Spatial Data Declustering Method Considering Spatial Locality for Parallel Spatial Database

Yan Zhou
School of Automation
University of Electronic Science and Technology of China
Chengdu, P.R.China
zhouyan_gis@yahoo.com.cn

Qing Zhu, Yeting Zhang
State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing
Wuhan University
Wuhan, P.R.China
zhuq66@263.net, zhangyeting@263.net

Spatial data declustering is an important data processing method for parallel spatial database especially in shared nothing parallel architecture. Spatial data declustering can achieve parallel dataflow to exploit the I/O bandwidth of multiple parallel nodes by reading and writing them in parallel, which can improve the performance of parallel spatial database evidently. Aiming at the unique spatial objects locality, this paper presents a novel spatial data declustering method, which uses Hilbert space-filling curve to impose a linear ordering on multidimensional spatial objects, and to partition spatial objects logical segments according to this ordering to preserve spatial locality of spatial objects, and then to allocate logical segments to physical parallel nodes based on round-robin rule. Experimental results show that the proposed method can obtain well spatial data declustering results.

spaital data deculstering; spatial locality; parallel spatial database; spatial data balancing;

I. INTRODUCTION

Parallel spatial database is an inevitable development trend for high performance spatial database. The most research of parallel spatial database has focused on shared nothing parallel architecture because of its availability, scalability and high cost performance ratio. Spatial data declustering plays an important role in shared nothing parallel spatial database by declustering spatial datasets to allocate each parallel node, which can provide parallel dataflow to exploit the I/O bandwidth by reading and writing them in parallel. Spatial data declustering means to partition spatial datasets into many segments to allocate these segments approximately evenly to parallel nodes.

Many data declustering methods have been proposed. Four typical one-dimensional declustering methods are provided in [1] and [2]. But these one-dimensional methods are unfit to multidimensional spatial data. Some multidimensional method is presented subsequently such as Grid [3], CMD [4], and HCAM [5]. Reference [6] discussed how to use these above multidimensional data declustering methods to partition spatial data. Oracle Spatial provides two ways, one is to decluster spatial datasets based on X or Y-coordinates value, other is based on X and Y-coordinates value in [7]. These data declustering methods were originally designed for structured data. Unfortunately, spatial data is unstructured variable length data, has non-uniform distribution and spatial locality relationship characteristics, which means spatial data declustering is more difficult than general structured data declustering, which need not only satisfy data volume balancing distribution on each parallel node, but also keep spatial locality of spatial objects after data declustering. According to Tobler's first law of geography in [8] is "Everything is related to everything else, but near things are more related than distant things". These motivate our research into a spatial data declustering method considering spatial locality.

Our research based on shared nothing parallel database to propose a novel spatial data declustering method. The rest of this paper is organized as follows. Section II describes the proposed spatial data declustering method, which includes three main parts: logical segments partitioning, logical segments optimizing, and physical segments declustering. Section III gives the experiments and discusses the results. Section IV is the conclusion.

II. SPATIAL DATA DECLUSTERING METHOD

Considering spatial locality means that adjacent spatial objects should store together physically because good clustering reduces the number of disk accesses on retrieval, improving the response time [9]. So the results of spatial data declustering should keep good clustering characteristic in physical storage organization. Hilbert curve is an availability way to achieve this goal.

A. Logical Segments Partitioning

To use initial order Hilbert space-filling curve to impose a linear ordering on multidimensional spatial objects, and then partition spatial objects according to this ordering to preserve spatial locality of spatial objects. To partition whole space into coarse grid cells, and take the Hilbert code as spatial object code when the center point of spatial object find in the interior of corresponding grid cell. Many spatial objects could share the same code. To sum up data volume of spatial objects in each grid cell according to the order of Hilbert code until the cumulative data volume is more than average data volume V_{ave}, obviously, it satisfies (1). Here, V_{total} denotes the total data volume of spatial objects. P denotes the number of parallel database nodes.

$$V_{ave} = V_{total} / P \qquad (1)$$

Subsequently, to decompose the last cumulative grid cell into four sub-grid cell and recalculate the Hilbert code of sub-grid. To add spatial object data volume of sub-grids to the cumulative data volume orderly, if need, hierarchical decomposition process of grid cell should be carried out until the cumulative data volume approximately equal to V_{ave}, or the order number of sub-grid satisfies (2). Here, n denotes number of spatial objects. M denotes final order number of sub-grid hierarchical decomposition.

$$M = \left[\frac{1}{2}\log_2 n\right] + 1 \tag{2}$$

Thus, spatial objects of cumulative grid cells form the first logical segment. The rest of logical segments could be partitioned by the same iterative processes.

B. Logical Segments Optimizing

Theoretically, the number of logical segments can be equal to or be an integer multiple of the number of parallel nodes. Actually, the too small number of logical segments could degrade the parallel degree of system, and too large number of those could destroy spatial locality of datasets. The number of logical segments could realize to optimize when the query frequency of spatial database is known.

