

RESEARCH ON THE DISTRIBUTED PARALLEL SPATIAL INDEXING SCHEMA BASED ON R-TREE

Yuan-chun Zhao^{a, b}, Cheng-ming Li^b

^aShandong University of Science and Technology, Qingdao 266510

^bChinese Academy of Surveying & Mapping, Beijing 100039

Commission II, SS-15

KEY WORDS: Distributed parallel computing environment, Parallel spatial indexing, Parallel R-tree indexing, Parallel spatial data Partitioning algorithm

ABSTRACT:

As one of the key technologies for improving the efficiency of parallel processing of a huge volume spatial data under the distributed parallel computing environment, the parallel spatial indexing mechanism is discussed in this paper. On the viewpoint of the existing research results, the classical R-tree indexing structure and its variations have the well characteristic for parallel organizing and processing of spatial data, therefore, a new multi-tiers parallel spatial indexing structure based on R-tree (HCMPR-tree, multi-tiers parallel R-tree based on Hilbert spatial filling curve) is presented. Differing from the present parallel spatial indexing structures, the attributes of the parallel computing pattern has been considered adequately, and a high performance parallel spatial data partitioning algorithm (HCS DP, spatial data partitioning algorithm based on the Hilbert spatial filling curve) has been applied in the HCMPR-tree. Based on the theoretical and technical issues of classical methodology of parallel algorithm design, HCMPR-tree has not only a well indexing structure for parallel computing, but it also obtains the high efficiency for load-balance between the different nodes under the distributed parallel computing environment. In the paper, using the system response time of the parallel processing of spatial scope query algorithm as the performance evaluation factor, the availability and the efficiency of HCMPR-tree has been sufficiently proved by implementing parallel processing on the testing spatial datasets.

1. INTRODUCTION

The capability of computing and data processing of traditional GIS will not fully satisfy the need of spatial applications of compute-intensive and I/O-intensive, such as spatial data mining, spatial decision support system, spatial multi-dimension dynamic visualizing analysis and simulation, etc. With the applications of grid computing technology in GIS has become hotspot, distributed parallel computing mode and system architecture will help to improve the whole performance and efficiency of traditional GIS. Therefore, distributed parallel computing will turn into a key method to solve insufficient computing capability of traditional GIS.

The organization and management of a huge volume of spatial data is not only the basis of all kinds of complex spatial applications, but the one of the core issues in GIS, in which, spatial indexing is the important research direction of data organization and management. At present, spatial indexing has plenty of research productions and quite extensive application in DBMS and GIS, but less research on the organization, management, storage, processing and application of a huge volume of spatial data, while existing research has difference in the platform and fields of applications. The paper discusses the parallel spatial indexing mechanism based on the R-tree Indexing and liner chain table structure, which is one of the key technologies for improving the efficiency of parallel processing of a huge volume of spatial data under the distributed parallel computing environment. The design and implement of the parallel spatial indexing structure presented in the paper is based on the classical parallel computing methodology of Ian Foster, and the factors of impacting the efficiency of the parallel spatial indexing, such as spatial data partitioning algorithm and load-balance strategy have been considered adequately,

forasmuch, the spatial indexing structure is proper for variant complex applications under distributed parallel computing environment such as GRID.

2. INTRODUCTION OF PARALLEL SPATIAL INDEXING SCHEMA

The few existing research productions refer to the storage management of a huge volume of spatial data in the parallel GIS. However, as one of the key factors of impacting the efficiency and the performance of parallel GIS, it is necessary to research the schema of the organization and management of mass spatial data. Therefore, the more research projects on the parallel spatial indexing structure have been developed. Especially, on the aspect of parallelization of R-tree indexing structure has had more research results.

As the pioneer of the research on the parallel R-tree indexing schema, Kamel and Faloutsos present the parallel R-tree indexing structure running on the single CPU and multi-disks hardware system that the indexing structure improves the data throughput of spatial query by using parallel I/O method. Schnitzer and Leutenegger discuss the M-R (Master R-tree) and MC-R (Master-Client R-tree) indexing structure. The common point of the two indexing structure is that all of the non-leaf nodes of the parallel R-tree have been stored on the main node of the multi-host computing environment. Differing from MC-R indexing structure, the leaf nodes and entity objects of M-R indexing structure have been stored on the sub-nodes of the multi-host computing environment. All of the three indexing structure above adopt the primary-secondary pattern, therefore, the main node of the multi-host computing environment is easy to become a hotspot and bottleneck to increase the system

response time and reduce the performance of the spatial query.

