SuperMap Objects .NET 6R Technology Documents ——Spatial Indexes

SuperMap Software Co., Ltd.

Beijing China

Legal Statement

1. The copyright of this document is covered by SuperMap Software Co., Ltd. in accordance with the Copyright Law of the People's Republic of China and the Universal Copyright Convention. If, without the written permission of the company, any part of the document shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

2. "超图", "SuperMap", and  are the registered trademarks of SuperMap Software Co., Ltd., protected by the Copyright Law of the People's Republic of China. If, without the written permission of the company, the trademarks shall not in any way or any reason be used, copied, modified, transmitted, or bundled with other products to be used, sold, tort reserved.

3. This document represents no responsibilities of any supplier or agent. Without statement, SuperMap Software Co., Ltd. has right to do modifications to this document.

4. The copyright of trademarks mentioned in this document belongs to the corresponding companies. Without the written permission of these companies, the trademarks shall not in any way or any reason be used, copied, modified, or transmitted.

5. The concerned software products and the updated products hereinafter in this document are developed and sold by SuperMap Software Co., Ltd.

Hereby declare

SuperMap Software Co., Ltd.:

Add: 7/F Tower B, Technology Fortune Center, No. 8 Xueqing Road,

Haidian District, Beijing, 100192, P. R. China

Tel: +86-10-82736655-4170

Fax: +86-10-82734630

HomePage: www.supermap.com

Sales: request@supermap.com

Tech Support: globalsupport@supermap.com

SuperMap Software welcomes all advices and suggestions from you.

Content

1 Summary 1

2 SuperMap SDX+ Spatial Indexes Introductions 2

2.1 Building Spatial Indexes 3

2.2 Deleting Spatial Indexes 4

2.3 Special Illustration 5

3 R-Tree Indexes 6

3.1 Principles 6

3.2 Characteristics 7

3.3 Implementation Circumstance 8

4 QuadTree Index 9

4.1 Principles 9

4.2 Implementation Circumstance: 11

5 Tile Index 12

5.1 Principles 12

5.2 Implementation Circumstance 14

6 MultiLevel Grid Index 15

6.1 Principles 15

6.2 Characteristics 16

6.3 Implementation Circumstance 16

6.4 Footnote 17

7 Nil Spatial Index 18

7.1 Implementation Circumstance 18

1

Summary

The development of GIS leads to increasingly huge data amount which drags the access speed of spatial data very much. Spatial indexes are data structures employed to raise the spatial query efficiency.

Spatial indexes of GIS are on the basic of field indexes. The field index is a data structure locating key values in a fast way, which is popular in database system and computer systems. Additionally, the field index is able to access some specific key values within a moment. The data structure of field indexes is B-tree index or derivatives of B-tree index. Since traditional field indexes cannot manage with fast locating to the spatial data which has spatial relationships with each other, spatial indexes are necessary to be able to access the spatial data in a second.

The four spatial indexes, Quadtree index, R-Tree index, Dynamic index, Tile index provided by SuperMap SDX+ are going to be introduced in this section.

2

SuperMap SDX+ Spatial Indexes Introductions

SuperMap SDX+ experienced a long history of spatial indexes research. Until now, SuperMap SDX+ provides indexes methods including QuadTree, R-Tree, Tile, and Dynamic. QuadTree encode spatial data by improved Hilbert coding, with integrative data and index storage. R-Tree index is the expansion of B-Tree in multi-dimensional space, providing higher query efficiency with separate spatial data and index data storage. Tile index has remarkable advantages in displaying tile map. Dynamic index is a new type of index developed by SuperMap GIS, supporting database engines only, using the database query as much as it could.

Each dataset can only apply one index. However, the applied index are allowed users to change, which means the new index can be used only after the old one deleted. In the editing dataset status, the system will protect the current index as default. Particularly, after repeated data compilations, the efficiency of index will be affected more or less. The system will judge if it is necessary to rebuild a spatial index.

The following are the main ports in the SuperMap Objects .NET, the program language used in this example is .NET.

2.1 Building Spatial Indexes

（1）Creating a spatial index for the vector dataset according to a given spatial index.

boolean DatasetVector.buildSpatialIndex (SpatialIndexType spatialIndexType);

spatialIndexType represents the spatial index type, whose values are listed below:

Table 1 Spatial Indexes Type Value Table

CONSTANT	DESCRIPTION
MULTI_LEVEL_GRID	Dynamic Index
TILE	Tile Index
QTREE	QuadTree Index
RTREE	R-Tree Index
NONE	Nil Index

（2）Creating a spatial index for the vector dataset according to the spatial index information.

boolean DatasetVector.BuildSpatialIndex (SpatialIndexInfo spatialIndexInfo);

spatialIndexInfo is the spatial index information, whose values are listed below.