Using $Q_1,...,Q_n$ describes n number spatial query on spatial relation R, $Tcpu_i$ denotes the CPU time to execute Q_i, $Tdisk_i$ denotes disk processing time of query Q_i, $Tnet_i$ is the communicating time of query Q_i, PQ_i is the number of data pages needed to search for query Q_i, FQ_i denotes the query frequency of Q_i. To defined the average CPU execution time, the average disk processing time, the average communicating time, and the average retrieving pages number respectively as (3), (4), (5), and (6).

$$Tcpu_{ave} = \sum_{i=1}^{n} Tcpu_i \times FQ_i \tag{3}$$

$$Tdisk_{ave} = \sum_{i=1}^{n} Tdisk_i \times FQ_i \tag{4}$$

$$Tnet_{ave} = \sum_{i=1}^{n} Tnet_i \times FQ_i \tag{5}$$

$$PQ_{ave} = \sum_{i=1}^{n} PQ_i \times FQ_i \tag{6}$$

Assuming CPU processing, disk operating and data communicating couldn't parallel work, then the average responding time (T_{ave}) for the average query (Q_{ave}) of spatial relation R on single parallel node should satisfy (7).

$$T_{ave} = Tcpu_{ave} + Tdisk_{ave} + Tnet_{ave} \tag{7}$$

Generally, for parallel system, the query performance should improve with increasing the number of parallel nodes. However, too many parallel nodes could bring extra start up costs. Assuming extra costs is a constant denoted as Ts, obviously, the response time to execute Q_{ave} using X number parallel nodes is (8).

$$T(X) = \frac{Tcpu_{ave} + Tdisk_{ave} + Tnet_{ave}}{X} + X \times Ts \tag{8}$$

The main goal of spatial data declustering is to minimize the responding time of spatial queries by data balancing distribution on parallel nodes. According to equation (8), to minimize $T(X)$ means (9):

$$\frac{dT(X)}{d(X)} = 0 \tag{9}$$

Based on (9), X is easy to infer as (10). That means the responding time of spatial queries achieve minimum value when the number of parallel nodes is equal to X. Here, X is looked as the optimized number of parallel nodes.

$$X = \sqrt{\frac{(Tcpu_{ave} + Tdisk_{ave} + Tnet_{ave})}{Ts}} \tag{10}$$

If executing Q_{ave} need to access PQ_{ave} number data pages, then in order to maximize query performance, to need X number parallel nodes to work in parallel to ensure the minimized $T(X)$. Corresponding, the optimized size of logical segment denoted as V_e is as (11), easy to infer that the optimized number of logical segments named as N is as (12), here C denotes the size of physical page, the means of V_{total} is as same as in (1). More related discussion for above assumptions could find in [2].

$$V_e = \frac{PQ_{ave} \times C}{X} \tag{11}$$

$$N = \frac{V_{all}}{V_e} \tag{12}$$

C. Physical Segments Declustering

After logical segments partitioning, to allocate logical segments to parallel nodes, our research uses round-robin principle to achieve physical allocation of logical segments because the round-robin is an easy and high efficiency method. According to (13) to allocate the *j-th* logical segment to the *i-th* parallel node named as P_i, here $i \in [0,1,...,P-1]$, $j \in [0,1,...,N-1]$, the P and N have the same meanings as mentioned above.

$$P_i = j \bmod P \tag{13}$$

The aim of logical segments partitioning is to obtain even size data segments, these logical segments doesn't means actual physical storage on parallel nodes. In order to achieve physical storage, physical segments declustering need to be done. The main of physical segments declustering is to decrease seek time and delay time of physical disks by selecting proper size of physical declustering. Considering the actual physical storage, the size of physical declustering generally satisfies (14).

$$V_{phyical} \le \alpha \times C \tag{14}$$

Here α is a storage capability coefficient, and satisfied $\alpha \le 1$. C denotes the size of physical page. Considering spatial database updating such as insert, delete and modify operating, generally, the experiential value of α is proper from 2/3 to 3/4.

The spatial data declustering method proposed in this paper is described briefly as follow.

- Construct initial Hilbert curve to partition the whole space into grid cells;
- Code and sort each spatial object according to the Hilbert code of the center point of spatial object;
- Sum up data volume of each grid cell according to the order of Hilbert code until the cumulative data volume is more than V_{ave} ;
- Decompose the last cumulative grid cell into four sub-grid cells and repeat previous step until cumulative data volume is V_{ave} or the order number of sub-grid equal to M;
- Map the above logical partition to the first parallel node, and then recursively carry out these steps until all of logical segments be mapped to corresponding parallel nodes used round-robin method.

