An Improved Hilbert Curve for Parallel Spatial Data Partitioning

MENG Lingkui  HUANG Changqing  ZHAO Chunyu  LIN Zhiyong

Abstract  A novel Hilbert-curve is introduced for parallel spatial data partitioning, with consideration of the huge-amount property of spatial information and the variable-length characteristic of vector data items. Based on the improved Hilbert curve, the algorithm can be designed to achieve almost-uniform spatial data partitioning among multiple disks in parallel spatial databases. Thus, the phenomenon of data imbalance can be significantly avoided and search and query efficiency can be enhanced.

Keywords  parallel spatial database; spatial data partitioning; data imbalance; Hilbert curve

CLC number  P208

Introduction

With the rapid development of GIS applications, especially in multi-dimensional dynamic GIS and WebGIS, the demand for capability and high efficiency of huge-amount data processing is ever increasing and has given rise to parallel GIS as well as parallel spatial databases.

The parallel spatial database is one of the foundations of the parallel GIS in both theory and application[1]. This database has the advantages of high performance, high reliability and good scalability, which enable it to meet requirements of various thick-data applications. However, many challenges emerged when applying the parallel database theory to construct the spatial database[2-4]. In this study, we address the data partitioning problem in parallel spatial databases and present an improved Hilbert curve and a new spatial data partitioning method based on it. The aim is a relatively uniform distribution among multiple disks, thus increasing the parallelization degree and improving the efficiency of searching and querying.

1  Spatial data partitioning problem

In general, spatial data has characteristics such as variable data types and structures, massive amounts and complex spatial relationships that make data partitioning strategy in relation-based spatial databases distinct from ordinary commercial ones. In particular, data characteristics are very different from each other. As the data types of general commercial databases are usually common, such as integers and characters, the size of their items, whose upper bound remains invariant, can be estimated. However, the memory size required for spatial entities vary extensively. In fact, sizes of all spatial objects are different except for point objects. The items of ordinary commercial databases are generally isolated and lack explicit relationships; while for spatial data there are many associations, such as topological relationships, directional
relationships, and metrical relationships, which should be taken into full account when the spatial data is partitioned\cite{5}.

The existing data partitioning algorithms are usually designed for general relational databases, without consideration of the data amount of each item itself. Generally, these algorithms put emphasis on the relatively uniform distribution of the amount of items among multiple disks, i.e., the number of items on each disk is kept balanced. However, for spatial data partitioning, the uniform distribution of item numbers might lead to high asymmetry of the total storage amount on each disk, thus giving rise to the severe data-imbalance phenomenon.

The Oracle Spatial is a typical method utilizing relation-oriented databases to manage spatial data. It provides two range partitioning schemes based on spatial positions, including: ① partitioning based on the range of the $X$ coordinate or $Y$ coordinate; ② partitioning based on the ranges of the $X$ coordinate and $Y$ coordinate. Such partitioning schemes are only concerned with position attributes of the spatial objects represented with MBR (minimal boundary re-dangle) and ignore the impact of physical storage size of the object. Suppose there are two regions of comparable area, in one where there are mainly point objects, while in the other there are mostly polygon objects. Obviously in this case the item numbers of the two regions could be very close, while the aggregate storage amount might differ very much and lead to load imbalance among disks.

Existing work in the area of parallel GIS mainly focuses on the study of mechanisms of parallel searching\cite{6}, while research on parallel partitioning of spatial data is rare. For example, Kamel and Faloutsos proposed a method of compact $R$-tree construction by utilizing the space filling curve as the one-dimensional sequence code and provided a spatial data partitioning scheme for multi-machine systems. But one shortcoming of the method is the possibility of the mainframe being the bottleneck when the number of users concurrently accessing the database increases. Meanwhile, because of the incapability of multiple datasets to implement partitioning according to the same boundary, the efficiency of parallel searching for multiple datasets is low. To address the above problem, we propose a Hilbert space filling curve for spatial data partitioning (HCSDP).

