IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 1, Issue 6, Nov 2013, 15-22
© Impact Journals

AN APPROACH TO STORE SPATIAL BIG-DATA USING MULTI-VALUED DATABASE

SUCHITRA REYYA, T. SUMALLIKA & G. V. M. VASUKI

Assistant Professor, Department of Computer Science and Engineering, Lendi Institute of Engineering & Technology, Andhra Pradesh, India

ABSTRACT

Spatial data has come to play an increasingly prominent role. Structured and unstructured data collected from diverse sources and used as an ensemble to derive information is referred to as Big Data. The size, variety, and update rate of mobility datasets exceed the capacity of commonly used spatial computing and spatial database technologies to learn, manage, and process the data with reasonable effort. Such data is known as Spatial Big Data (SBD). SBD has transformative potential. We formally define spatial big data preference queries and nested integrated inverted indexing technique (nI^3). S2I index are an extension of the R-tree for efficient spatial search. As the data representing in the R-Tree, IR-Tree, S2I index, Integrated Inverted Index (I^3) are in relation database (1st Normal Form), where this can occupy more storage space. To give an efficient reduction of storage space, we proposed nested relational database on integrated inverted index (nI^3) to reduce the space complication and elimination of redundancy.

KEYWORDS: Spatial Big Data (SBD), Nested Integrated Inverted Index (Ni^3), S2I Index

INTRODUCTION

Spatial database is a collection of spatially referenced data that acts as a database, which is in model of reality and the sense that the database represents a selected set or approximation of phenomena. These selected phenomena are deemed important enough to represent in digital form. The digital representation might be for some past, present or future time period, which stores in relational databases. The spatial database includes geographical database (GBD) to store model reality in relational databases.

The Geographic databases (GDB) [1] are simply databases containing geographic data for a particular area and subject. Spatial database systems offer the underlying database technology for geographic information systems and geographic databases. The model reality of spatial big data includes geographical databases in relational databases (1st Normal Form).

Both the number and the size of spatial databases are rapidly growing because of the large amount of data obtained from satellite images, X-ray crystallography or other scientific equipment. This growth by far exceeds human capacities to analyze the databases in order to find implicit regularities, rules or clusters hidden in the data. Therefore, automated knowledge discovery becomes more and more important in spatial databases. Knowledge discovery in databases (KDD) is the non-trivial extraction of implicit, previously unknown, and potentially useful information from databases (Frawley , Piatetsky-Shapiro & Matheus 1991). So far, most of the KDD methods have been based on relational database systems which are appropriate to handle non-spatial data.

In this paper, we use the multi-valued attribute to decompose the data space into the cells and propose a nested integrated inverted index, named nI^3 [2], nI^3 inserts data using multi-valued attributes [1] to reduce redundancy and disk space. Our index stores keyword cell as the basic unit, which captures spatial locality for a keyword. A keyword cell, denoted by *(w_i,C_j)*, refers to a list of documents containing keyword w_i and having their associated locations in cell C_j. Moreover, nI^3 stores multivalued information of keyword cell for effective pruning. The summary information includes a signature file which aggregates document id in the keyword cell and the upper bound score of keyword relevance. Based on the summary information, nested integrated inverted index has the following advantages:

- nI^3 takes much less update cost than existing methods and is more suitable for big data scenarios.
- nI^3 is effective in storage utilization.

The active techniques are R-trees[3], IR-Tree[1], S2I index[2], which are the tree data structures used for spatial access methods, i.e., for indexing multi-dimensional information such as geographical coordinates, rectangles or polygons. These techniques generate databases which are in relation database (1st Normal Form) leads to an excess storage space on disk.

The remaining paper is organized as, related work, introduction to conversion of relation database into nested integrated inverted index (nI^3) database to decrease storage space, methodology to decrease storage space and elimination of redundancy, experimental analysis and conclusion of the paper.

RELATED WORK

The R-tree [3], which was proposed by Antonin Guttman in 1984 and has found significant use in both research and real-world applications. R-tree is a height-balanced tree similar to a B-tree with index records in its leaf nodes containing pointers to data objects nodes correspond to disk pages. If the index is disk-resident, and the structure is designed so that a spatial search requires visiting only a small number of nodes. The index is completely dynamic, inserts and deletes can be intermixed with searches and no periodic reorganization is required.

A spatial database consists of a collection of tuples representing spatial objects, and each tuple has a unique identifier which can be used to retrieve it leaf nodes in an R-tree contain index record entries of the form. The uncertainty of the R- Tree database storage is in such way that which gives the group of values, which is shown in the Table 1.

