2011 International Conference on Computer Science and Information Technology (ICCSIT 2011)
IPCSIT vol. 51 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V51. 108

ArcSDE vector model analysis and vector SDE realization

Wang Yun-fei[a,*], Yan Hong-yu[b]

[a]Institute of Remote Sensing Application. Chinese Academy of Sciences, Beijing 100101, China
[b]China Aero Geophysical Survey & Remote Sensing Center for Land and Resources, Beijing 100083, China

Abstract. Spatial Database Engine (SDE) is a middleware technology. It manages spatial data in a relational database management system (RDBMS) and enables it to be accessed by clients. This paper mainly focuses on Vector Spatial Database Engine and do research on ArcSDE. ArcSDE vector model mainly includes four parts, which are vector data table relations, coordinate storage format, spatial index and query mechanism. Based on these, we realized a basic Vector Spatial Database Engine on oracle database.

Keywords: Spatial database, Spatial database engine, Spatial index

1. Spatial Database Technology

1.1. Spatial Database

Spatial database is a database that is optimized to store and query data that is related to objects in space, including points, lines and polygons. Traditional relational database management system is only designed for storing two-dimensional tables. So it is unable to store complicate objects, such as maps, images and so on. There are two storage patterns for spatial data. The first is dividing spatial data into 2 parts, spatial information and attribute information, store spatial information in particular file system and attribute information in RDBMS. This pattern is difficult to manage and maintain spatial data. The second is storing both spatial and attribute information in RDBMS by extending RDBMS functions. This is more efficient than first pattern.

1.2. Spatial Database Engine

SDE is a middleware technology. It extends RDBMS (such as Oracle, SQL Server, etc.) functions and enables spatial data storage and query in the database. From the perspective of logic structure, SDE is located between clients and database. SDE is integrated with database on the server side. It translates spatial operation instructions from client into standard SQL statements and executes these SQL statements in the RDBMS, then returns the results to the client [1] (Fig. 1).

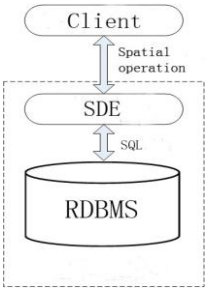

Figure 1. SDE functional structure

2. Arcsde Vector Data Model Analysis

ArcSDE [2, 3] is a spatial database engine produced and marketed by ESRI Company. It aims to enable the usage of RDBMS for spatial data. It facilitates storing and managing spatial data (raster, vector, and

* Corresponding author. Tel.:+86 010 95675268; fax: +86 010 95675268.
*E-mail address:*king1302217@yahoo.com.cn.

survey) in a RDBMS and makes the data available to many applications. This paper mainly focused on vector spatial data engine and does analysis on ArcSDE vector data model structure, which mainly has 4 parts:

1) Vector data table relations. It introduces the database tables used by ArcSDE to store vector data information.

2) Coordinate storage format. This part introduces the format used by ArcSDE to store feature (point, polyline and polygon) coordinates.

3) Spatial index. This part introduces how to create spatial index for a map and store spatial index in the relational database.

4) Spatial query. This part introduces how to query spatial data information from database using spatial index.

2.1. Vector data table relations

ArcSDE extends relational database to store spatial data. Essentially, it divides vector data into several two-dimensional tables so as to store it in database. ArcSDE vector model mainly contain 4 basic tables (Fig 2): Layers Table, Feature table, Spatial index table and the Business table.

The Layers table maintains data about each vector layer (feature class) in the database. It includes layer name, layer id, owner, spatial index grid cell sizes, envelope (MinX, MinY, MaxX, and MaxY) of the layer, feature type and Layer descriptions. Layers table relate with Feature table and Spatial index table through layer id field.

The Feature table stores the geometric shapes for each feature. This table is identified by the spatial column layer number, using the name F<layer_id>. Each record in Feature table represents a feature in the map layer. Feature table includes feature id, feature envelope, numofpts (the number of points defining the shape) and Points (contains the byte stream of point coordinates that define the shapes geometry).

The Spatial index table stores grid index information of the layer. This table is identified by the spatial column layer number, using the name S<layer_id>. It includes 7 fields, sp_fid (feature id), the grid cell coordinate (gx, gy) and the envelope of the shape (eminx, eminy, emaxx, emaxy).

The Business table stores attribute information of the layer.

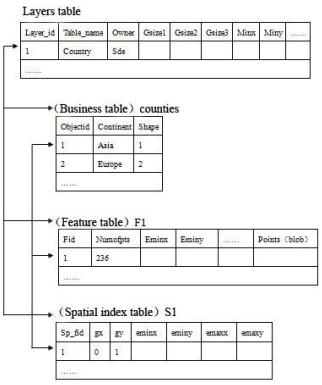

Figure 2. Vector table relations

Other tables in ArcSDE vector model are functional tables, for example version table, topological table, network table and so on. These tables are not included in our research.