## 2 Hilbert space filling curve (HSFC)

The main idea of the space fitting curve, such as the $Z$ curve and Hilbert curve, is to map an object in a multi-dimensional space onto a one-dimensional curve via a linear sequence\cite{7,8}. Due to its excellent spatial clustering performance\cite{9}, the Hilbert space filling curve researched is the main tool of spatial partitioning.

### 2.1 Hilbert space filling curve

Hilbert space filling curve is derived from the classical Peano curve family, which is sequential mapping from the close interval cell $I = [0, 1]$ to close rectangle cell $S = [0,1]^2$. The spatial relationship among all the spatial entities will be held to a certain extent as the Peano curve transforms from one dimension to $N$ dimensions.

As one of the Peano curve family, HSFC has higher spatial clustering performance as the $Z$ order and can gradually partition the sequential space, in which the array order is correlative of the degree of space subdivision. The process of gradually subdividing the Hilbert curve is that an atom is set to recursion to produce the HSFC, as shown in Fig.1.

![Fig.1 Subdivision based on Hilbert space filling curve](image)

### 2.2 The generation of Hilbert value

In GIS, spatial objects include the point, line, polyline, and polygon. Without loss of generality, we choose the type of spatial dataset as polygons. As the Hilbert curve is designed for the point objects, the datasets need to be pre-processed to assign the poly-
gon object a Hilbert value: ① get the centre point $O_i$ of the MBR of each polygon and represent the $i$th polygon as $O_i$; ② according to the coordinates of $O_i$, calculate the corresponding Hilbert value for the $i$th polygon. Through this method, the polygon data set can be transferred to a point data set on which the Hilbert curve is constructed. The complexity of the construction of Hilbert curve is $O(2^m)$, in which $m$ denotes the iteration numbers to generate the curve; we can refer it to the order of the Hilbert curve. To approach the true position of the point object as well as possible and thus reduce repetition of Hilbert values for point object sets, generally the choice of the value of $m$ should be determined by the number of the spatial objects of interest, i.e., it should meet the constraint $n < 2^{2m}$. Throughout our simulation, we assign $m = 8$, which means we can support up to $2^{16}$ spatial entities.

### 2.3 HCSDP algorithm

First, the definition of variables in the HCSDP algorithm is given.

$N$ is the number of disks in the system; $n$ is the number of all spatial entities to be partitioned; $V_i$ is the size of the $i$th spatial entity in bytes; $i$ takes value from $\{0, 1, \ldots, n-1\}$; $V_{\text{other}}$ is the total size of all non-spatial fields in each item. Because the size of non-spatial fields can be estimated in advance, we can represent the aggregate size of the non-spatial fields by a single variable whose unit is also byte; $V_{\text{avg}}$ is the average volume of all the spatial entities on each disk in bytes; and $B_j$ is the total size of the objects contained in the $j$th bucket, with each bucket corresponds to a disk.

The above variables satisfy the equation:

$$V_{\text{avg}} = \frac{n V_{\text{other}} + \sum_{i=0}^{n-1} V_i}{N} \quad (1)$$

Based on the HSFC, algorithm procedures are designed as follows:

1) Scan all spatial data sets, construct the Hilbert curve and attribute each spatial entity a Hilbert value; account for each $V_i$ and calculate $V_{\text{avg}}$ via Eq. (1); initialize all $B_j$ to be zero.

2) Sort the spatial entities in increasing order according to Hilbert value.

3) Beginning from $i = 0$, $j = 0$ and $k = 0$, put the $k$th spatial object to the $j$th bucket and update $B_j$ according to $B_j = B_j + V_k$.

4) Compare $B_j$ and $V_{\text{avg}}$. If $B_j$ is smaller than $V_{\text{avg}}$, then $i = i + 1$, $k = k + 1$, otherwise $j = j + 1$, $k = i$. Repeat this step until $i = n - 1$, $k = n - 1$, $j = N - 1$.

5) If $B_{N-1} < V_{\text{avg}}$ and $k < n - 1$, the remaining spatial entities could be assigned to each bucket in decreasing bucket order through the round-robin method.

