

基于 Oracle DBMS 的森林资源空间数据库存储优化策略

张茂震¹, 唐小明², 王亚欣², 张怀清², 吴东亮³

(1. 浙江林学院 环境科技学院, 浙江 临安 311300; 2. 中国林业科学研究院 资源信息所, 北京 100091; 3. 北京联合大学 旅游学院, 北京 100101)

摘要: 针对当前森林资源数据库应用中海量数据和集中式管理等因素所导致的系统效率问题, 在分析 ArcSDE 的空间数据存储管理机制的基础上, 结合应用系统开发, 从数据库设计、Oracle DBMS 配置、ArcSDE 配置和数据维护等 4 个方面总结了基于 Oracle DBMS 的空间数据库系统优化策略。以福建省省级森林资源空间数据库系统为例, 就主要优化策略进行了验证。结果表明: 合理调整数据库管理系统 (DBMS) 和 ArcSDE 参数 (Oracle SGA 区大小和 ArcSDE 索引格网大小, MAXBUFSIZE 大小), 优化数据库结构设计, 采用中间表技术等都可以有效提高查询效率; 优化森林资源数据库设计必须考虑合理的逻辑设计和分散的物理设计相结合。图 5 表 1 参 12

关键词: 森林经理学; ArcSDE; Oracle DBMS (数据库管理系统); 空间数据库; 系统优化

中图分类号: S757 文献标志码: A 文章编号: 1000-5692(2008)04-0411-06

Oracle DBMS based forest resources spatial database storage optimization strategy

ZHANG Mao-zhen¹, TANG Xiao-ming², WANG Ya-xin², ZHANG Huai-qing², WU Dong-liang³

(1. School of Environmental Science and Technology, Zhejiang Forestry College, Lin'an 311300, Zhejiang, China; 2. Research Institute of Resources Information, The Chinese Academy of Forestry, Beijing 100091, China; 3. School of Tourism, Beijing Union University, Beijing 100101, China)

Abstract: To cope with the system efficiency problems caused by mass-data and the centralized management in the forest resources database, the paper analyzed the spatial data management mechanism of ArcSDE spatial engine and worked out the optimization strategies for Oracle DBMS based spatial database system from four perspectives including database design, DBMS configuration, ArcSDE configuration and database maintenance. The provincial level forest resources spatial database of Fujian Province was taken as an example to validate the main optimization strategies. The results indicated that the rational adjustment of the parameters of DBMS and ArcSDE engine (Oracle SGA size, ArcSDE index size and MAXBUFSIZE), optimization of the database design and adoption of the middle table technique could all improve the query efficiency significantly. For optimizing forest resources database, efficient logical database structure and distributed physical database structure should both be considered. [Ch, 5 fig. 1 tab. 12 ref.]

Key words: forest management; ArcSDE; Oracle DBMS; spatial database; system optimization

在森林资源管理应用中, 空间数据库系统都面临着由于海量数据而导致的系统效率问题^[1]。随着森林资源监测内容的不断增加和集中式空间数据管理模式的广泛应用, 森林资源数据库规模空前扩张, 应用需求不断升级, 森林资源数据库系统效率问题日益突出^[2]。因此, 从 ArcSDE 的存储分析入手, 进行旨在提高数据库系统效率的研究具有重要现实意义。数据库系统优化的方法包括应用程序设计优化、SQL 语句优化、内存使用优化、数据库设计优化等多个方面^[3-6]。目前, 有关研究较多地集中在

收稿日期: 2007-09-25; 修回日期: 2008-01-07

基金项目: “十一五”国家科技支撑项目 (2006BAD23B0204-4); 浙江林学院科学研究发展基金资助项目 (2006FR058)

