RESEARCH ON SPATIAL DATABASE DESIGN AND TUNING BASED ON ORACLE AND ARCSDE

Li Yingcheng, Li Ling

Chinese Academy of Surveying and Mapping, Beitaiping Road 16, Beijing, 100039, casmrsec@sina.com

Commission VI, WG VI/5

KEY WORDS: Database, Block, Configuration, GIS, Image

ABSTRACT:

With the development of the Database technology and Geographic Information System technology, the spatial data managing mode go through three stages, entirely File system management, File system and Relation Database System mixed management and entirely Database management. At the same time database model also develop from Relation Model, Object-Relation Model, to Object-oriented Model. At the present time there are two main spatial database management approaches using commercial database. One is using Spatial Data Engine (Such as ArcSDE) in the Relation Database Management System. The other is using Object-Relation Database. Whichever spatial database management approach is used, in practical project we need a set of technics flow about the data preparing, database design, database configuration and tuning. Especially in commercial database, the rational database design, configuration and tuning are more important.
In this article the experiment construct a fundamental spatial database system for Tibet province using Geodatabase model of the ArcSDE and the Oracle database management System. Concretely, the main fields are researched as following: (1) Analyse the character of fundamental spatial data including the Digital Orthophoto Map data, the Digital Elevation Modal data, the Digital Raster Graphic data and the Digital line Graphic data. And scheme out a data processing plan for spatial database construction. (2) Rationally Design Database, this process includes the design for Geodatabase model and the design for Oracle database. (3) Configure and tune Oracle database, this process includes the configuration of the table and index of the spatial object and the configuration and tuning of the database initial parameters.

1. INTRODUCTION

With the development of the RS (Remote Sensing), DP (Digital Photogrammetry) and GPS (Global Positioning System), the National Spatial Data Infrastructure constructing, the Digital Geo-Spatial Data Framework (spatial fundamental data, such as Digital Orthophoto Map, Digital Elevation Modal, Traffic data, Water System data and Administration Boundary data etc.) brought forward, the amount of available spatial data, especially the most fundamental and public datasets for researching and observing the earth is expanding continuously and the kinds of available spatial data are increasing frequently. Such a large amount of spatial data and their great demands make it is no time to delay to store and manage the spatial data efficiently, reasonably and safely. At present, the popular data management approach based on database system is applying to spatial data management. Whoever, the Geographic Information software producer, the Remote Sensing software producer and the Database software producer, provides his own storage and manage spatial data approach.

Database model develop from Relation Model, Object-Relation Model, to Object-oriented Model. And the main large commercial database manage systems, such as Oracle, DB2, Informix, SQL Server, Sybase etc., extend their capabilities for spatial data base on a certain database model. At present there are two main spatial database management approaches using commercial database. One is creating a new Spatial Data Engine, Such as ArcSDE, based on the Relation Database Manager System. It uses the internal large object data type, such as Long Raw or Blob of oracle database, to storage the spatial object. The other is directly using the object data type of the Object-Relation Database, such as the SDO_GEOMETRY object of oracle. Whichever spatial database management approach is used, in practical project we need a set of technics flow about the data preparing, database design, database configuration and tuning. Especially in commercial database, the rational database design, configuration and tuning are more important.

2. SPATIAL DATABASE MANAGEMENT APPROACHS BASE ON ARCSDE AND ORACLE

Oracle is one of large commercial database manage systems. When the research and evolution of Database technology entering the Object-oriented technology phase, Oracle Database becomes an Object-Relation Database. The great function and reliability of Relation Database and the flexibility and modeling function of Object Database make it easy to design, update and develop for expert application. The support and application of Oracle embody the character of Relation Database and Object-Relation Database both.

ArcSDE is a Spatial Database Engine product of ESRI. It use the middleware technology to store and manage the spatial data in the commercial database, such as Oracle, DB2, Informix, SQL Server, Sybase etc. Although there are differences between different Spatial Database Engine products, there are also similarities in spatial data management approaches. So in this article we make an example base on the ArcSDE and Oracle.

In Geodatabase, the object-oriented data model, of ArcSDE the real world is described as a set of data objects. ArcSDE models data objects in three basic ways: as a collection of discrete features in vector format, as a grid of cells with spectral or attribute data, or as a set of triangulated points modeling a

surface. [1]

2.1 ArcSDE Data Dictionary[2]

The ArcSDE data dictionary includes tables that maintain information about the feature classes and feature datasets.