Since the inherent parallel characteristic of R-tree indexing structure, it is fit for applying in parallel GIS and parallel spatial database management. However, there are some obvious deficiencies found easily from the existing research results of parallel R-tree indexing structure under the parallel computing environment, including 1) the more close to the root nodes of R-tree the non-leaf nodes of R-tree are, the more they are accessed possibly by database transactions, eventually they would become the hotspots and the bottleneck. 2) When the leaf nodes of R-tree are merged or split, these operations can affect generally other neighboring nodes, and even the whole R-tree. 3) The share lock and exclusive lock would be applied when the leaf nodes of R-tree are accessed. For example, some applications need to update the nodes of R-tree, and these nodes accessed will be locked, it results in that other transactions can not do anything for the R-tree, consequently, the parallel processing abilities of R-tree are reduced. Although there are some shortages of R-tree indexing structure, it is the most choice proved by practice for building the parallel spatial indexing structure by optimizing the structure design.

On the viewpoint of the existing research results of parallel R-tree indexing schema, the key issue of building parallel R-tree indexing structure under distributed parallel computing environment is that the new leaf nodes how to distribute among the multi-disks. There are three principles for distributing operation; including 1) the data balance principle. The quantity of leaf node is same on each disk, and the sum of data must be balanced. 2) The area balance principle. The covered area by all the spatial entity on the each disk should be balanced; otherwise, the disk that has the more covered area would be the hotspot and bottleneck. 3) The spatial relationship balance principle. For enhancing the data throughput of spatial data query, the

overlapping or the neighboring leaf nodes should be distributed on the different disk. Based on the principles as above, the paper presents the method for building the parallel R-tree indexing structure under the parallel computing environment, such as network cluster system.

3. PARALLEL R-TREE INDEXING SCHEMA

3.1 Spatial Data Partitioning Technology

The data partitioning technology in the realm of database management system has been proved that is an effective method to solve the data non-balance problem distributed on the different disk after data partitioning, namely data skew issue. Similarly, the spatial data partitioning technology in the spatial database play also an important role for avoiding data skew problem and enhancing the parallel query and search performance of a huge volume of spatial data. The spatial data partitioning technology is the basis of building the parallel R-tree spatial indexing structure presented in the paper, and also is the precondition to guarantee high performance and high efficiency of parallel R-tree spatial indexing structure.

For keeping to the principles of building parallel spatial indexing schema addressed in the second section of the paper and satisfying the requirement of spatial data partitioning, the paper adopts the spatial data partitioning algorithm based on Hilbert curve, HCS DP, presented in the reference paper [5].

3.2 Parallel R-tree Indexing Structure

The paper presents the Hilbert space-filling curve based multi-tiers parallel R-tree based on the HCS DP, HCMPR-tree, and its structure design shown as fig.1.

