ISSN:0975-9646
Kumutha Priya M et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 6 (6) , 2015, 5163-5165

Hierarchical Structure of Geospatial Field Data Using Enhanced Rtree

Kumutha Priya M,
Student, M.Tech,
Department of Information Technology
Vivekananda College Of
Engineering For Women,
Namakkal, India

Brinda B M,
Assistant Professor,
Department of Information Technology,
Vivekananda College Of
Engineering For Women,
Namakkal, India.

***Abstract*-Geosciences include remote sensing images, climate model simulation and so on. The spatial-temporal data have become more and more multidimensional, enormous and are constantly being modernized. As an outcome, the integrated maintenance of these data is becoming a challenge. A blocked Effective Hierarchical structured version within the split-and-merge hypothesis for the compacted storage, endless updating and data querying of multidimensional geospatial field data. The unique multidimensional geospatial field data are split into small blocks in accordance to their spatial-temporal references. The blocks are then characterized and compressed as Hierarchical structures for updating and querying. They are then combined into a single hierarchical tree. The use of buffered binary tree data structure and equivalent optimized operation algorithms, the original data can be constantly compressed, attached, and queried. In comparison with conventional systems, the new approach is revealed to keep hold of the features of the original data with much lesser storage expenses and faster computational performance. The outcome implies an efficient structure for integrated storage, presentation and computation of multidimensional geospatial field data.**

Key words: **Geospatial data, Spatial-Temporal Reference, HTR, RTree.**

1. INTRODUCTION

The data observation and model simulation rapidly develops in geosciences. The data from these systems have high dimensionality and huge volumes. Bulk amount of observation of exiting attributes/variables are successively produced by large-scale observation systems. These data are compressed for storage and the lately arrived data must be constantly compressed and attached to the present data, such that these data are integrated to the existing data as a whole. This updation procedure should be done in a short time and can be continually applied for the next piece of fresh data. The compression and storage must preserve the reliability of the spatial-temporal reference (STR) of this data. Balances the data accuracy, compression performance and improve the index and query analysis. The explosion of both the data volumes and dimensionality makes storage, management, query and processing a scary approach for existing results. Conventional methods make use of data indexes to speed up the query and storage. When the dimension grows, the data segmentation along with data structure are becoming complex and inefficient. Big data or data-intensive computing results use parallel data I/O and computation to fasten the data accessing and updating. On the other hand, huge computers and complex computation architectures are required to provide the I/O bandwidth and computation power needed. This condition turns out to be worse when the continuous data compressing, attaching and updating are necessary. Within the current data version and analysis framework, neither the conventional methods nor the big data or data-intensive computing solutions are matched for dynamic data attaching and updating. Hence finding optional data structures that fit the essential storage architecture may be difficult. The current exiting solutions for constant data processing need different data structures in the management, query and analysis measures that requires to undergo numerous difficult processing steps before they attain the final stage. The regular data transmit between different data structures slows down the processing throughput.

Tensor is a vital tool for multidimensional data processing and analysis. It is derived from data-intensive purposes. Computationally-oriented researchers classify multidimensional arrays as tensor-structured datasets. Tensor decomposition, tensor-based PDE solving and signal mining are then be applied for pattern mining, high dimensional data strategy and prophecy. Yet, many of these tools are for particular computations and contain inadequate functions. In many case, the curse of the dimensionality and Null Space problems are present.

These problems call for a new data structure and algorithms that supports data organization, compressed storage, data attaching and query. The successive data attaching and arbitrary data access of multidimensional geospatial field data involves data to be stored in every dimensional configuration parallel with blocks rather than stored as a whole. In this manner, immediate and secure data organization, competent and compressed data storage, uninterrupted data attaching can be applied to every block separately. A hierarchical tensor decomposition based on the split-and-merge concept is developed for constantly compressing and attaching of multidimensional geospatial field data. Our intention is to propose a hierarchical data structure to reformulate and store the huge volume of geospatial field data and to expand the techniques for data storage, querying and computational support by means of this data structure.

