Point collection partitioning in MongoDB Cluster

Shuai Zhang[1], Bolei Zhang[2], Zhenjie Chen[1], Sanglu Lu[2]

1Department of Geographical Information Science, Nanjing University, Nanjing, Jiangsu, 210093, China
Telephone: 86+02583593770
Fax: 86+02583593770
Email:zhangshuai.nju@gmail.com
[2]State key laboratory for novel software technology, Nanjing University, Nanjing, Jiangsu, 210093, China
Telephone: 86+02589683467
Fax: 86+02589683467
Email: zblhero@gmail.com

1. Introduction

Parallel spatial database has seemed to become an inevitable trend of high performance spatial database development. Partitioning datasets in order to balanced loads among multi processors is an important ambition. However, spatial data has its own characteristics which make it quite different from others. Its key problem is how to partition spatial data to distributed nodes in the parallel environment with regard to its spatial relationships between features. Spatial data has its topological relationships, spatial locality and spatial resemblance etc. to consider; otherwise the performance of geographic algorithms (operators) will be slowed down and the waste of computing resources can be resulted in.

Not surprisingly there are considerable literatures on the topic of partitioning geographical data (Goodchild, 1989; Samet, 1989; Sloan et al 1999; Harel & Koren, 2001; Han et al., 2001). However, the existing spatial data partitioning methods are mainly developed under the environment of relational DBMSs. Such systems will encounter big problems when a large number of read/write operations per second occur. Recent years a number of new systems, as “NoSQL” database, have been designed to provide good horizontal scalability for simple read/write database operations. Horizontal scaling allows dozens or hundreds of machines to operate as a single database system, performance improving approximately linearly with the number of machines, while traditional relational database systems failed to scale well when their data is distributed over many servers.

So, we launched a project trying to develop a NoSQL spatial database to make most of the advantages of NoSQL technology. MongoDB not only actually shares the characteristics of NoSQL database, but also has the ability to store points collections and support some simple spatial query and analysis, such as proximity queries, Bounded queries, special geospatial index and so on. So, here in this paper we represent three types of point cluster data partitioning strategies, Random Partitioning Strategy (RPS), Space filling curve Partitioning Strategy (SPS), and K means Partitioning Strategy (KPS), in such NoSQL database cluster system to see their performance in spatial query.

2. Spatial data partitioning

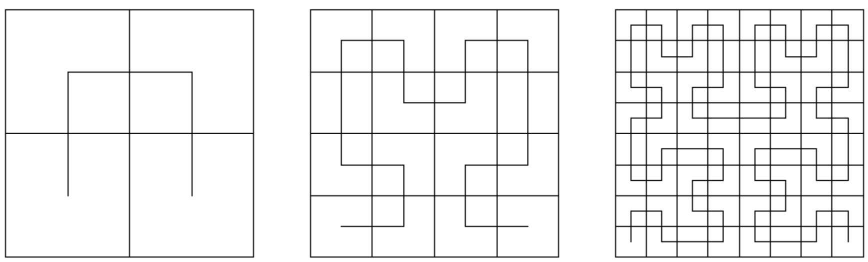
Sharding is MongoDB’s approach to scaling out. Sharding partitions a collection and stores the different portions on different machines. In order to shard collections, a

Supported by the National High Technology Research and Development Program of China (2011AA120301) and the Scientific Research Foundation of Graduate School of Nanjing University (2013CL09)

specific shard key will be needed. The shard key, a field that exists in every document in the collection, determines the distribution of the collection's documents among the cluster's shards. MongoDB distributes documents according to ranges of values in the shard key. A given shard holds documents for which the shard key falls within a specific range of values. So in this case, if we take no consideration of spatial relationships in the NoSQL database, e.g. we choose the objectID of each document as the shard key. The spatial distribution of the points collections will be quite random, so we call this strategy as Random Partitioning Strategy (RPS).

2.1 Hilbert space filling curve

Space-filling curves (SFCs) have been extensively used as a mapping scheme from the multi-dimensional space into the 1-D space. A SFC is a thread that goes through all the points in the space while visiting each point only one time. Thus, a SFC imposes a linear order of points in the multi-dimensional space. SFCs are discovered by Peano where he introduces a mapping from the unit interval to the unit square. Hilbert generalizes the idea to a mapping of the whole space. Following Peano and Hilbert curves, many SFCs are proposed，but among these, the Hilbert curve achieves better clustering properties than other SFCs (Bongki,etc, 2001, Mokbel etc.2002). So we choose Hilbert space filling curve code as the sharding key of the points collections to distribute spatial data.



(a) First step (b) Second step (c) Third step

Fig.1 The first three steps of the Hilbert space filling curve

2.2 k means Clustering

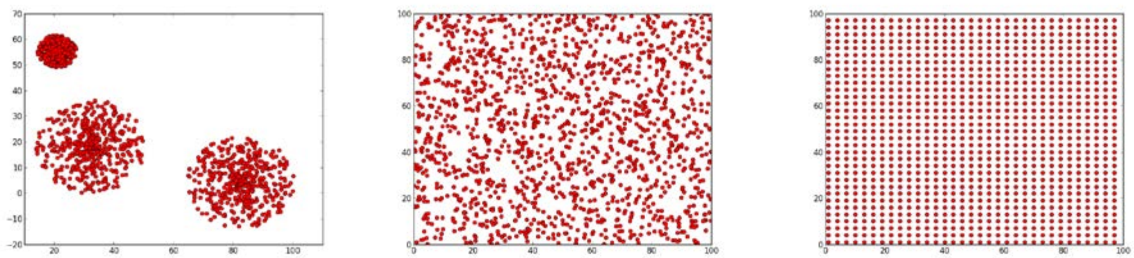
K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. In this work we use K means algorithms to cluster the point collections and give a continuous range of values within each cluster, so that points within one cluster are supposed to be distributed in the same shard server in the MongoDB cluster. The main idea of Kmeans is to define k centroids, one for each cluster. this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.