III. EXPERIMENTS AND RESULTS

Our experiment builds a shared-nothing parallel network environment with 6 general computers, one of these simulates client computer, and the rest simulate parallelism handling nodes. Each computer has the same hardware: CPU 1.8GHz, memory 256M, and connect with 100Mbps Ethernet. Experimental dataset contains 54122 spatial objects and the size of dataset is 20 736 875 byte. Table I is the results of spatial data declustering on three parallel nodes, and Table II is the results on five parallel nodes. The results of two tables show that our proposed data declustering method always obtain spatial data balancing results whatever the number of parallel nodes is. Moreover, each node includes different numbers of spatial objects when the size of each node is approximately evenly, which showed the unstructured variable length characteristics of spatial data. Fig. 1 illustrates the declustering results with five colors. Different color denotes corresponding spatial sub-datasets stored on different parallel nodes. From the Fig.1, we can see that the same color sub-datasets consist of many locality data objects, which show spatial datasets achieve well logical segments clustering status. Meanwhile, each logical segments stored on same node detaches from each other, which help to maximize parallel degree when facing to spatial large querying.

TABLE I. SPATIAL DATA DECLUSTERING (P=3)

Node No.	Spatial Objects Num.	Dataset Size（Byte）
P_1	16 231	5 848 160
P_2	17 864	5 848 448
P_3	15 375	5 850 544

TABLE II. SPATIAL DATA DECLUSTERING (P=5)

Node No.	Spatial Objects Num.	Dataset Size（Byte）
P_1	9 932	4 147 340
P_2	11 239	4 146 994
P_3	10 726	4 147 184
P_4	10 624	4 148 070
P_5	11 583	4 148 070

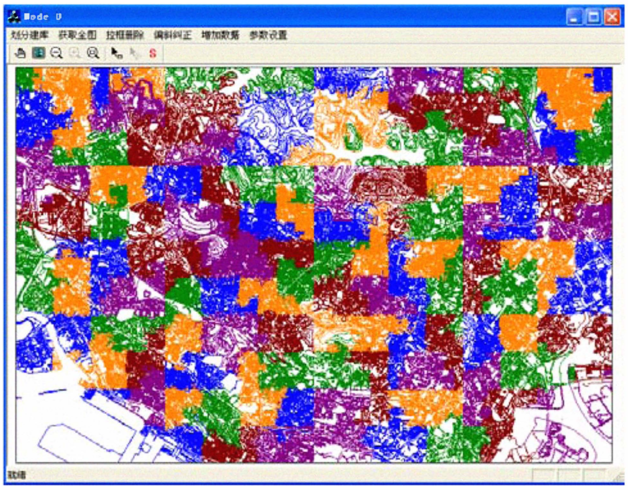

Figure 1. Spatial locality results after spatial data declustering.

IV. CONCLUSION

Spatial data declustering is an important research content for parallel spatial database. Considering spatial locality and unstructured variable length characteristics of spatial data, this paper proposed a novel spatial data declustering method by using Hilbert curve to impose a linear ordering to keep spatial locality of spatial objects. Our experimental results show that the proposed spatial data declustering method could effectively achieve spatial data balancing and keep well spatial locality based on shared nothing parallel architecture.

ACKNOWLEDGMENT

This research supported by the National High Technology Research and Development Program of China (No. 2008AA121602), National Natural Science Foundation of China (No. 40871212 and 40701144) and Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (No. 08I02).

REFERENCES

[1] DeWitt D J, Ghandeharizadeh S, Schneider D, et al. "The Gamma Database Machine Project," IEEE Transaction on Knowledge and Data Engineering, vol. 2(1), 1990, pp. 44 -62.

[2] Ghandeharizadeh S, DeWitt D J. "Hybrid-range partitioning strategy: a new declustering strategy for multiprocessor databases machines," Proceedings of the sixteenth international conference on Very large databases, Brisbane, Australia, 1990, pp. 481- 492.

[3] Hua K A, Lee C. "An adaptive data placement scheme for parallel database computer systems," Proceedings of the 16th International Conference on Very Large Databases, 1990, pp. 493-506.

[4] Li J Z, Srivastava J, Rotem D. "CMD: A Multidimensional Declustering Method for Parallel Database Systems," In Proceedings of the 18th International Conference on Very Large Data Base Conference, Canada, 1992, pp. 1-14.

[5] Faloutsos C, Bhagwat P. "Declustering using fractals," Proceedings of the second international conference on Parallel and Distributed Information Systems, San Diego, California, United States, 1993, pp. 18-25.

[6] Shekhar S and Chawl S., Spatial Databases A Tour, China Machine Press, Beijing, 2004.

[7] Abugov D, Oracle Spatial Partitioning: Best Practices (an Oracle White Paper), Oracle Inc., 2004.

[8] Tobler W., "A computer movie simulating urban growth in the Detroit region". Economic Geography, vol. 46(2), 1970, pp. 234-240.

[9] Moon B, Jagadish H V, Faloutsos C, et al., "Analysis of the clustering properties of the Hilbert space-filling curve," IEEE Transactions on Knowledge and Data Engineering, vol. 13(1), 2001, pp. 124 -141.