2.2 Raster Dataset Management Approach [2]

ArcSDE tiles the pixels of the raster into blocks according to a user-defined dimension to make it easy to research on spatial analysis and application either in a special area or in a whole area. With a certain of compression method provided by ArcSDE, we can save disk space and decrease the I/O of network. The user-defined raster block and compression method determine the need of the disk space. At the same time, ArcSDE populates the raster blocks table according to a declining resolution pyramid. Spending more disk space it improves the display speed.

ArcSDE store the raster as a series of tables in a database. They are RASTER_COLUMNS Table, Business Table, Raster Table, Raster Band Table, Raster Band Auxiliary Table and Raster Block Table.

2.2.1 RASTER_COLUMNS: RASTER_COLUMNS table is an ArcSDE system table. It records the raster information about the database name in which the raster store, the business table name the raster belong to, The date the raster column was added to the business table etc.

2.2.2 Business Table: Business table is closed to the real world. It regards the raster as one object, and describes the attribute of this object. Business table contains the raster column. It is a foreign key reference to the raster table.

2.2.3 Raster Table: Raster table has a description and a bitmap set according to the characteristics of the raster. And raster_id is the primary key of the raster table and unique sequential identifier of each image stored in the raster table.

2.2.4 Raster Band Table: Each image referenced in a raster may be subdivided into one or more raster bands. The raster band table stores the raster bands information, such as the tile size, the original coordinate of the band, etc. Raster_id column of the raster band table is a foreign key reference to the raster table's raster_id primary key.

2.2.5 Raster Band Auxiliary Table: Raster band auxiliary table stores optional raster metadata such as the image color map, image statistics, and bitmasks used for image overlay and mosaicking. The rasterband_id column is a foreign key reference to the primary key of the raster band table.

2.2.6 Raster Block Table: Raster blocks table stores the actual pixel data of the raster images. The raster is tiled into a lot of blocks of pixels. One tile corresponds to one record of raster block table. The rasterband_ id column of the raster block table is a foreign key reference to the raster band table's primary key. A composite unique key is formed by combining the rasterband_id, rrd_factor, row_nbr and col_nbr columns.

ArcSDE realizes the Raster database Management. With the relation of the tables the description of the raster has its own integrity. At the same time, ArcSDE ensures the safety saving of raster dataset by an appropriate database backup and recovery procedure.

2. 3 Feature Dataset Management Approach[2]

ArcSDE uses a compressed binary format to store geometry in either an Oracle binary LONG RAW or BLOB data type. Arc-SDE stores compressed binary spatial data in the POINTS columns of the Feature Table. Compressing the geometry offers efficient storage and retrieval of spatial data by reducing the size of the geometry. ArcSDE verifies the geometry, compresses it and sends it to the Oracle instance, where it is inserted into a feature table in compressed binary format. Compressing the geometry on the client offloads the task from the ArcSDE server and reduces the transmission time to send the geometry to the ArcSDE server. Storing compressed geometry data reduces the space required to store data by as much as 40 percent.

A feature dataset is a collection of feature classes that share a common coordinate system. The simple feature class can be inside or outside of feature dataset. In order to describe the relationship of feature dataset and feature class ArcSDE uses a series of ArcSDE system table. The feature class is stored in a database as several tables too. They are GEOMETRY_ COLUMNS Table, Business Table, Feature Table and Spatial Table.

2.3.1 GEOMETRY_COLUMNS: GEOMETRY_ COLUMNS table is an ArcSDE system table. It records the feature class information about the database name in which the feature class store, the business table name the feature class belong to, The date the raster column was added to the business table, Geometry type of the feature class etc.

2.3.2 Business Table: The business table contains attributes and a spatial column. The spatial column is a key to the feature and spatial index tables. The relationship between the business table and the feature table is managed through the spatial column and the FID column.

2.3.3 Feature Table: The feature table contains the geometry attributes information of the feature entity, such as the type of geometric feature stored in the spatial column, the number of points defining the geometry, the envelope of the geometry etc. The feature table stores the geometry, annotation, and CAD in the POINTS column. The POINTS column may be defined as either LONG RAW or BLOB.

2.3.4 Spatial Table: The spatial index table defines the grid range and extent of all geometry in an ArcSDE feature class. It is an assistant table for Building a spatial grid index. The spatial index table contains two indexes. One index is on the SP_FID column, which contains the feature ID. The other is a composite index that includes all of the columns of the spatial index table. Since all of the columns of the spatial index table are indexed, the values of the table are read from the leaf blocks of the index and not the table data blocks. Therefore, when considering how to position the tables and indexes to reduce disk I/O contention, you should be concerned about the positioning of the indexes of the spatial index table but not the table itself.