2.2. Coordinate storage format

As we introduced above, feature coordinates is stored in binary stream format in “points” field of the Feature table. Coordinates are converted into byte stream according to X-Y order of the feature (X1/Y1，X2/Y2，...，XN/YN). In order to facilitate coordinates storage procedure and compress storage space, we use two steps to store coordinates.

1) Convert feature coordinates from double value into integer value

Internally, all coordinates in ArcSDE are 32-bit positive integers between 0 and 2147483647. This format provides better data accuracy, data integrity, and processing speed than real numbers. Because real-world coordinates are often neither positive nor integer, coordinate data requires an offset distance (a false origin) to ensure numbers are positive and a minimum resolution multiplier (called the scale) to convert real numbers to

integers. Offset distances are specified in the same units as the data. The scale can be any positive value up to 2147483645.

2) Compress coordinates binary stream

Within the binary stream, each of the x/y is compressed in a byte-order independent manner. Because x/y coordinates in one feature has little difference, we can compress binary stream by two steps. First, all values are converted to a relative-offset scheme, then each relative-offset value is packed into the minimum number of bytes required to represent the value. The goal of converting coordinate values to a relative offset scheme is to make the values as small as possible so that they require fewer bits to represent them. In an array of relative-offset values, the first value is an absolute value (stored as a 32-bit integer) while each subsequent value is the offset, or difference, from the previous absolute value. So the coordinates order in binary stream is (X1/Y1，△X2/△Y2，...，△XN/△YN). Therefore, given N absolute values, the relative-offset values are calculated by equations in Fig. 3:

$$\triangle X_2 = X_2 - X_1; \quad (1)$$
$$\triangle Y_2 = Y_2 - Y_1; \quad (2)$$
$$\ldots$$
$$\triangle X_N = X_N - X_{N-1}; \quad (3)$$
$$\triangle Y_N = Y_N - Y_{N-1}. \quad (4)$$

Figure 3. Equations for calculating relative-offset values

2.3. Spatial index

Spatial index is used to speed up spatial query. Common spatial indexes include grid spatial index [4], quad-tree index [5] and R-tree index [6]. Grid spatial index is widely used because it is simple to create, use and maintain. ArcSDE use multi-level grid indexes. The level count of grid index is decided by layer type and feature count in the layer. For example, point layer or layer with few features, grid index of one level is enough. In principle, grid index at most have 3 levels, and the grid cell size in next level must be 3 times lager than prior level.

There are two ways to generate grid index for a layer [7]. The first is to overlay the extent of each feature onto the lowest grid level to obtain the number of grid cells. If the feature exceeds four cells, promote the feature to the next grid level, if you have defined one. Then promote the feature until it fits within four cells or until the highest defined grid level is reached. On the highest defined grid level, geometries can be indexed by more than four grid cells. The second is to overlay the envelope of the each feature onto the grid. This way is more simple and efficient. But the disadvantage is grid index create by this method is not as precisely as first one. Take efficiency into account, second method is always used.

Next, we will introduce how to calculate the grid cell size and level count of the grid index. There are three layer types.

For point layer, we generally use grid index of one level. The grid cell size is calculated by (1):

$$\text{Gridsize} = 2 * \left((\text{MaxX} - \text{MinX}) + (\text{MaxY} - \text{MinY}) \right) / \text{PC} \quad (1)$$

MinX, MinY, MaxX and MaxY are the map boundary; PC is the point count in the layer.

For polyline and polygon layer, the grid cell size in first level of grid index is calculated by (2):

$$\text{Gridsize} = (\text{avgW} + \text{avgH}) / 2 \quad (2)$$

avgW and avgH, respectively, are average width and height of all features in the layer. If the feature envelope exceeds 4 grid cells, then we well create second level of grid index, which grid size is 4 times larger than first level, continue this step until we create the third grid index level.

We build a spatial index by applying a grid to the feature class. It records features that fall within each grid cell in an index table (the S table). A feature that falls in more than one cell is listed in each. Grid cells with no data are not included in the table. In order to distinguish grid cell coordinate (gx, gy) of each level in spatial index table, we add grid cell coordinate of second grid level with 20000000 and third grid level with 30000000. The spatial index table is shown in Fig. 4.

SP_FID	GX	GY
0	17	1
1	30000001	30000000
2	11	3
3	20000004	20000000
3	20000005	20000000
4	13	2
4	14	2
4	15	2
5	20000005	20000000
5	20000006	20000000

Figure 4. Spatial Index Table

It is easy to maintain grid index in database. Update the spatial index table when data in feature table is changed. For example, when you add a new feature into feature table, then new index records of this feature needs added into spatial index table.

2.4. Spatial index

This section we will introduce how to query data use spatial index described in section 2.3. Take spatial query by rectangle area as example, there are 5 steps to query spatial data. Using Fig. 5 to illustrate, there are 4 polygons in the layer, blue rectangle is the envelope of the polygon, red rectangle is query range, and the grid represents grid cells in grid index.

1) Draw a rectangle in the map as the query range, convert the coordinate of four rectangle vertices (upper-left, upper-right, lower-left, lower-right) from screen coordinate system to map coordinate system and get grid cell coordinate (grid cell coordinate gx, gy from (1,2) to (4,4)).