作者简介: 张茂震, 副教授, 博士, 从事森林资源信息管理等研究。E-mail: zhang@mail.jhptt.zj.cn

属性数据的 SQL 语句优化^[7]，以内存参数调整为主的空间数据库优化，以及如 CLOB 等特殊存取方式的优化^[8]。对于森林资源空间数据库，其优化研究相对缺乏。森林资源管理数据库应用系统的数据一般都集中式管理，无论是 C/S 或 B/S 体系结构，数据存储在服务器上，所有对数据请求的响应都在服务器端完成。因此，优化服务器端的应用是提高响应速度的关键。本文从优化空间数据库数据存储出发，从 Oracle DBMS(数据库管理系统)和 ArcSDE 的功能、特点等方面探讨基于 Oracle + ArcSDE 的森林资源数据库应用系统优化方法，并以福建省森林资源数据库为例，对优化策略进行验证。

1 空间数据存取中间件——ArcSDE

1.1 ArcSDE 的构成

ArcSDE 是一种中间件技术。它在地理信息系统(GIS)应用程序和基于关系型数据库管理系统(RDBMS)的空间数据库间提供了一个开发接口^[9]。从总体上看，ArcSDE 体系结构是一种利用关系型数据库(RDB)存储及管理空间数据和属性数据的数据库应用系统的基本构架。ArcSDE 采用客户/服务器(C/S)体系结构，是位于 RDBMS 上层一个层次的服务程序，为客户的请求提供空间数据的查询访问和分析服务。ArcSDE 应用程序接口为用户提供标准空间查询和分析函数，客户端通过应用程序接口(API)访问 ArcSDE 的这些函数和服务。

1.2 ArcSDE 的工作原理

所谓中间件就是处于操作系统与应用软件之间的一类能为处于自己上层的应用软件提供运行与开发环境和实现互操作的软件。ArcSDE 属于中间件技术，其本身并不能够存储空间数据，它的作用可以理解为数据库的“空间扩展”。

在基于 Oracle 的 ArcSDE 空间数据库中，ArcSDE 保存了一个被称为资料档案库的 Oracle 表空间，其中包括空间数据字典和 ArcSDE 软件包等一系列 Oracle 对象，用于管理空间信息。空间数据引擎(SDE)用户负责 ArcSDE 与 Oracle DBMS 交互，通过维护 SDE 模式下的空间数据字典及运行程序包来保证空间数据库的读写一致性。通过 Oracle 服务端程序，ArcSDE 使用 going 进程与数据库交互。每个 ArcSDE 服务都有一个 going 进程在监听用户的连接请求，清理断开的用户进程。每个连接 ArcSDE 的客户端应用程序都被指定一个 graver 进程，该进程由 going 生成。通过 Oracle 服务端程序，graver 提交用户所有的数据库查询及编辑的请求。在操作系统中，Oracle 将对象逻辑地存储在一个或多个数据文件所支撑的表空间内。对于不同类型的数据，在 Oracle 中一般都分配给它们各自独立的表空间^[10]。

传统的 ArcSDE 应用服务器连接(application-server connection)的工作方式，是 ArcSDE 服务器进程(giomgr)分配名为 gsvr 的进程来全面负责客户机与服务器的元数据通信^[11]。ArcSDE 8.1 版本之后，新增了直接连接方式(direction-connection)。在此连接方式中，gsvr 进程功能嵌入到客户机连接应用程序中。此种方式下，gsvr 的功能由客户端连接应用程序完成。ArcSDE 的工作机制如图 1 所示。

1.3 ArcSDE 中的空间数据存储管理

SDE 存储和组织模式是将空间数据类型加到关系数据库中，不改变和影响现有的数据库应用。它用在数据表中加入一个几何数据列(shape column)的方式实现对空间数据和属性数据的统一管理。SDE 将空间数据和空间索引放在不同的数据表中，通过关键项将其相联。

SDE 通过将信息存入层表(layers table)来管理空间可用表^[12]。层表为 ArcSDE 的元数据，对应于每一个图层，其中 Table_name 字段存储业务表的名称。