3. SPATIAL DATABASE DESING

3.1 Spatial Data Preparing

The data of Digital Line Graphic (DLG), Digital Orthophoto Map (DOM), Digital Elevation Modal (DEM) and Digital Raster Graphic (DRG) are needed during our spatial database design.

3.1.1 Digital Elevation Modal (DEM): This source data spends disk space about 8.7G in GRID format of ARC/INFO. And its resolution is 25 meters. Its reference frame is 1980 Sian coordinate. Its map projection is Gauss-Kruger system dividing the globe into sixty zones.

3.1.2 Digital Orthophoto Map (DOM): This source data made from the fifth band, the fourth band and the third band of ETM in 1999 –2000 spends disk space about 5.69G in Raw format with .ifo information file. And its resolution is 30 meters. Its reference frame is 1954 Peking coordinate. Its map projection is Gauss-Kruger system dividing the globe into sixty zones.

3.1.3 Digital Raster Graphic (DRG): This source data made from relief map of china spends disk space about 3.15G in JPEG format with .ifo information file. And its resolution is 10 meters. Its reference frame is 1954 Peking coordinate. Its map projection is Gauss-Kruger system dividing the globe into sixty zones.

3.1.4 Digital Line Graphic (DLG): This source data spend disk space about 120.6M in Coverage format of ARC/INFO. Its reference frame is 1954 Peking coordinate.

The data source is different. In order to load the data into spatial database the data processing is needed. The detail of the process is show as Figure 1.
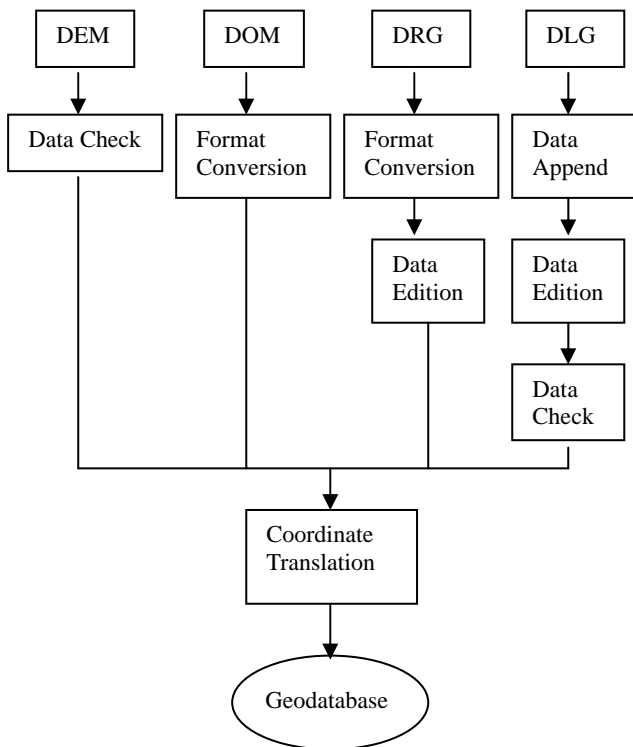

Figure 1. Spatial Data Preparing

Because the four kinds of data are standard products of survey, we mainly process data to make them have a uniform coordinate and their own format which can be accessed by Arc/Info.

3.2 Spatial Database Design Using Geodatabase Model

In Geodatabase, the object-oriented data model, the DEM, DOM and DRG spatial data are described as raster datasets. The DLG spatial data are described as feature dataset. It consists of feature classes, such as regionalism feature class, residential area feature class, water system feature class etc. The detail of the Design is show as Figure 2.
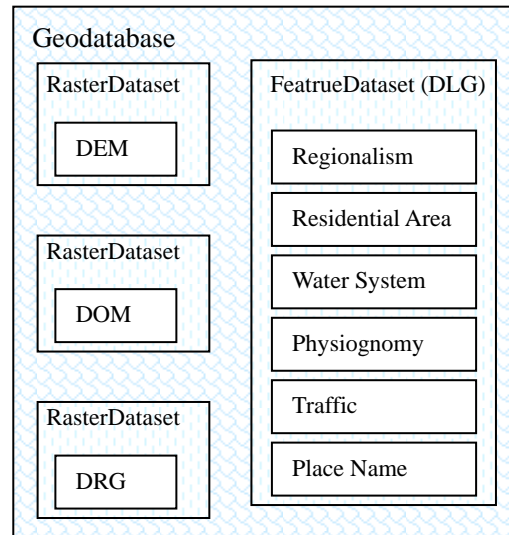