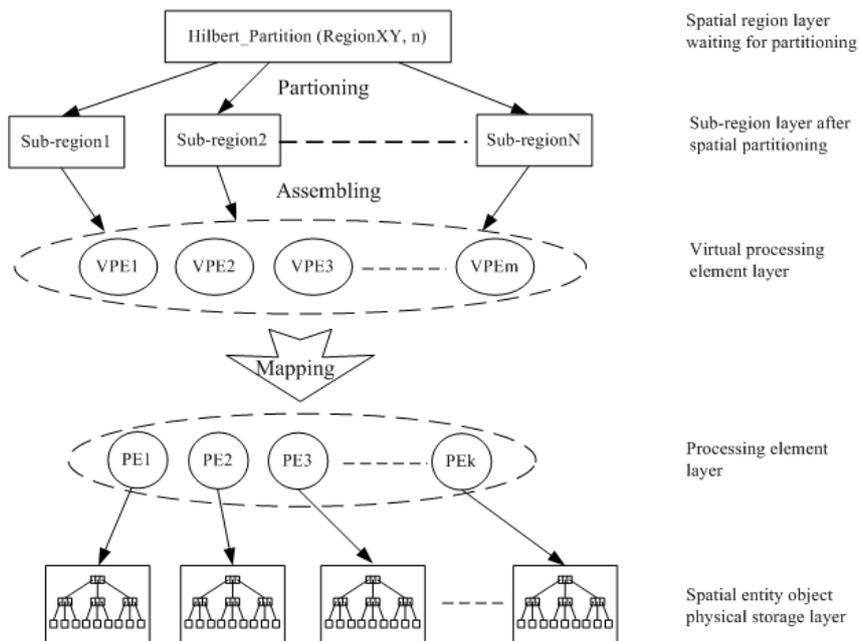


Fig.1 HCMPR-tree Indexing Structure

For meeting the application requirement of the parallel R-tree indexing structure under the distributed parallel computing environment, the structure design of HCMPR-tree is based on the classical methodology of the parallel algorithm design

pattern of Ian Foster, that is, partitioning, communication, assembling and mapping. The introduction to each layer of HCMPR-tree has been given as follows.

(1) Spatial region layer waiting for partitioning. The layer is an aggregation in which all of the spatial entities and their relative attribute data are. Each entity in the sub-region from which the layer has been partitioned by using the HCSDP algorithm has a sequential Hilbert code. The coding method of spatial entity by using HSCDP algorithm can ensure that there has the better spatial relativity among each of spatial entity in one sub-region and avoid that spatial entities have been dissevered badly after spatial data partitioning. Simultaneously, HCSDP algorithm can ensure the balance of sum of spatial entity among each sub-region.

(2) Sub-region layer. The layer is the results that the spatial region layer as above has been partitioned by using HCSDP algorithm.

(3) Virtual processing element(VPE) layer. The layer has been applied to implement the assembling operation. The sum of the VPE is m, generally, $m \ll n$ (n is the sum of sub-region). VPE would be mapped to the practical processing element in practice. As we know, the superposition of each of nodes of R-tree is one of the crucial factors to affect the performance of R-tree indexing structure. VPE would provide a better solution to reduce the superposition of each of nodes of R-tree. Each sub-region in the sub-region layer can be distributed into the different VPE by using some data partitioning method, such as round cycle method, and the spatial area covered by all the sub-region in each VPE is disjunctive each other.

(4) Processing element layer. The layer includes storage element or processing element (PE), such as control nodes, computing nodes and storage nodes in the network cluster. Generally, the sum of PE k is less than or equal to the sum of VPE m. In the layer, VPE would be partitioned and mapped to the PE by using round cycle algorithm or other method. Each PE will include one or more VPE by applying the mapping operation, and the upper-limit of the sum of sub-region included in the PE is determined by the value of n, m and k.

(5) Spatial entity object physical storage layer. As fig.1 shown, all of the sub-region has been ultimately distributed onto the physical processing element by applying one partitioning, one assembling and one mapping operation. Since there are two or more sub-region in each PE and the area covered by each sub-region is disjunctive each other, therefore, the course of building R-tree on each PE can be processed individually and be propitious to enhance the speed of building R-tree indexing structure. When all the R-tree of each PE on the network cluster has been built successfully, the distributed characteristic of all the R-tree on different PE is help to implement the parallel query and update of a huge volume of spatial data.

3.3 The physical storage structure of HCMPR-tree

The physical storage structure of leaf node of HCMPR-tree includes a set of item that consists of I and a row identifier, and I denotes the minimum bounding rectangle (MBR), the row identifier identifies the entity object denoted by the MBR in the spatial database. There are two kind of non-leaf node in HCMPR-tree; one is the non-leaf nodes built from sub-region which be denoted by a set of item including RID, I and the pointer of sub-node, RID identifies the sub-region, I denotes the MBR of the sub-region and the pointer points to the root node of R-tree corresponded to the sub-region; the other is the inner non-leaf node of R-tree of sub-region which be denoted by a set of item including I and the pointer of sub-node, I denotes the MBR of the node and the pointer points to the node of the next