对于每一个图层，SDE 建立 3 个表(业务表、要素表、空间索引表)实现对空间数据的存储和

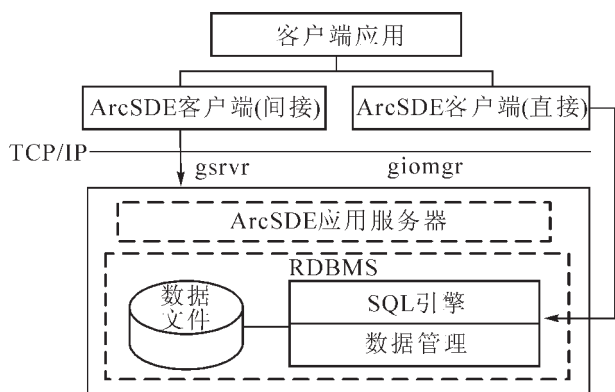


图 1 ArcSDE 工作机制
Figure 1 The mechanism of ArcSDE

管理。这些表之间以及与图层表之间的关系如图 2 所示。业务表(business table, 即 B 表)。业务表通过向一个已存在的关系数据库表中加入几何数据项, 使该表成为空间可用 (spatially enabled)。几何数据项是用来唯一标识一个几何体的整型数据。这个整型值为要素标志(ID)号, 业务表将通过该 ID 号与要素表、空间索引表关联。要素表(feature table)。要素表也称特征表或 F 表, 它存储每一地物的几何特征值。要素表采用 BLOB 二进制形式来存储地物要素的几何形状(shape)。要素表包括要素标志符(fid)、shape 的类型及 shape 的封装边界。要素表由空间字段层号的编码 F(Layer_ID) 来命名。业务表和要素表通过 fid 连接。空间索引表(spatial index table)。空间索引表(简称索引表或 S 表)是存储对落在一个规则格网内的几何体的引用。它由空间项层数来识别, 用 S(layer_id) 来命名。格网单元可以随时进行修改, 最佳格网单元的大小受平均封装边界大小的影响, 格网单元不能小于要素封装边界平均大小, 否则导致索引表中很多的格网单元都索引出相同的要素边界, 使索引表变大, 查询时间变长。

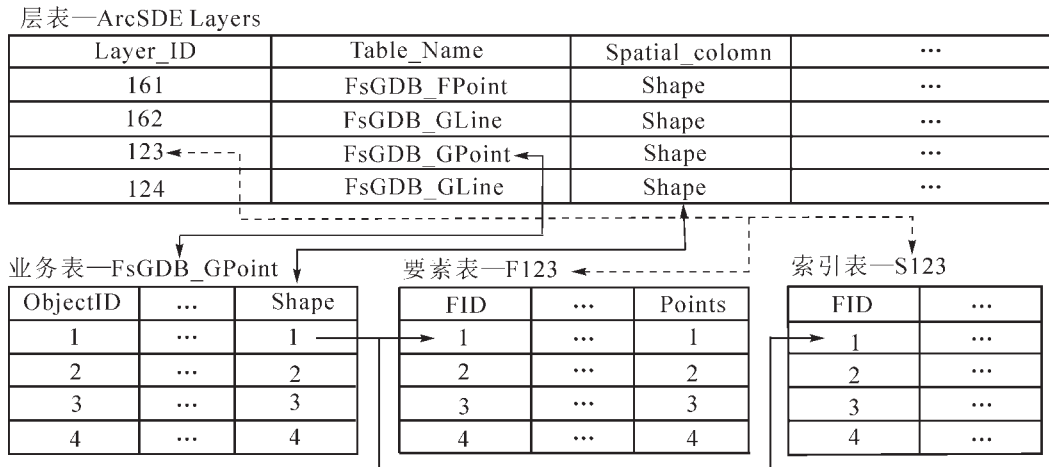


图 2 ArcSDE 3 个表之间的关系

Figure 2 The relations between the tables of business, features and spatial index for ArcSDE