Table 2 Spatial Indexes Information Table

Parameters	Classes	Description
GirdCenter	Point2D	The central point of grid index, which is usually the central point of datasets.
GridSize0	double	The first layer grid size of dynamic index should not be zero. Its unit must be the same with the geography unit.

GridSize1	double	The second layer grid size of dynamic index whose unit must be the same with the geography unit.
GridSize2	double	The third layer grid size of dynamic index whose unit must be the same with the geography unit.
LeafObjectCount	int	The number of leaf nodes in the R-Tree Spatial Index
TileField	String	Tile fields used in creating a tile index
TileHeight	double	The height of a tile
TileWidth	double	The width of a tile
Type	SpatialIndexType	The type of Spatial Indexes, whose value are listed in table: Spatial Indexes Information Table

2.2 Deleting Spatial Indexes

When the index of a dataset is broken during the compilation, there will be dirty area index which affects the index efficiency large. At this stage, we have to delete the old index and creat a new one. However, SDBPlus data does not support deleting indexes. (1) The port used to check if the index is broken is:

boolean DatasetVector.IsSpatialIndexDirty ();

If the True is returned, the index is broken which means a new spatial index should be created.

(2) The port used to delete indexes is:

boolean DatasetVector.DropSpatialIndex ();

If indexes are deleted, the True will be returned, otherwise the False will be returned.

2.3 Special Illustration

The current SDBPlus datasource does not support creating and deleting indexes. With the building of SDBPluse datasource, R-Tree Index will be the default spatial index.

Points datasets of the database does not support QuadTree, neither R-Tree indexes;

Network datasets does not support any spatial index;

Composite datasets does not support dynamic index;

Routes datasets does not support tile index;

Property datasets does not support any spatial index;

Database datasets that have more than 1000 records are able to create index.

3

R-Tree Indexes

3.1 Principles

R-Tree index is the expansion of B-Tree in multi-dimensional space, based on the structure of disks. R-Tree index has capacity to integrate with current popular database, support various spatial query operations. It is one of the most widely used spatial index methods.

R-Tree index includes some objects that are spatially near to each other into a virtual rectangle object, please see Diagram One. The rectangles, which contain the pointers of all spatial objects inside, are considered as nodes in the spatial index. For instance, A contains D, E, F; B contains H, I, J, K; C contains L, M. The R-Tree index structure is showed in Diagram 2.

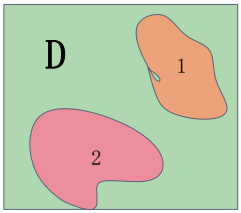

Diagram 1 The virtual rectangle

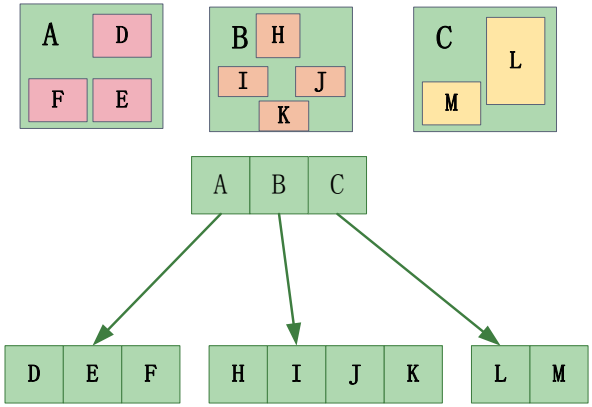

Diagram 2 R-Tree Index

In the spatial index, the first thing that needs to be done is making sure which rectangles are located in the index window. Second, select the content that should be indexed from the window. This process is able to improve the index speed.

3.2 Characteristics

（1）R-tree index provides high precise and efficiency results

（2）The index data and geography data in R-tree index are stored separately. The update of R-tree affects its balance and division, which is complicate and slowly. Usually, a new spatial index should be created after large scale of compilation. R-tree index is suitable for read-only data and data that rarely used in spatial compilation.

3.3 Implementation Circumstance

R-tree index is more suitable to static data. For instance, the data used as base map or rarely compiled (exclude the compilation of property data) could be organized by an R-Tree index.