layer. The physical storage structure of the root node of R-tree on each PE consists of SID, I and the pointer of sub-node, SID identifies the PE, and I denotes the MBR covered by the all spatial entity on the PE, and the pointer points to non-leaf node of R-tree. Different from the existed parallel R-tree spatial indexing structure, the top-level structure of HCMPR-tree is a link list, and each node in the link list includes five item, RID, SID, I, HS and HE. RID identifies the sub-region. SID is the identifier of the PE that stores the sub-region, I denotes the MBR of the sub-region. HS is the original value of Hilbert code, and HE is the terminal value. The link list would be stored on the main control host of Network Cluster. The n value is limited. Therefore, the link list can be load into the main-memory of the main control host, and high effective query to link list can be achieved. The link list used by HCMPR-tree can avoid the accessing collision when all the non-leaf nodes of R-tree are stored on the main control host.

4. EXPERIMENTATION AND ANALYSIS

Different from the traditional R-tree indexing, the size of the leaf-node of HCMPR-tree is one of the key issues to affect the efficiency of spatial data query and search. Generally, there are two instances, ①if the size of leaf-nodes is smaller, such as it only includes one spatial entity object, then many PE would be accessed by CPU when performing a small scope spatial query, and because of increasing of the network communication throughput, the network might be blocked. ②If the size of leaf-nodes is bigger, then only one PE maybe be accessed by CPU when executing a bigger scope spatial query, resulting in reducing the degree of parallelization of spatial data query and search and weakening the whole performance of parallel GIS and parallel spatial database.

Many kinds of factors can impact the confirmation of size of leaf-node of HCMPR-tree, such as transformation speed of network data, computing capacity of PE, counts of PE, etc. The paper takes system response time of spatial extent query aiming at spatial database as function evaluation index to get minimize system response time while finding the best value of leaf-node. Table 1 is the description of kinds of parameters to determine system response time, which reference standard is the computer of Intel Pentium IV 2.4GHz running on Windows 2000 professional. Table2 is the necessary parameters and description to decide the size of the best leaf-node. K denotes the sum of PE accessed by some spatial query, and the mathematic relationship of C, K and Q is expressed as follows.

$$C_{opt} = \frac{-Q}{N_{pe} \times \log(1 - K_{opt} / N_{pe})}$$

Firstly, the top-level link list of HCMPR-tree located on the main host node of the network cluster must be traversed when performing any one spatial scope query to determine the sum of PE and SID covered by the spatial query scope qx and qy, then main host node sends the spatial query message to all the K PEs and waits the query results responded from them. Under the condition of the computer hardware shown in the table 1, the optimized value of C is one disk page. For validating the efficiency of the optimized size of the leaf-node of HCMPR-tree indexing structure, the paper selects two variables as the parameters, the system response time of spatial query and the size of the leaf-node. The testing dataset in the

experimentation is the district boundary map of the county and city of the state which is the scale of one to one million and the size of the testing dataset is 17.6 MByte. The sum of PE is 2, 4 and 8 respectively, and the value of the q_x or q_y ($q_x = q_y$) is 0.1 to 0.5. Fig 2 illustrated the relationship between the system response time of spatial query and the size of the leaf-node under three PE configurations. The scheme of the experimentation in the paper refers to the correlative methods

presented by Koudas. The results of the experimentation are similar to Koudas' s, that is, when the size of the leaf-node is one disk page (one Spage); the system response time of any one scope query under the condition of the different sum of PE is optimized.

Parameter	Description	value
N_{pe}	The sum of the PE on the network cluster	2~8
B	The bandwidth of the LAN on the network cluster	100 MBit/s
T_{page}	The time of the I/O operation to random disk page	0.01s
$T_{cpu-msg}$	The time of building a message package by CPU	0.001s
S_{page}	The size of the disk page	4K bytes
S_{msg}	The size of each message in the processing of the data transfer(IEEE 802.3 protocol)	1500 bytes
S_{header}	The size of the header section of each message(IEEE 802.3 protocol)	72 bytes
T_{idle}	The time of sending a message on the idle network	0.01s