3. Experiment and Results

Three point data sets (figure 2) were used to evaluate control point distribution effects in the framework of the MongoDB cluster. The first point collection has three clusters and each of them has a random distribution within it but has a different radius. The second point collection has a random distribution within the range [0,100]. And the third one has a uniform distribution. All of them has exactly 100 thousand points in and are respectively de-clustered by the three strategies mentioned above, that are RPS, SPS, and KPS. And equal amount of spatial queries loading were performed on the nine points collections. The results seen from table 1 are the average response time of ten thousand spatial queries operating ten times.



(A) (B) (C)

Fig 2 Data sets of one hundred thousand points within a range of [100, 100] in its x and y values, (A) three clusters, each of which has a random distribution,(B) a random distribution, (C) a uniform distribution

partitioning Strategy / data type	RPS	SPS	KPS
A	103.47	68.97	58.96
B	79.49	58.79	57.03
C	83.66	59.43	58.37

Table 1 The average response time of ten thousand spatial queries operating ten times, dimensionless unit

4. Conclusion

We studied the three types of point cluster data partitioning strategies, Random Partitioning Strategy (RPS), Space filling curve Partitioning Strategy (SPS), and K means Partitioning Strategy (KPS), in MongoDB database cluster. And from the results of the experiments we can easily see that KPS has much better spatial query stability than the other two, and that spatial data partitioning strategy is a very important factor to improve the performance of parallel spatial database. Spatial data unbalancing distribution can severely degrade the performance of parallel spatial database and their performance in spatial query.

References

Bongki Moon, H.v. Jagadish, Christos Faloutsos, Joel H. Saltz, "Analysis of the Clustering Properties of the Hilbert Space-Filling Curve," IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 1, pp. 124-141, Jan.-Feb. 2001, doi:10.1109/69.908985

Frank, A. & Timpf, S. (1994) Multiple representations for cartographic objects in a multi-scale tree: an intelligent graphical zoom. Computers and Graphics Special Issue: Modelling and Visualization of Spatial Data in Geographic Information Systems,18(6), 823-829.

Goodchild, M.F., 1989, Tiling large geographical databases. *the Design and Implementation of Large Spatial Databases*, pp137–146.

Han, J., Kamber, M., & Tung, A.K.H. , 2001,. Spatial Clustering Methods in Data Mining: A Survey. *Geographic Data Mining and Knowledge Discovery*, pp1-29.

Harel, D. & Koren, Y. , 2001, Clustering spatial data using random walks. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp281 - 286.

Jonathan K. Lawder and Peter J. H. King. 2000. Using Space-Filling Curves for Multi-dimensional Indexing. In Proceedings of the 17th British National Conferenc on Databases: Advances in Databases (BNCOD 17), Brian Lings and Keith G. Jeffery (Eds.). Springer-Verlag, London, UK, 20-35

J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297

Kothuri, R.K., Ravada, S., & Abugov, D. (2002) Quadtree and R-tree Indexes in Oracle Spatial: A Comparison using GIS Data. ACM SIGMOD Madison, Wisconsin, USA.

Mark, D.M. (1986) The Use of quadtrees in geographic information systems and spatial data handling. Procs.Auto Carto London, Vol.1, pp. 517-526.

MongoDB sharding, 2013.01, http://docs.mongodb.org/manual/core/sharded-clusters/

Mokbel, Mohamed F.; Aref, Walid G.; and Kamel, Ibrahim, "Performance of Multi-Dimensional Space-Filling Curves" (2002). Computer Science Technical Reports. Paper 1546.

Samet, H., 1989, The design and analysis of spatial data structures Addison Wesley, Reading, Massachusetts.

Sloan, T.M., Mineter, M.J., Dowers, S., Mulholland, C., Darling, G., & Gittings, B.M. (1999). Partitioning of Vector-Topological Data for Parallel GIS Operations: Assessment and Performance Analysis. Euro-Par'99 Parallel Processing, Vol.1685/1999, pp691 -694.

Tu, J., Chen, C., Huang, H., & Wu, X. (2005) A visual multi-scale spatial clustering method based on graph-partition. In Geoscience and Remote Sensing Symposium, 2005.IGARSS '05. Proceedings, Vol.2,745-748.

van Oosterom, P. (1995). The GAP-tree, an approach to `on-the-fly' map generalization of an area partitioning. GIS and Generalization: Methodology and Practice, pp.120-132.

van Oosterom, P. & Schenkelaars, V. (1995) The development of an interactive multi-scale GIS. International Journal of Geographical Information Systems, 9(5), 489-507.

van Putten, J. & van Oosterom, P. (1998) New results with Generalized Area Partitionings. Proceedings of the International Symposium on Spatial Data Handling, pp485-495.

Wu, H., Pan, M., Yao, L., & Luo, B. (2007) A partition-based serial algorithm for generating viewshed on massive DEMs. International Journal of Geographical Information Science, 21(9), 955-964.

Wu, A. & Leahy, R. (1993) An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11), 1101-1113.

Yang, W. & Gold, C.M. 1999, Managing spatial objects with the VMO-Tree. Proceedings Seventh International Symposium on Spatial Data Handling , pp. 711-726, Delft, The Netherlands.

Yu-Lung Lo, Kien A. Hua, and Honesty C. Young, 2001, GeMDA: A Multidimensional Data Partitioning Technique for Multiprocessor Database Systems, *Distributed and Parallel Databases*, 9(3):211-236