2 空间数据存储优化

2.1 数据库设计优化

2.1.1 合理的逻辑组织, 分散的物理存储 当 Oracle 通过访问索引来定位时, 它必须同时访问索引表和索引所指向的数据表。如果表和索引存放在同一个物理磁盘上, 磁头需不断地来回移动, 导致 I/O 性能下降。如果将表和索引分开存放在不同的物理硬盘上, 访问时就有 2 个不同的磁头参与定位, 节省了定位时间。因此, 应尽量把索引和表分别存储在不同的表空间中, 而且这些表空间分别对应于在不同物理硬盘上的数据文件。一般来说, 数据性质类似、访问类型相同应在一个表空间; 不同类型数据, 应分别拥有各自独立的表空间。但是, 数据量和访问量也是重要的参考指标, 需要综合考虑。

2.1.2 设计索引 由于空间数据量大, 其索引表如何存储也对系统效率有重要影响。一般来说, 索引表空间要独立于其他表空间, 即索引表空间仅用于索引存储和处理。如果并发用户不多, 可将所有索引放在一个索引表空间。而当并发用户多、数据访问量大的情况下, 必须为每个类型的数据/每个数据表空间设立一个对应的索引表空间, 以获得更高的效率。

2.2 DBMS 配置优化

Oracle 初始化参数在参数文件 initorcl.ora(“init”后面 4 个字符对应于系统的 SID) 位于 \$ORACLE_BASE/admin/oradb/pfile。合理配置其主要参数, 可取得较好的优化效果。可考虑调整的主要参数有以下几个。

2.2.1 db_block_buffers 缓冲区命中率(buffer hit ratio)是指用户进程直接从内存读取数据与从数据文件中读取数据的次数之比。它是衡量数据库性能的重要指标, 直接受参数 db_block_buffers 的影响。

缓冲区大小以 db_block 为单位, 其值不应超过 1/4 内存。Hit Ratio 值达到 95%以上就可以认为是非常理想的, 但至少不能低于 90%。通过下 SQL 语句和计算可得 Hit Ratio 值:

```
select name, value from v$sysstat where name in( 'db blockgets ', 'consistent gets ', 'physical reads ');
Hit Ratio = 1 - ( physical reads/( db block gets + consistent gets ) )。
```

在 Oracle9i 中增加了参数 db_cache_size, 系统默认值是 48 MB。为了提高系统性能, 可适当提高这一参数的值。

2.2.2 shared_pool_size 共享池包含共享游标、存储过程、控制结构和并行执行消息缓冲区等对象, 适当增加其值(以字节为单位)可改善多用户系统的性能。其值不超过 1/4 内存, Cache 命中率达到 95%以上就可认为已达到优化。以下语句用于获取相关参数:

```
select ( sum( pins-reloads ) ) / sum( pins ) 'Lib Cache ' from v$librarycache;
select ( sum( gets-getmisses-usage-fixed ) ) / sum( gets ) 'RowCache ' from v$rowcache;
select * from v$sgastat where name = 'free memory '。
```

2.2.3 sort_area_size 这是当查询需要排序的时候, 数据库会话将使用这部分内存进行排序, 当内存大小不足的时候, 使用临时表空间进行磁盘排序。这个参数影响了“基于代价的优化”做出的执行索引访问还是执行对结果集的排序的决定。这个值至少应大于或等于需要排序数据大小的平方根, 即 sort_area_size 的平方不小于需要排序数据的记录数。整个 SGA 区占内存的比例一般为 1/3 至 1/2。