4

QuadTree Index

4.1 Principles

QuadTree index is the best choice for two-dimensional data. The whole workspace is encoded as 0, creating four same parts encoded as 00, 01, 02, 03 by a horizontal line and a vertical line. The up left one is 00, the downright one 03. Repeat the stated process to the four parts respectively. For instance, to the 00 one, encoding 000, 001, 002, 003 to its four sub-spaces. This process could be carried on if necessary. For instance, number 031 could be separated into 0301, 0311, 0312, and 0313 and so on, as Diagram 3 displayed. The number of the divisions should be carried out decided by amount and average size of the research zone object.

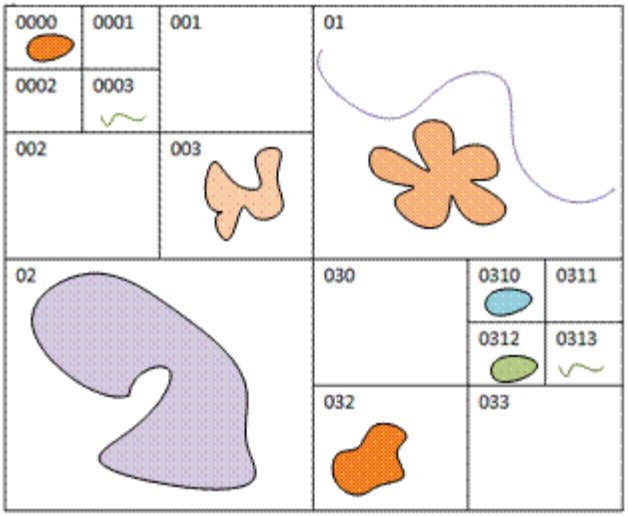

Diagram 3 QuadTree divisions

The following diagram shows the QuadTree index of the whole workspace in Diagram 3：

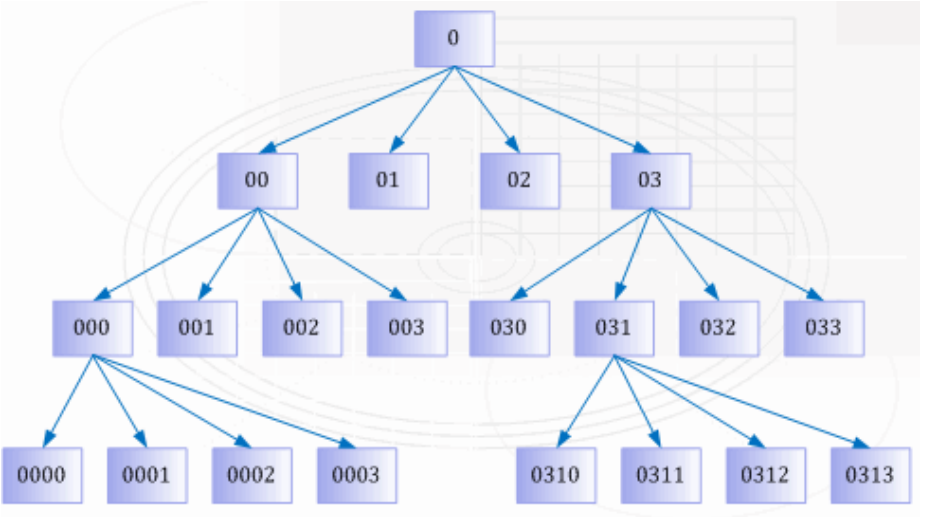

Diagram 4 QuadTree Index

4.2 Implementation Circumstance:

QuadTree index is ideal to small data amount and high rate of concurrent compilation.

5

Tile Index

5.1 Principles

In SuperMap SDX+, the spatial objects will be classed according to an attribute field of the dataset or a given scope (the length and width of a tile). The index is used to manage the divided spatial objects to improve the query speed. The diagram below is a Chinese Administration Boundary map in tile. When we create tile index according to the given scope, the scene numbered h4708 is displayed. Tile index is more suitable to massive data (more than a million objects) than others. When you query one or several tiles which will only be displayed by the tile index, large amount of time is saved.