2. RELATED WORK

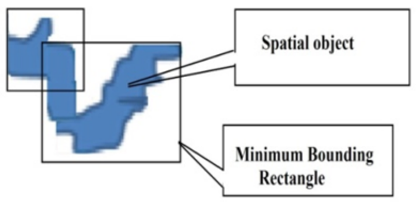
Hierarchical Tensor Representation for spatial data is a recent trends in geospatial data. So many approaches are already in use to compress, updating and data querying of multidimensional spatial field data. Each technique used its own representation of spatial data. Our algorithm falls into HTR.

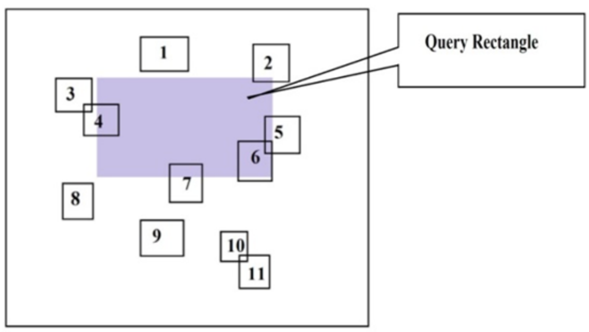
3. IMPLEMENTATION OF MODULES

3.1 Geospatial Data: (Spatial Index RTree)

A familiar method to look for an object based on their spatial position is location-based search, for instance to find all restaurants within 5 Kms of my current location, or find all colleges within the zip code of 651101. All spatial objects can be signified by an object id, a minimal bounded rectangle (MBR), with other attributes. So the space can be signified by assortment of spatial objects. A query can be represented as an additional rectangle. The query is regarding the location of the spatial objects whose MBR go beyond with the query rectangle.

RTree is a spatial indexing method that is given a query rectangle. This is to quickly locate the spatial object results. The concept is related to BTree. The spatial objects are grouped that are close to each other and structure a tree whose intermediary nodes contain "near-by" objects. Since the MBR of the parent node has all MBR of its children, the Objects are close by if their parent's MBR is minimized.





A world as a set of spatial object

3.2 Geospatial Search

Begin from root; check each child MRB to see if it overlaps with the query MBR. Skip the entire sub tree if there is no overlapping, or else, recur the search by drilling into each child. Unlike other tree algorithms the trees travels down a path. Our search here need to travel down multiple paths if the overlaps occur. Hence, to minimize the overlapping as high as possible the tree should be structured. This means minimizing the sum of MBR areas along each path (from the root to the leaf) as much as possible.

3.3 Geospatial Insert

To insert a new spatial object, start from root node, pick children node whose MBR will be the absolute least if the new spatial object is further, stride along this path until getting the leaf node. If the leaf node has space, insert the object to the leaf node. After insertion update the MBR of the leaf node as well as all its parents. If not, divide the leaf node into two and construct a new leaf node and copy several content of the original leaf node to this new one. And then insert the newly created leaf node to the parent of the original leaf node. If there is no space left in the parent , the parent will be split as well. If the split starts from the root, the original root will then be split and a new root is created.

3.4 Geospatial Delete

To delete, the spatial node will initially look for the data containing the leaf node. Eradicate the spatial node from the leaf node content and renew its MBR along with the parent MBR all the way to the root. If the new leaf node has less than m node, subsequently we have to reduce the node by deleting the leaf node. And now we eradicate the leaf node from the parent with updating them. Remove the parent from the parent's parent if the parent node is less then m. At present the intact node which is marked delete is separated from the RTree. Since all the nodes are not invalid several children that are valid (but removed from the tree) are reinserted and all these valid nodes are added back to the tree. Ultimately, the root node is checked to contain only one child and we discard the original root and use its own child to become the new root.