2) Search grid cell coordinate in the spatial index table, get the list of polygon id (polygon 1, polygon 2 and polygon 3 are in id list).

3) Classify polygons in id list, if the envelope of polygon is completely within the query rectangle (polygon 1); add the polygon id into result list.

4) Do precise spatial judgment for each polygon which remains in id list. If the polygon is intersect with the query rectangle (polygon 3), then add the polygon into result list.

5) Return the result list. Polygons in result list are the query results (polygon 1 and polygon 3).

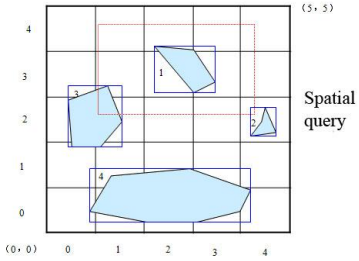

Figure 5. S Grid spatial query

3. Vector Spatial Database Engine Realization

Finally, using C# language, we develop a vector spatial database engine (SVSDE) based on Oracle database. It realized basic functions of vector spatial database engine and is able to store, manage and query vector data in Oracle database.

3.1. Vector data import and display

Take American counties data (counties.shp, shapefile format) as example, we use SVSDE to import counties.shp to Oracle and develop a client program using C# language and GDI [8] to display data. Fig. 6 and Fig. 7 show the information of counties data.

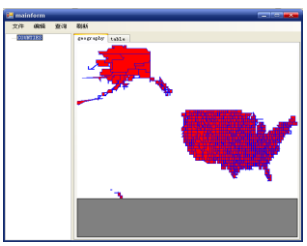

Figure 6. counties layer

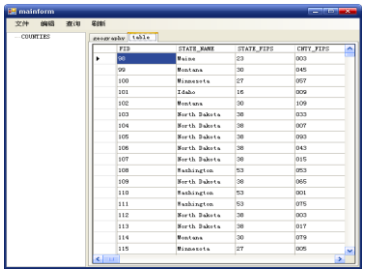

Figure 7. counties layer

3.2. Spatial query

We draw a rectangle on the client as query range. SVSDE return the query results from database. Attribute information of counties in query results is showed on the client (Fig. 8).

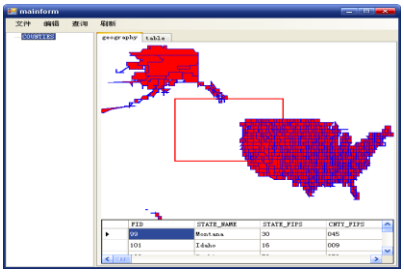

Figure 8. Spatial query

3.3. Efficiency comparison

At last, we compare the efficiency of data import between SVSDE and ArcSDE. ArcSDE is better than SVSDE in efficiency. One reason may be that ArcSDE utilized more key technologies. So the efficiency problem is the further research issue.

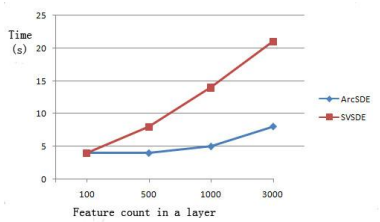

Figure 9. SVSDE and ArcSDE efficiency comparison

4. Conclusions And Outlook

In the paper, we analysis ArcSDE vector model and develop a vector spatial database engine based on Oracle (SVSDE). SVSDE realize basic functions of a vector spatial database engine, which can store, manage and query vector data. Other aspects such as version editing, topology building and engine efficiency etc. also needs further research and improvement.

5. References

[1] Wu Mengquan, Cui Weihong, Mei Xin. Design and Construction of Spatial Database Based on Spatial Database Engine [J]. Computer Engineering, 2007, 33(6): 48 - 50.

[2] ArcSDE Developer Help, ESRI, 2002.

[3] Understanding ArcSDE, ESRI, 2003

[4] Zhou Yong,He Jiannong,Tu Ping. An Algorithm of Improved Auto-selection Multi-layer Grid Spatial Index[J]. Computer Engineering and Applications, 2006.7: 159 - 161.

[5] Dong Peng Yang Chongjun Rui Xiaoping Gao Jiliang Algorithm of Spatial Select Query in GIS Based on the Improved Quadtree —— with the Case of ESRI SHAPE File [J]. Computer Engineering and Applications,2003: 58 - 61.

[6] A. Guttman. R-trees: A dynamic index structure for spatial searching [C]. In Proceedings of the International Conference of Management of Data (ACM SIGMOD), pages 47-57. ACM Press, 1984.

[7] Guo Jianzhong,Ou Yang,Wei Haiping. Checking the Grid Spatial Index Based on File System Against That Based on Database System [J]. Journal of The Pla Institute of Surveying and Mapping,2002,19(3): 220 – 223.

[8] MickeyWilliams. Visual c#.NET Core Reference [M]. BeiJing: Tsinghua University Press, 2003