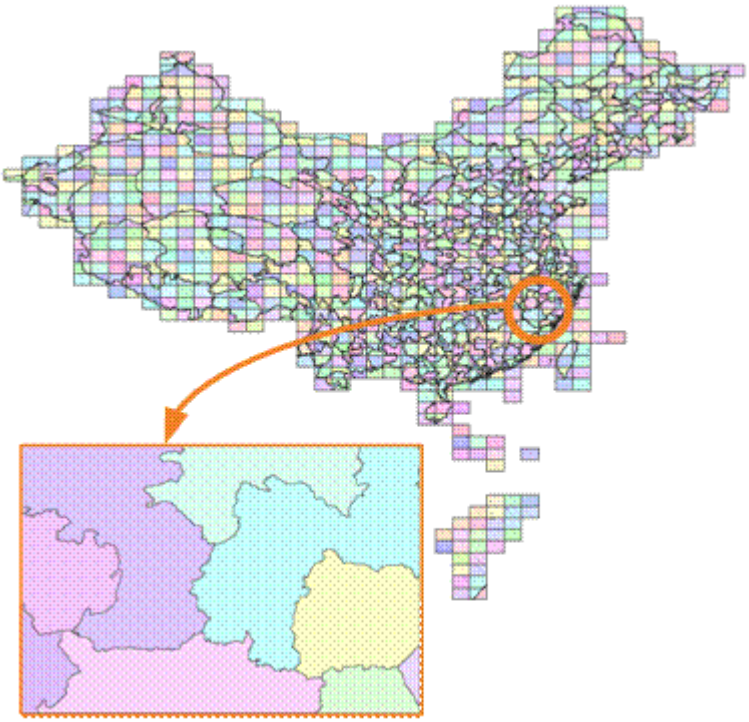

Diagram 5 Tile Index Diagram

Particularly, tile index could be stored in local cache (in binary format), which can further improve the query speed. File cash is intelligent distribution scheme to relieve the loads of network and servers, increase application integral performance. When you turn on the file cache item, application program will check if there is new version of the wanted data on the local cache library. If there is no desired cache data or the data is not the latest one, the right data will be retrieved from server and update the correspondent local ones. Consequently, the local data could be read directly. If the latest version is just located in the local cache, the application program will retrieve it to display or analyse, rather than ask from the server by network. Hence, the loads of database server and

network will largely decline, while the application integral performance will be improved greatly.

Users can configure the location of the cash file. The dataset file is able to be put into a table of contents through the property CacheFileLocation of ugc.xml, the system configuration file.

5.2 Implementation Circumstance

The dataset that consists of data stored in tiles according to standard scales (e.g. 1:250000, 1:100000, 1:50000, etc) will produce amazing outcome with the tile index. Furthermore, using together with the cache option will provide a wonderful map browsing speed.

Engines that support tile index are: SQLPlus, OraclePlus.

6

MultiLevel Grid Index

6.1 Principles

MultiLevel Grid index manage data by creating multiple level grid layers. The dataset is divided into grids with identical size or not, according to particular principles. The grid records the locations of every geography object. In GIS, the regular net grid is popular. In spatial query, the grid contains the desired objects is located first, and then the geography object will be found on the right grid. Pleas look at the diagram of the procedure:

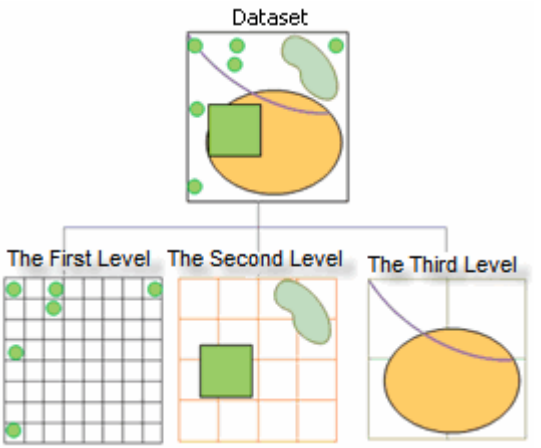

Diagram 6 MultiLevel Grid Index Diagram

In the current version, the indexes of grid are the First Class, the Second Class, and the Third Class. Each class has its individual division principle. The first class has the smallest grid unit which carries 50% of the data. The second class grid is bigger than the first class', and the third class grid unit is the biggest of them, and so forth.

6.2 Characteristics

1. In browsing state, the speed of MultiLevel Grid index will be faster.
2. Easily to update and concurrent;
3. High spatial index precise and accuracy.

Note: this index method supports dynamic concurrent compilation of datasets.

6.3 Implementation Circumstance

MultiLevel Grid index combine the advantages of R-Tree index and QuadTree index, and provides fully support to concurrent compilation. It is the most popular index approach, and the default index method in the current and further versions.

It is recommended to use MultiLevel Grid index if you are not sure which index method should employ.

6.4 Footnote

The engines supporting to create MultiLevel Grid index include SQLPlus, OraclePlus.

With MultiLevel Grid index, users are able to store data in local cache which can improve the query and browse speed. To the data bigger than GB class, this method will give great performance.

7

Nil Spatial Index

7.1 Implementation Circumstance

Nil spatial index means there is no any spatial index method involves in the spatial query. This is used when the data size is very small. For instance, when the table information is broken (by users...), SDX+ will consider the dataset has no index at all.