客户可以使用的格式。因此,DBMS 实质上是形式各异的空间数据的容器,而空间数据库引擎就是空间数据出入该容器的通道。

各种空间数据库引擎在本质上都是通过利用和扩展符合工业技术标准的 RDBMS 的数据类型和功能,来实现空间数据在数据库中的物理存储。为了支持异构平台的访问和互操作,各种几何类型都遵循 OGC(Open Geospatial Consortium)等国际标准。借助 RDBMS 的强大功能,空间数据库引擎实现了“矢量—栅格”一体化存储、数据完整性和一致性的维护,提供严格的规则/有效性检查,支持长事务处理和版本管理、用户权限管理和数据安全机制,并可实现分布式存储和用户透明的分布式数据库操作。

理论上空间数据库模型包括混合数据模型(Hybrid Model)、统一数据模型(Integrated Model)、扩展结构模型(Extended Model)、面向对象模型(Object-Oriented Model)和时空数据模型(Spatial-Temporal Model)(汤国安等,2000)。随着各种商业数据库的发展,目前在 GIS 领域中应用比较成熟的是统一数据模型、混合数据模型和扩展结构模型。

混合数据模型把空间数据和属性数据分开存储,即空间数据及其拓扑关系存放在文件中,属性数据存放在关系数据库中,二者通过唯一的标识符建立联系。SuperMap 软件中 SDB 和 MapInfo 软件的 Table 格式采用的都是混合数据模型。混合数据模型适用于小型 GIS 应用系统,无法应对海量空间数据的处理,同时,数据的安全性、一致性难以保证,基于文件的存储系统也难以实现多用户并发操作。

统一数据模型是一种纯关系数据模型,空间数据和属性数据都用关系数据库的二维关系表来存储,使用标准关系连接机制建立空间数据与属性数据的关联。国内 SuperMap SDX+ 和国外 ESRI 的 Arc SDE、MapInfo 的 SpatialWare 软件均采用了统一数据模型。统一数据模型易于与数据库现有的功能相结合,保证数据的安全性;由于空间数据和属性数据都用关系表存储,数据的一致性容易维护,“空间数据—属性数据”一体化存储和“矢量—栅格”一体化都较容易实现;数据库提供的并发控制机制也可以用于空间数据的多用户并发操作。

扩展结构模型采用统一的 DBMS 存储几何数据和属性数据,与统一数据模型不同的是,它在标准的关系数据库上增加几何管理层,也可称为对象—关系模型。这种扩展模型结构一般都在数据库内核进行扩展,从数据库底层提供对空间对象、空间操作、空间查询和空间分析的支持。随着 GIS 相关市场的发展不断看好,各大数据库厂商意识到对空间数据支持的重要性,通过引入抽象数据类型等手段支持异构空间数据的存储和管理。其发展态势大有成为数据库管理系统标准配置的趋势。其中,最为著名的国外商用数据库空间数据管理扩展产品有 Oracle 公司的 Spatial Spatial(Oracle,2002)、IBM 公司的 DB2 SpatialExtender(IBM,2003)和 Informix Spatial DataBlade(IBM,2003)及其以开放源码而引起广泛关注的 MySQL 公司的 Spatial Extensions(MySQL,2003)和 PostgreSQL 数据库的 PostGIS(PostGIS,2005)等。

### 1.2.2 空间数据库引擎的体系结构及对比

自 1995 年 ESRI 公司发布 ArcSDE 1.0 以来,各 GIS 平台纷纷推出自己的空间数据引擎,各大数据库厂商也意识到支持空间数据的重要性,在原有关系数据库的基础上开发了空间扩展模块以支持空间数据的存储和操作。空间数据库引擎是与 GIS 平台或者数据库紧密结合的,在体系结构上必然存在差异,本节提取三种典型结构并以 Oracle Spatial、ArcSDE 和 SuperMap SDX+ 为代表进行对比

分析。

从空间数据库引擎和 DBMS 与应用程序结合的紧密程度来看,可以将空间数据库引擎的体系结构分为 DBMS 内置方式、中间件方式和客户端方式,本节分别以 Oracle Spatial、ArcSDE 和 SuperMap SDX+ 为例进行对比(图 1-1)。