2.3 ArcSDE 配置优化

2.3.1 精度值 在 ArcSDE 中, 精度值是影响数据存储和检索效率的 3 个主要参数之一。Geodatabase 使用整数来存储空间数据以取得更高的效率, 而空间数据基本上都为浮点型, 因此, 在两者之间转换时需要一个精度控制。精度值是在创建数据的时候指定的, 选择小的精度值会导致几何信息损失, 而选择大的精度值则会使原始坐标值放大过多, 浪费了存储空间, 而使得数据库存储效率下降, 系统性能降低。精度值必须根据数据的实际情况确定, 若知道数据的小数点后两位为有效数值, 则 precision 应该选择 100, 选择 1 会导致 2 位有效数值丢失, 选择 10 000 则多保留了 2 位无效数据。一般情况下, Geodatabase 使用最大 32 bit 来存储正型数值。其中 1 bit 位用于存储符号, 另外 31 bit 用于存储数值。因此, 在设置精度值时, 必须考虑空间域与精度值, 满足以下 2 个条件: $(maxx - minx) * precision$ 小于或约等于 2 的 31 次方; $(maxy - miny) * precision$ 小于或约等于 2 的 31 次方。

2.3.2 空间网格(spatial grid) 格网(grid)是 ArcSDE 对图层进行逻辑区划的单位, ArcSDE 将整个图层划分为多个规则的格网单元(cell)来建立空间索引。通过建立合适格网大小的空间索引可以大幅度提高系统的响应速度。格网单元的大小随不同图层而改变。SDE 允许划分最多为 3 级的格网空间, 由低到高格网单元的尺寸逐渐增大。理论上选择网格大小的原则是使得一个索引网格里面只有一个要素, 但同时又不能因过度强调以上原则而造成过多的网格。空间索引使得在用空间过滤器进行查询的时候查询变得更快。权衡每格网的要素数量和每要素所占格网数量, 索引格网单元大小选择的标准是让 S 表尽可能地小。通常以要素封装边界平均大小的 3 倍来设置初始的格网单元大小, 然后在进一步调整。

2.3.3 传输缓冲参数(transport buffer parameters) 用户可以通过 ArcMap 和 ArcCatalog 或其他应用程序连接 ArcSDE 应用服务, 访问数据库中的要素类和表。每次访问时, 就会产生 1 个 ArcSDE 的“流”(stream), 这是客户端通过 ArcSDE 连接数据库的数据传输机制。当 ArcSDE 的“流”产生后, ArcSDE 服务器会给客户端和服务端分配缓冲。传输缓冲减少了 I/O 操作, 并且在通过网络传输的时候是把多条记录批量(数据块)传输而不是单个记录传输, 从而提高了性能。控制传输缓冲的有 3 个参数: MAXBUFSIZE(内存分配给每个传输缓冲的总大小)、MINBUFSIZE(每个传输缓冲的最小值)、MINBUFOBJECTS(每个传输缓存的最小记录数)。调整原则是: MAXBUFSIZE * 用户数小于物理内存, MINBUFSIZE 不要超过最大值(MAXBUFSIZE)的一半, 一般 1/3 为佳, 否则数据只有达到最小值才发送, 会增加反应时间。

2.4 数据维护

及时有效地数据库系统维护也有助于查询效率的提高。主要包括以下 3 个方面：采用中间表技术。中间表技术即随用户查询兴趣而动态地预先生成的最直满足查询要求的临时表，可以大幅度节省查询开销。当一个需求固定以后，其作为数据源的临时表也相对固定。在数据更新以后，需要定期进行索引重建。定期进行数据库统计信息更新，以保证查询分析器能够正确分析查询代价，选出最优的执行方案。

3 效果与评价

3.1 测试环境

服务器: Xeon 3.0 G, 4 G RAM, Double 160 G Hard Disk, Windows 2003; 客户机: P4 2.4 G , 512 M RAM, Double 120 G Hard Disk, Windows XP。测试数据: 福建省全省二类森林资源调查数据(2000 年), 1 843 173 个小班。

3.2 测试结果分析

3.2.1 参数调整 测试使用语句:

```
select features from fjls_d_p_xb_10k_03
that completely contain the features in this layer fjls_d_l_ri_4m.
```