6) According to the final value of all $B_j$, map the $N$ buckets to corresponding $N$ processing nodes (disks).

### 3 Trials and analysis

In this section we provide some numerical results and corresponding analysis of the experimental data.

#### 3.1 The numerical results

We adopt a national county-level administrative border map with a scale of one to a million as the trial data, which has a total of 2330 polygon objects and spatial data stored in the Shape file format with 17.6 Mb size. To validate the performance of the HCSDP algorithm, only the ID field and geometric object field are used. The type of ID field is an integer, which means that $V_{\text{avg}}$ equals 4 bytes; assuming that the $i$th spatial object comprises $m$ points, then the object size $V_i$ can be approximated as 16 m, with byte being the unit. For the case of three and five disks, the numerical results are shown in Table 1 and Table 2 respectively.

| $V_{\text{avg}}$ | Aggregate size | Entities amount |
|------------------|----------------|-----------------|
| 5 894 936        | 8 841 841 5 894 878 5 895 089 | 1 768 480 8 |
| 1 768 480 8      |                | 563 962 805 2330 |

| $V_{\text{avg}}$=3 536 961 | Aggregate entities size | Entities amount |
|---------------------------|-------------------------|-----------------|
| $B_0$                     | 3 536 911               | 377             |
| $B_1$                     | 3 536 945               | 409             |
| $B_2$                     | 3 536 988               | 558             |
| $B_3$                     | 3 536 979               | 403             |
| $B_4$                     | 3 536 985               | 583             |
| Total                     | 17 684 808              | 2 330           |
From Table 1 and Table 2, it can be shown that the algorithm can achieve almost uniform distribution among multiple disks. Here “uniformity” means the relative balance of the aggregate storage size, rather than the number of spatial entities on each disk. The amount of entities on the disks varies significantly, which is correlated with the complex degree of the contained objects.

For comparison, we provide the spatial data partitioning results of the same test data we adopted in the above simulation by utilizing the partitioning strategy provided by Oracle Spatial, which is illustrated in Fig.2.

![Data partitioning results by Oracle Spatial](image)

The data partitioning strategy of Oracle Spatial is analogous to the construction of the grid index for the data sets and then mapping each grid unit to multiple storage nodes by the round-robin method. The mapping rule is as follows: first index each grid unit, and then determine the index of the disk for each grid unit through the round-robin method. It can be seen from Fig.1 that the amount of spatial objects covered by each grid unit varies greatly, which leads to the imbalance among the disks. Experiments indicate that such a load imbalance problem is hard to avoid by adopting other border partitioning schemes, including even the irregular border scheme, whose complexity of obtaining the grid border is not low.

### 3.2 Algorithm analysis

The HCSDP algorithm belongs to a range-based partitioning strategy, with the value of \( V_{avg} \) being the boundary. Due to the good spatial clustering capability of the Hilbert curve, neighboring objects are very likely to be partitioned to the same disk or adjacent disks, thus guaranteeing the amount of the disks accessed in a range query is as small as possible. As the aggregate size of the spatial entities on each disk is relatively balanced, the number of pages for storing the spatial objects has minimal difference. This can not only balance the accessing load, but also improve the throughput and ability of processing concurrent data. In Table 3 and Table 4 we make a response time comparison between the HCSDP algorithm and the Oracle Spatial algorithm. The querying boundary in the two tables refers to the normalized length of borderline of the queried area, i.e., mapping the object to the \([0,1] \times [0,1] \) square. For every querying boundary, we run 100 queries and then average the response time with ms as the unit.

| Query boundary | HCSDP | Oracle Spatial |
|----------------|-------|----------------|
| 0.1            | 221   | 238            |
| 0.2            | 796   | 897            |
| 0.3            | 1487  | 1863           |

| Query boundary | HCSDP | Oracle Spatial |
|----------------|-------|----------------|
| 0.1            | 202   | 215            |
| 0.2            | 598   | 832            |
| 0.3            | 1237  | 1634           |