Figure 2. Spatial Database Design Using Geodatabase Model

3.3 Oracle Database Design[4]

3.3.1 Oracle Database Logical Design: The Standard Optimal Flexible Architecture can make logical objects distribute more efficiently in oracle database. It classes the logical object by the type and operation mode. This logical design makes the database administration and management of database growth Easier. And it can decrease Disk contention and safeguard against disk failures. Optimal Flexible Architecture defines the logical object category as SYSTEM Tablespace, DATA Tablespace, INDX Tablespace, TOOLS Tablespace, USERS Tablespace, TEMP Tablespace and RBS Tablespace.

Except the DATA Tablespace and USERS Tablespace we use the other tablespaces in our spatial database logical design, moreover, we add some other tablespaces.

- ArcSDE Tablespace stores the information of ArcSDE data dictionary.
- RASTERDERIVE Tablespace stores the Business Table, Raster Band Table and Raster Band Auxiliary Table of raster datasets which need few disk space.
- DEM Tablespace stores Raster Block Table of DEM raster dataset which need a large disk space.
- DOM Tablespace stores Raster Block Table of DOM raster dataset which need a large disk space.
- DRG Tablespace stores Raster Block Table of DRG raster dataset which need a large disk space.
- DLGF Tablespace stores Feature Table of DLG feature dataset.
- DLGS Tablespace stores Business Table and Spatial Table of DLG feature dataset.
- METADATA tablespace stores the metadata of the spatial database.

3.3.2 Oracle Database Physical Design: Oracle database physical design is a file distribution plan on different equipments. In order to accomplish physical design we must tell the I/O contention of data files, the I/O bottleneck of database files, the concurrent I/O process of the background process and the disk used for database. Define the safety and performance goal of the database, system hardware environment and image structure.

In this spatial database we give a reference of four disks physical design, Table 1. Its rationality has not been ensured.

Disk	Contents
1	The Tablespace Files of SYSTEM, SDE and TEMP, Control File, Redo Log File
2	The Tablespace Files of INDEXS and RBS, Control File, Redo Log File, Export Files
3	The Tablespace Files of DOM, DEM, DRG, DLG, METADATA and TOOLS, Control File, Redo Log File
4	Oracle Software Files, ArcSDE Software Files, Archive Log File, RASDERIVE Tablespace Files.

Table 1.   Four disks physical design

4. SPATIAL DATABASE CONFIGURATION AND T UNING

4.1 Table Object Configuration

Once the size of oracle data block, a basic unit on database operation, is determined, it can't be changed until recreate a new database. So it is important to define the size properly. As mentioned above, Raster Block Table needs a large disk space. In this table the spatial data is stored as long raw or blob data type. Large row size and rows accessed sequently need a large data block size. However the Business Table, Raster Band Table, Raster Band Auxiliary Table need few disk. Too large data block size will make inverse effect. The ARCSDE suggest that the least block is 8KB and the block size of 16KB has more efficient for polyline and polygon features. So here we use the 16KB data block size.

4.1.1Table Object of The Raster Dataset

4.1.1.1 Small Table Objects: The small table objects such as Business Table, Raster Table, Raster Band Table and Raster Band Auxiliary Table only need one data block. So the storage

parameter in the dbtune.sde file of ARCSDE is configured as following.

B_STORAGE   "PCTFREE 10 PCTUSED 90 INITRANS 4 TABLESPACE RASTERDERIVE STORAGE (INITIAL 16K NEXT 16K FREELISTS 4 MINEXTENTS 1 PCTINCREASE 0)"
RAS_STORAGE   "PCTFREE   10   PCTUSED   90 INITRANS 4 TABLESPACE RASTERDERIVE STORAGE (INITIAL 16K NEXT 16K MINEXTENTS 1 MAXEXTENTS 200 PCTINCREASE 0 FREELISTS 4)"
BND_STORAGE   "PCTFREE   10   PCTUSED   90 INITRANS 4 TABLESPACE RASTERDERIVE STORAGE (INITIAL 16K NEXT 16K MINEXTENTS 1 MAXEXTENTS 200 PCTINCREASE 0 FREELISTS 4)"
AUX_STORAGE   "PCTFREE   10   PCTUSED   90 INITRANS 4 TABLESPACE RASTERDERIVE STORAGE (INITIAL 16K NEXT 16K MINEXTENTS 1 MAXEXTENTS 200 PCTINCREASE 0 FREELISTS 4)"

4.1.1.2 Large Table Objects: The large table objects such as Raster Block Table need a great amount of data blocks. The amount of data blocks depends on the character of source data and the compress method used. And another factor is the user-defined dimension of the raster block in ArcSDE. We make an experiment such as table 2 and table 3.