调整内存。调整对查询速度影响较大的参数高速缓存，分别将 Oracle 的高速缓存设为 8, 16, 24, 40, 64 MB。不同高速缓存值的查询时间的变化趋势如图 3。图 3 表明，高速缓存值由 8 MB 增为 16 MB 时，时间变化较明显，随着高速缓存值的增加，曲线逐渐趋于平缓。高速缓存应在数据库系统运行过程中根据需要来调整，该系统测试条件下 40 MB 即可满足需要。调整 spatial grid。福建省小班图层包含 1 843 173 个要素。经计算这些要素的平均封装边界为 347(近似值)，将 spatial grid 的值设为平均封装边界的 3 倍即 1 040。分别将 spatial grid 的值设为 1 040, 2 000, 3 000, 4 000。对应于不同 spatial grid 值的查询时间趋势如图 4。图 4 显示，索引格网值过大使每个格网包含的要素过多会降低查询的效率，所以应设置合理的值，一般设为平均封装边界的 3 倍。调整 Transport buffer parameters, MAXBUFSIZE 的值分别设为 31072BYTES, 51072BYTES, 131072BYTES 和 201072BYTES, 不同的 MAXBUFSIZE 值的查询时间变化趋势如图 5。图 5 表明，当 MAXBUFSIZE 值增大，查询时间减少。但要把 MAXBUFSIZE 的值增大，要考虑服务器的内存。每增加一个用户连接，默认的 MAXBUFSIZE 值每个流会增加 64 kB，如：100 个用户调用 7 个要素类，ArcSDE 会分配 44 800 kB (100 × 7 × 64) 内存给服务器的传输缓冲，所以过大的 MAXBUFSIZE 值会迅速耗尽服务器的物理内存。

3.2.2 数据维护 数据维护与数据存储处理关系密切，这里仅对采用中间表技术进行测试。将福建省二类调查统计查询语句分类，按分类语句的最小粒度将查询结果存入中间表，供报表统计查询，大幅

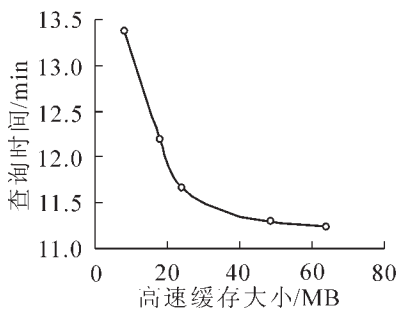


图 3 不同高速缓存值的查询时间

Figure 3 The response time and cache size

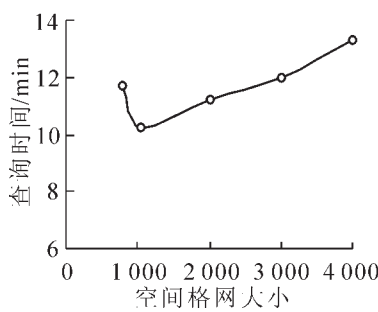


图 4 不同格网值的查询时间

Figure 4 The response time and grid size

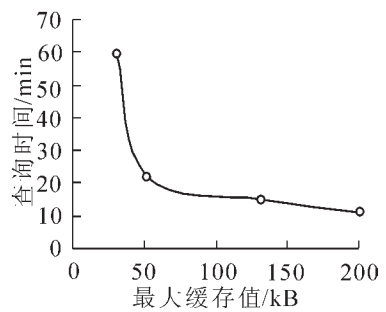


图 5 不同 MAXBUFSIZE 值的查询时间

Figure 5 The response time and MAXBUFSIZE size

表 1 森林资源数据库系统中数据预处理对查询效率的影响
Table 1 The affects on query efficiency by data pre-processing of forest resources database

原始数据名称	查询目标	查询分组因子	查询处理方式	响应时间/s
小班基本数据	面积和蓄积	权属, 林种, 优势树种	预处理结果作为数据源*	0.10
小班林分数据	面积	地类, 权属	预处理结果作为数据源**	0.06
小班基本数据	面积和蓄积	权属, 林种, 优势树种	直接使用原始数据	245.02
小班林分数据	面积	地类, 权属	直接使用原始数据	201.00