The above comparisons indicate that the response time advantage of the HCSDP algorithm over the Oracle Spatial exhibits itself when the querying boundary value is relatively large. The main reason is as follows: in the HCSDP algorithm the neighboring spatial entities tend to be assigned to the same or adjacent disk(s). When the querying range enlarges, the number of accessed disks is very likely to be smaller in HCSDP than in the Oracle Spatial, thus the response time can be reduced for HCSDP. But when the querying range is small, there is not much difference between the two algorithms, and the Oracle Spatial algorithm can even outperform the HCSDP in certain cases. For example, when the queried range happens to fall in some grid unit in Fig.1, it only takes access to one disk for the Oracle Spatial algorithm, while the HCSDP algorithm needs to compute the corresponding Hilbert values of the queried range, leading to more time being used.
4 Conclusions

The parallel spatial database, which is the foundation of parallel GIS applications, can provide powerful support for future applications of grid computation in GIS. Being a key technique in the parallel spatial database, spatial data partitioning strategies directly influence the overall performance of the database. In this study we extend the Hilbert space filling curve and apply it to spatial data parallel partitioning. Numerical results validate the ability of load balancing and relatively high efficiency of parallel spatial data querying.

References

[1] Mineter M J, Dowers S, Gittings B M(2000) Towards a HPC framework for integrated processing of geographical data: encapsulating the complexity of parallel algorithms [J]. Transaction in GIS, 4(3): 245-262
[2] Dowers S, Gittings B M, Mineter M J(2000) Towards a framework for high-performance geocomputation: handling vector-topology within a distributed service environment [J]. Computers, Environment and Urban Systems, 24: 471-486
[3] Mineter M J(2003) A software framework to create vector-topology in parallel GIS operations [J]. International Journal of Geographical Information Science, 17(3): 203-222
[4] Healey B, Dowers S, Gittings B M, et al.(1998) Parallel processing algorithms for GIS [M]. UK: Taylor & Francis Ltd.
[5] 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
[6] Patel J, Yu Jiebing, Kabra N, et al.(1997) Building a scalable geo-spatial DBMS: technology, implementation, and evaluation[C]. SIGMOD’97, AZ, USA
[7] Kumar A, Muhanna W A(1998) Analysis of the performance of spatial ordering methods [J]. International Journal of Geographical Information Science, 12(3): 269-289
[8] Lu Feng, Zhou Chenghu(2001) A GIS spatial indexing approach based on Hilbert ordering code [J]. Journal of Computer-Aided Design & Computer Graphics, 13(5): 424-429
[9] Moon B, Jagadish H V, Faloutsos C, et al.(2001) Analysis of the clustering properties of the Hilbert space-filling curve [J]. IEEE Transactions on Knowledge and Data Engineering, 13(1): 124-141

---

Call for Papers—the Eighth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences

The International Symposia on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences have been held at various locations since 1994. This series of biannual symposia is organized by the International Spatial Accuracy Research Association and focuses on uncertainty in spatial information, analysis, and applications emphasizing natural resources and the environment.

To emphasize strategic and systematic research, Accuracy 2008 will include the following topics: accuracy metrics and spatial assessment; data assimilation and uncertainty quantification; error analysis in geomatic engineering; geostatistics for spatial uncertainty; models of uncertainty for discrete objects, continuous fields, and their integration; models of uncertainty for nominal and ordinal fields; models of scale and methods for scaling; semantic interoperability and uncertainty analysis; spatial data conflation and information fusion; uncertainty characterization in digital terrain analysis; uncertainty characterization in ecological and environmental modeling; uncertainty characterization in social systems and their interactions with environmental systems; uncertainty characterization in landscape dynamics and global change; vagueness in spatial information and analysis; validation in remote sensing; visualization of uncertainty. These topics are non-exclusive and set no limits to potential contributors.

Detailed schedule is as follows. Submission of extended abstracts: January 12, 2008; notification to authors: February 11, 2008; submission of final papers: April 29th, 2008; early registration: April 29th, 2008; symposium: June 25th-27th, 2008.

Please refer to http://2008.spatial-accuracy.org for more details.