Block Table Name	SDE_BLK_38	SDE_BLK_23	SDE_BLK_24	SDE_BLK_27
Raster Block Dimension	32*32	64*64	128*128	256*256
Avg. Row Size	2252	9041	33759	133469
Number of Rows	520778	130671	32889	8331
Number of Chaining Rows	0	0	30368	8187
Number of Data Block of Table	87699	130671	86744	72889
Number of Data Block of Index	952	-------	72	24

Table 2. The Experiment of the Raster Block Table

Block Table Name	The Ratio of Efficient Disk Space	Number of Data Block Per Row	Disk Space
SDE_BLK_38	81.6%	0.17	1.34G
SDE_BLK_23	60.5%	0.87	1.73G
SDE_BLK_24	78%	2.63	1.32G
SDE_BLK_27	93%	8.75	1.12G

Table3. The Analysis Result of the Experiment

Along with the Raster Block Dimension increasing the Number of Data Block of Table and the Number of Data Block of Index have a change in this experiment. At the same time the need of the disk space is different. So we must choose one proper Raster Block Dimension to save the raster data. Because of spending too much disk space Raster Block Dimension of 64*64 can't be used. Though the Raster Block Dimension of 256*256 needs small disk space. The great Number of Data Block Per Row will affect the performance of database operation. The Raster Block Dimension of 32*32 may be better. But too much rows will waste a lot of time on indexing. So the Raster Block Dimension of 128*128 is the better one in this experiment.

In dictionary-managed tablespace, in order to decrease the conflict among extents the proper number of the extents is 1000, and the maximum is 10000. The large table objects are not fit for the dictionary-managed tablespace. We usually use the tablespace that manages its extents locally to save the large table objects. The initial extent and next extent is in the same size defined by the tablespace. So the storage parameter in the dbtune.sde file is configured as following.

BLK_STORAGE   "PCTFREE 10 PCTUSED 90 INITRANS 4 STORAGE (MINEXTENTS 1 MAXEXTENTS 1000   PCTINCREASE 0 FREELISTS 4)"

4.1.2 Table Object of Feature Dataset

The table objects of the Feature datasets are smaller than the raster datasets' in this spatial database. So the dictionary-managed tablespace is used. According to the quantity of the data we define the initial extent and next extent size. For an example, the storage parameter in the dbtune.sde file is configured as following.

B_STORAGE   "PCTFREE 10 PCTUSED 90 INITRANS 4 TABLESPACE DLGS STORAGE (INITIAL 5M NEXT 400K FREELISTS 4 MINEXTENTS 1 PCTINCREASE 0)"
F_STORAGE   "PCTFREE 10 PCTUSED 90 INITRANS 4 TABLESPACE DLGF STORAGE (INITIAL 19200K NEXT 800K FREELISTS 4 MINEXTENTS 1 PCTINCREASE 0)"
S_STORAGE   "PCTFREE 10 PCTUSED 90 INITRANS 4 TABLESPACE DLGS STORAGE (INITIAL 9600K NEXT 800K FREELISTS 4 MINEXTENTS 1 PCTINCREASE 0)"

4.2 Oracle Database Configuring and Tuning[5]

The oracle and ArcSDE have a detail description about the database initial parameters, and give the reference. But we need to configure and tune the parameter for our use.

4.2.1 System Global Area (SGA)

In general the size of the SGA is no more than 2/3 physical memory of the server machine. SGA contains four parts, Database Buffer Cache, Redo Log Buffer, Shared Pool and Large Pool.

4.2.1.1 Shared Pool: Shared Pool consists of Data Dictionary Cache, Library Cache and User Global Area.

● Data Dictionary Cache keeps information about dictionary objects. Eventually, the database should reach a "steady state" in which the most frequently used dictionary data is in the cache. At this point, very few cache misses should occur. To tune the cache, make the ratio of getmisses and gets is less than 15% in the V$ROWCACHE view.