说明: * 以 select 县, 权属, 林种, 优势树种, sum(面积), sum(蓄积) group by 县, 权属, 林种, 优势树种的查询结果作为目标查询的数据源。** 以 select 县, 权属, 地类, sum(面积), sum(蓄积) group by 县, 权属, 地类的查询结果作为目标查询的数据源。

度提高了查询效率。表 1 的结果显示, 对于林分数据, 同样是查询按权属和地类分布的林分面积, 从中间表查询和直接从原始数据表查询的效率相差最大达 3 000 倍以上。

4 结论与讨论

基于存储的空间数据库查询优化主要有两方面的工作: 系统优化和数据库设计优化。系统优化是对后台 RDBMS 和 ArcSDE 进行调整。对于 Oracle 后台数据库管理系统, 有效的优化措施包括调整 SGA 区大小, 改变数据文件和日志文件的物理存储路径, 修改 Oracle 数据库启动参数等。对于空间数据环境 ArcSDE, 其优化措施包括有关精度值、索引网格大小和传输缓冲块大小的调整等。实践证明, Oracle 的 SGA 区大小、ArcSDE 的索引网格大小、MAXBUFSIZE 等是影响查询和整个系统性能的主要因素, 这些参数的调整, 对系统效率提升的作用显著。在数据维护中动态地创建于用户查询目标一致的中间表, 作为用户查询数据源可以极大地提高查询效率。数据库设计优化中合理的逻辑组织、分散的物理组织以及有效的索引策略是提高查询效率的基本策略。

空间数据库应用系统优化是一个综合性的复杂工程, 涉及 DBMS、空间数据环境、应用程序和网络等多方面, 本文仅讨论其中的一个主要方面——与 DBMS 相关的存储方面的问题, 如何从其他方面挖掘潜力、提高系统效率仍有待于进一步研究。

参考文献:

- [1] 张茂震. 森林资源数据库查询优化策略与技术研究[D]. 北京: 北京林业大学, 2006.
- [2] 庄晨辉, 陈铭潮, 潘俊忠. 林业资源管理地理信息系统的研制与开发[J]. 福建林学院学报, 2005, 25 (1): 1 - 4.
- [3] 熊丽华, 杨峰. 基于 ArcSDE 的空间数据库[J]. 计算机应用, 2004, 24 (3): 90 - 96.
- [4] 王亚欣, 唐小明, 张茂震, 等. 基于 SQL 的森林资源数据库查询优化[J]. 农业网络信息, 2006 (5): 81 - 84.
- [5] 赵村民, 宋利好, 赵晓民. 基于 Oracle 与 ArcSDE 的空间信息访问优化[J]. 吉林大学学报: 信息科学版, 2004, 22 (3): 283 - 288.
- [6] 杜庆峰, 张卫山. Oracle 的中大型应用系统性能优化分析[J]. 计算机工程, 2005, 31 (14): 91 - 93.
- [7] 成典勤, 崔杜武. J2EE 架构下数据库访问的性能优化[J]. 计算机系统应用, 2006 (4): 64 - 67.
- [8] 聂红梅, 赵建民. Oracle 数据库中 Clob 大字段的查询优化技术研究[J]. 计算机技术与发展, 2006, 16 (8): 97 - 99.
- [9] 喻冰春. 基于 ArcSDE 的省级基础空间数据库管理系统的设计与实现——以广东基础库为例[D]. 长春: 吉林大学, 2006.
- [10] 王旭红, 周明全, 陈燕. 基于 Oracle 和 Arcsde 分布式空间数据库的设计与建立[J]. 西北大学学报: 自然科学版, 2004, 34 (2): 151 - 154.
- [11] 熊丽华, 杨峰. 基于 ArcSDE 的空间数据库技术的应用研究[J]. 计算机应用, 2004, 24 (3): 90 - 96.
- [12] 许捍卫, 冯学智. 空间数据存储机制研究[J]. 计算机应用研究, 2003(2): 39 - 40.