Table 1 Parameters for evaluating respond time of system query and the introduction

parameter	desription
T_{msg}	The time for sending a message on busy network
q_x, q_y	The boundary of scope query
D	The sum of spatial entities in spatial dataset
Q_{data}	The sum of testing dataset(bytes)
Q	The sum of the disk page occupied by all spatial entities in the query boundary q_x, q_y
C	The size of leaf-node(pages)
C_{opt}	The size of the optimized leaf-node(pages)

Table 2 Parameters using for evaluation of leaf node size and the introduction

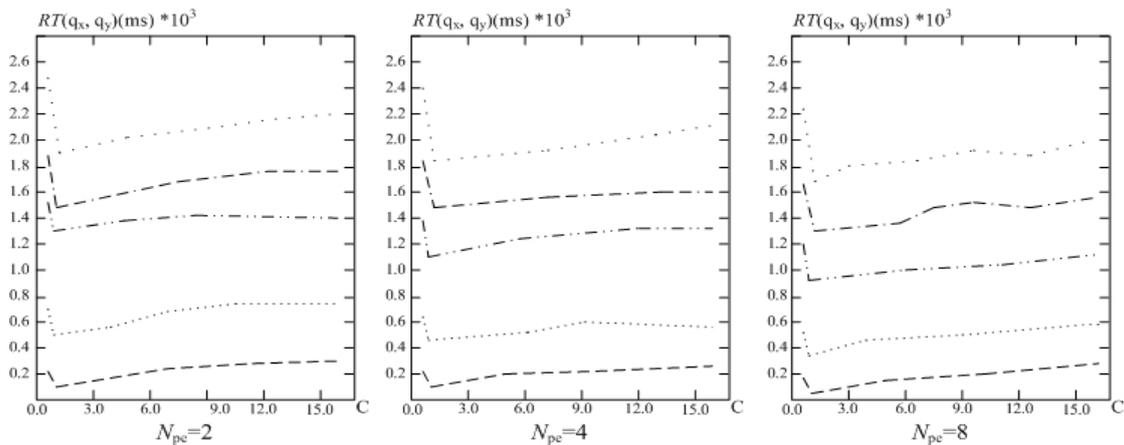


Figure. 2 Relationship between the Response Time of System Query and Leaf Node Size

5. CONCLUSION

The existed R-tree parallel indexing structure can not fully satisfy the need of management and parallel processing of huge volume spatial data under the distributed parallel computing environment for not designed aiming at the characteristics of

distributed parallel computing environment and parallel computing methods. Based on high efficient HCDSP algorithm and according to methodology of parallel algorithm design of Ian Foster, the paper constructs HCMPR-tree structure. It' s proved that the indexing structure can not only obtain better load balance, but be suitable for parallel design of spatial data

processing algorithms, such as spatial extent query algorithm, spatial analyzing algorithm. With reasonable design and high efficient function, the structure can improve parallel processing ability of huge volume spatial data effectively.

REFERENCE

ALI M H, SAAD A A., 2005. The PN-Tree: A Parallel and Distributed Multidimensional Index [J]. *Distributed and Parallel Databases*, 17: 111-133.

HEALEY R G, DOWERS S, GITTINGS B M, MINETER M J., 1998. *Parallel Processing Algorithms for GIS* [M]. London: Taylor and Francis.

KARNEL I, FALOUTSOS C. 1992. Parallel R-trees [C]. ACM SIGMOD, USA

NICKERSON B G, GAO F., 1998. Spatial Indexing of Large Volume Swath Data Sets [J]. *International Journal of Geographical Information Science*. 12(6): 537-559.

TANIAR D, RAHAYU J W. ,2004. Global parallel index for multi-processors database systems [J]. *Information Sciences*. 165(1-2): 103-127.

WANG B, HORINOKUCHI H, KANEKO K, MAKINOCHI. ,1999. A parallel R-tree search algorithm on DSVM [C]. *Proceedings of the International Conference on Database Systems for Advanced Applications*, 237-244.

Zhao Chun-yu, Meng Ling-Kui, Lin Zhi-Yong., 2006. Searching In a Spatial Data Partitioning Towards the Parallel Spatial Database System [J]. *The Transaction of Wuhan University (Information Sciences)*. 31(11):962-965.