● Library Cache stores shared SQL and PL/SQL areas. If a user issues a statement that is already in the cache, the Oracle server can use the cached version without having to reparse it. To tune the cache, make the keywords gethitratio is more than 15% in the V$LIBRARYCACHE view. And if an execute call for a SQL statement is done and the shared SQL area containing the parsed representation of the statement has been deallocated from the library cache to make room for another statement or because the objects the statement refers to have been invalidated, the Oracle server implicitly reloads the statement and therefore reparses it. The number of reloads is counted for each of these namespaces. The keywords reloads should be less than 1% in the V$LIBRARYCACHE view.

● User Global Area keeps information about multithreaded connections. The total memory requirement for the multi-threaded server is no larger than if you use dedicated servers. You may need to increase SHARED_POOL _SIZE, but your private user memory is less.

In this spatial database the initial parameter shared_pool_size is 73400320 bytes. We do an analysis to assure actually size we used in share pool.

● The SHARABLE_MEM For stored objects such as packages and views is 26301236 bytes.

● The sharable_memery for SQL statements restricted by frequently issued statements is 7301304 bytes, though this will not include dynamic SQL.

● In a test environment, you can measure shareable memory by selecting the number of open cursors for a test user. You multiply the resulting value by the total number of users. The sharable memory for the user open cursor is 47500 bytes.

As mentioned above, the actually size of the total memory is 32M. Then plus 20%-30% of the actually size for unexpectedly use. The sharable memory of 41.6M is enough for this application.

4.2.1.2 Database Buffer Cache: The Buffer Cache holds copies of the data blocks from the data files. ARCSDE suggest the size of Database buffer cache is [3]:

DB_BLOCK_BUFFERS=

$$\frac{(\text{Physical RAM} \times (\frac{2}{3}) \text{ - shared\_pool\_size - log\_buffer}) \times (90\%)}{\text{db\_block\_size}}$$

To tune the cache, make the Buffer Cache Hit Ratio is greater than 90%.

Buffer Cache Hit Ratio =

$$1 - \frac{\text{physical reads}}{\text{db block gets + consistent gets}}$$

where   Physical Reads, Db Block Gets and Consistent Gets are the keywords in the V$LIBRARYCACHE view.

In this spatial database the initial parameter, shared_pool_size, is 13680.

4.2.1.3 Redo log buffer: The Oracle server processes copy redo entries from the user's memory space to the redo log buffer for each data manipulation language (DML) or data definition language (DDL) statement. The redo log buffer is a circular buffer. Thus, server processes can copy new entries over the entries in the redo log buffer that have been written to disk. LGWR normally writes fast enough to ensure that space is always available in the buffer for new entries. The SECONDS_IN_WAIT value of the Log Buffer Space event in the v$session_wait view indicates the time spent waiting for space in the redo log buffer because the log switch does not occur. The Redo Buffer Allocation Retries statistic in V$SYSSTAT view reflects the number of times a user process waits for space in the redo log buffer to copy new entries over the entries that have been written to disk.

In this spatial database the initial parameter, log_buffer, is 512000 bytes.

4.2.2 Program Global Area (PGA)

A Program Global Area (PGA) is a memory region containing data and control information for a single process (server or background). PGA initial size is fixed and operating-system specific. The initialization parameters OPEN_LINKS and DB_FILES affect the sizes of PGA.

4.2.3 Sort Area

As Oracle constructs an index, it writes the index blocks to the Sort Area. When the Sort Area becomes full, its contents are transferred to the temporary tablespace. The larger the sort area, the less frequently Oracle must write intermediate results to temporary disk space. You can determine if SORT_AREA_SIZE is large enough by comparing the value of sorts (memory) and sorts (disk) of the Oracle dynamic view V$SYSSTAT. If the sorts performed on disk are greater than 10 percent of those performed in memory, consider increasing the sort area.
In this spatial database the initial parameter, SORT_AREA_SIZE, is 65536 bytes

5. CONCLUSION

In practical project we need a set of technics flow about the data preparing, database design, database configuration and tuning. The technics flow is complex. And every step is important in the flow. When the database application takes a change, the database configuration may take a change too. So we need more experimentation to do it better.

REFERENCE

[1] Enviromental Systems Research institute, inc., 1999, *Modeling our world the ESRI Guide to Geodatabase design.* ESRI Press,
[2] Environmental Systems Research Institute, Inc., 1999, *Managing ArcSDE™ Services*, ESRI Press

[3] ESRI, 2002, *ArcSDE Configuration and Tuning Guide for Oracle*
[4] Kevin Loney, Li Jisong etc. Translate, 1998.10, *Oracle8i DBA Handbook*, China Machine Press
[5] Oracle Corporation, 1999.9, *Enterprise DBA Part 2: Performance and Tuning*