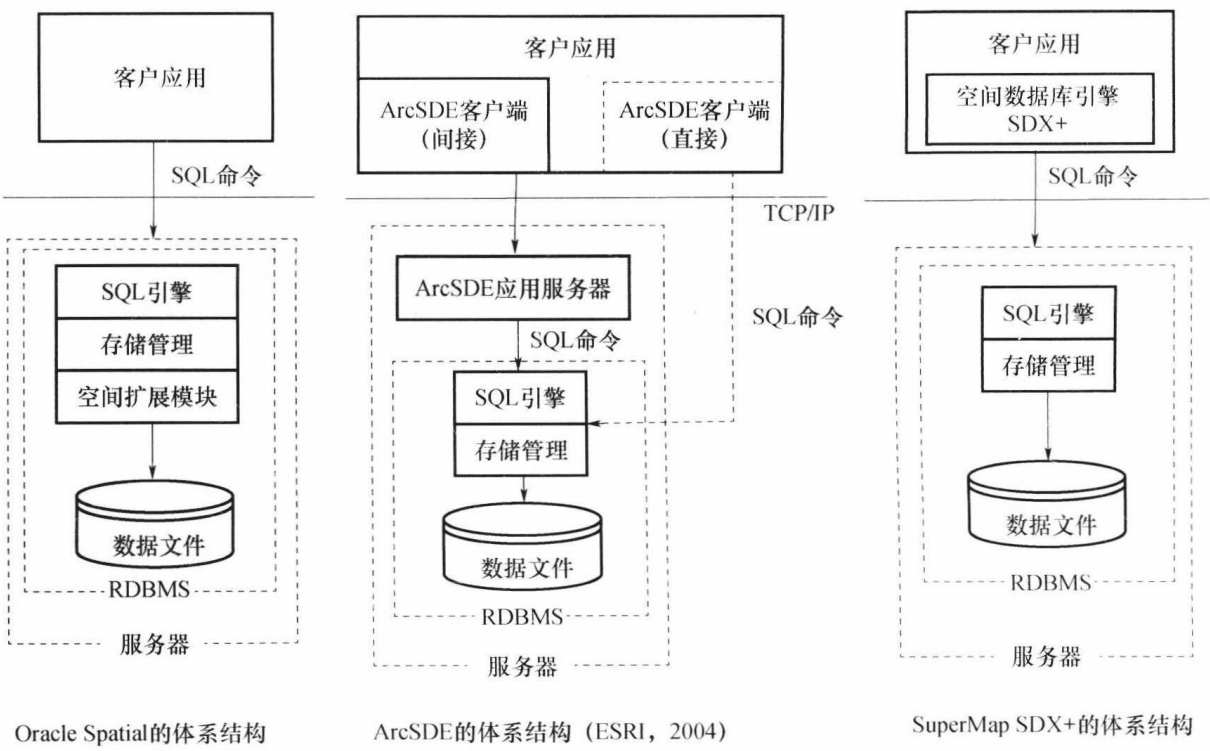


图 1-1 空间数据库引擎的体系结构对比

Fig. 1-1 Comparison of Spatial Database Engine Architectures

(1)DBMS 内置模式

DBMS 内置模式见图 1-1,它是在数据库内核进行空间模块的扩展,定义并实现了一套空间数据模型和空间 SQL,使得数据库本身支持空间类型数据的存储、查询和分析操作。一般由数据库厂商实现。Oracle Spatial 是 Oracle 数据库的空间数据扩展模块,它在数据库管理系统中提供了开放式的体系结构来管理空间数据,并提供了对空间数据的索引、查询和分析功能,采用的开放式空间数据存储模型得到了多家 GIS 平台软件的兼容支持,为不同 GIS 平台软件间的数据共享和交流提供了方便。

用户可以直接通过 SQL 语句来定义和操纵空间数据,可以直接使用 Oracle 数据库本身提供的完整性、可恢复性和安全机制,并充分利用服务器的处理能力。

但 Oracle Spatial 在栅格和影像数据的管理能力上存在着不足,直到 Oracle

10g 版本才开始提供栅格数据存储能力且没有提供有效的压缩功能,难以满足海量影像/栅格数据存储和管理的需求。此外,Oracle Spatial 提供的空间数据操作算子和空间分析方法有限,空间索引采用 R 树和四叉树,尽管在空间查询时采用了二级过滤模式,查询效率也不高。

在数据访问接口方面,由于缺乏面向对象的支持完整 GIS 功能的开发接口,客户端应用开发的难度很大。

### (2) 三层结构模式

ArcSDE 是典型的三层结构模式,其体系结构见图 1-1。它是美国著名的地理信息研究机构 ESRI 公司自 1995 年推出的空间数据库引擎,它在现有的关系数据库或对象—关系型数据库管理系统的基础上进行空间扩展,实现了基于 Oracle、DB2、SQL Server 等数据库管理系统的空间数据管理与操作。

ArcSDE 服务器一般需要与数据库绑定,为数据库解释空间数据,把客户端对数据的请求转换为数据库识别的 SQL 语句,将结果数据在服务器端缓存并返回给客户端。像所有数据库应用一样,ArcSDE 服务器必须通过 RDBMS 服务器来访问数据库中的内容。

在数据的访问和开发接口方面,ArcSDE 提供了功能完备的 GIS 应用客户端(ArcMap、ArcCatalog、ArcIMS 等)和丰富的二次开发手段(SDE C API、SDE Java API、Arc Objects),但 ArcSDE 没有实现在数据库端的空间数据结构化查询语言的扩展,不能通过 SQL 语句来直接访问和操作空间数据。

### (3) 两层结构模式

SuperMap SDX+ 采用的是两层结构模式,其体系结构见图 1-1。它可以支持 Oracle、SQL Server、Sybase、DB2、KingBase、DM 等多种数据库服务系统,并可以在多种操作系统之上运行,包括国产化的红旗 Linux 等操作系统。

SuperMap SDX+ 可以将空间数据、索引数据和属性数据存储的关系数据库的一张连续表中,实现了空间数据与业务属性数据的一体化存储和管理。支持丰富的空间数据类型的存储和管理,既包括常规的 GIS 数据对象,也包括 CAD 参数化对象、拓扑模型以及大数据量的遥感影像数据和格网数据。

SuperMap SDX+ 不需要与数据库绑定,完全作为数据库的客户端,空间操作和空间分析数据需要从数据库服务器取到本地,在本地内存中进行计算最后获得结果数据。处理海量数据时需要将数据取到本地,使得网络带宽将成为性能瓶颈