3.5 Geospatial Update

When the present spatial node modified from its original dimension updation occurs. The economical way is to modify the spatial node's MBR but not to alter the RTree. A better but expensive is to delete the node, transform its MBR and after that include it back to the RTree.

4. CONCLUSION

The data examination and model replication quickly develops in geosciences. The data from this information have huge volumes with high dimensionality. A fresh computational tool and data demanding scalable design that can sustain integrated storage, query and difficult scrutiny for such immense multidimensional datasets that will be crucial .Tensor is an expected means of representing multidimensional field data. They are the mathematical representations which are complicated for analysis. In this paper, a tree representation which can support the process of updating, compression, query and analysis over a substantial multidimensional geo-spatial field data was projected. By the split-and-merge concept, the RTree achieves the stability among data precision, memory occupation and running time for the data. Constant data appending and compression permits the data dimension to be controlled at meticulous levels with no losing the exactness of data representation. Since the computational competence is incredibly high and the memory cost is low even with high level of data, our process has the prospective for processing huge amounts of data on a single PC.This structure provides a successful composition for integrated storage. A blocked data separation mechanism for splitting the huge tensors into small blocks and compress them to store. The retrieval of data from stored function is done by RTree. RTree enhance the performance of retrieval and updating.

5. Future Work

Ongoing works include: 1) resolve the stratagem for finding best block splitting and rank purpose in accordance with data distribution, 2) improving the search and update method by using enhanced RTree. The individual canvasser cannot simply keep up with the prose in this domain. Up to our awareness this is the first method with HTR. This set up will be an efficient one. The tree could be extended by several instructions, so that the efficiency can be improved in future.

Reference

1. A. Aji, F. Wang, H. Vo, R. Lee, Q. Liu, X. Zhang, and J. Saltz, “Hadoop gis: A high performance spatial data warehousing system over map reduce,” Proc. VLDB Endowment, vol. 6, no. 11, pp. 1009–1020, 2013.
2. C. Lubich, T. Rohwedder, R. Schneider, and B. Vandereycken, “Dynamical approximation by hierarchical tucker and tensortrain tensors,” SIAM J. Matrix Anal. Appl., vol. 34, no. 2, pp. 470– 494, 2013.
3. L. Grasedyck, D. Kressner, and C. Tobler, “A literature survey of low-rank tensor approximation techniques,” GAMM-Mitteilungen, vol. 36, no. 1, pp. 53–78, 2013.
4. T. L. L. Siqueira, C. D. de Aguiar Ciferri, V. C. Times, and R. R. Ciferri, “The sb-index and the hsb-index: Efficient indices for spatial data warehouses,” Geoinformat., vol. 16, no. 1, pp. 165–205, 2012.
5. G. Cugola and A. Margara, “Processing flows of information: From data stream to complex event processing,” ACM Comput. Surv., vol. 44, no. 3, pp. 15:1–15:62, 2012.
6. M. L. Yiu, H. Lu, N. Mamoulis, and M. Vaitis, “Ranking spatial data by quality preferences,” IEEE Trans. Knowl. Data Eng., vol. 23, no. 3, pp. 433–446, Mar. 2011.
7. S. J. Kazemitabar, F. Banaei-Kashani, and D. McLeod, “Geostreaming in cloud,” inProc.2ndACMSIGSPATIALInt. WorkshopGeoStreaming, 2011, pp.3–9.
8. H. Plattner and A. Zeier, In-Memory Data Management: An Inflection Point for Enterprise Applications. New York, NY, USA: Springer, 2011.
9. I. Oseledets and E. Tyrtyshnikov, “TT-cross approximation for multidimensional arrays,” Linear Algebra Appl., vol. 432, no. 1, pp. 70–88, 2010.
10. I. Arad and Z. Landau, “Quantum computation and the evaluation of tensor networks,” SIAM J. Computer., vol. 39, no. 7, pp. 3089–3121, 2010.