Table 1: Nearest Group of Values (R-Tree)

City	Street	Near Services
MH	Ave	Restaurant1
MH	Ave	Restaurant2
MH	Ave	Restaurant3
MH	Ave	Restaurant4
MH	Str Ave	Restaurant1
MH	Str Ave	Restaurant2

R-tree extended with inverted files is known as IR-Tree. Here IR-tree's search for the nearest single object which is located in an area. IR tree is a tree data structure which is used as an index to handle location based queries. IR tree is designed such that it performs spatial clustering first and then textual filtering. Here first spatial filtering is done so that search space can be abridged because there may be many documents that are textually related but only very few of those are bounded within spatial scope. Now textual filtering is done so as to reduce search cost.

Finally, the joint relevance and ranking is done simultaneously such that, as soon as top k (the number of documents to be retrieved) documents are obtained the search process stops. IR-Tree is designed to perform spatial filtering, textual filtering, relevance computation, and ranking simultaneously. Even storage and access overheads are considered.

The minimum distance of the IR-Tree finds out the exact near services in an area basing on the spatial object location. Table 2 shows the nearest services in an area with distance. Basing on the IR-Tree the Table 2 can find the exact nearest distance (minimum distance) location which shown in Table 3.

Table 2: Nearest Group of Values with Distance

City	Street	Near Services	Distance (KM)
MH	Ave	Restaurant1	3
MH	Ave	Restaurant2	5
MH	Ave	Restaurant3	4
MH	Ave	Restaurant4	2
MH	Str Ave	Restaurant1	1
MH	Str Ave	Restaurant2	4

Table 3: Exact Nearest Value of Object Using Minimum Distance Equation (IR-Tree)

City	Street	Near Services	Distance (KM)
MH	Ave	Restaurant4	2
MH	Str Ave	Restaurant1	1

Recently, S2I Index [2] was proposed for more efficient spatial keyword search. It partitions the spatial database first by the textual attribute. If a keyword is infrequent, all the elements in the inverted list are stored sequentially for efficient retrieval to save Input/Output cost. Otherwise, an aggregated R-tree is built for spatial pruning. S2I is scalable to the number of keywords in the database because given a set of query keywords, only the related inverted lists will be accessed. However, it is difficult to do spatial aggregation across different R-trees.

Integrated Inverted Index (I^3) [2] is an efficient index for spatial keyword search is required to support both spatial pruning and textual pruning simultaneously. Existing solutions prefer the combination of R-tree and inverted list. However, those hybrid indexes are not scalable and require high maintenance cost. Our proposed nI^3 adopts textual partition first just like the I^3 index. We discard R-tree and use multi-valued attribute to split the space into a hierarchy of cells. The basic unit in our index is named keyword cell which captures spatial locality for a keyword.

Besides the traditional spatial keyword search problem, variants of the topic have been proposed. One is to allow the query key-words to appear in multiple documents or collective spatial keyword.

ELIMINATION OF REDUNDANCY & STORAGE SPACE IN SPATIAL-BIG DATA

This paper introduces multi-valued[1] data in spatial big databases, which can give efficiency in data storage and elimination of redundancy using nested Integrated Inverted Index (nI^3)[2].

Definition of Multi-Valued

A multi-value attribute is the practice of maintaining more than a single value in a database column.

Our contribution is to convert first normal form database into nesting (nested Integrated Inverted Index) database using spatial big database.

Definition of First Normal Form

A relation is said to be in First Normal Form (1st Normal Form) if and only if each attribute of the relation is atomic. More simply, to be in 1st Normal Form, each column must contain only a single value and each row must contain the same number of columns.

Definition of Spatial Database

Spatial database system as a database system that offers spatial data types in its data model and query language and supports spatial data types in its implementation.

Definition of Spatial Big Data

Spatial datasets are exceeding the capacity of current computing systems to manage, process, or to analyze the data with reasonable effort due to volume, velocity and variety**.**

Table 4: Relational Database Conversion into Nested Integrated Inverted Index Database

City	Street	Near Services
MH	Ave	{Restaurant1, Restaurant2, Restaurant3, Restaurant4}
MH	Str Ave	{ Restaurant1, Restaurant2}

Table 1 database representing single-valued information, but the data relation is in multi-valued and leads to redundancy and more storage space i.e., CITY: MH and STREET: Ave has representing more than one Restaurant as a unique restaurant name. This gives the row wise data as a unique, and column wise data is in redundant. Our intent is to reduce the redundancy in the particular column. While reducing redundancy in particular column the database storage is also effects, which is shown in Table 4. The Table 1 having 6 rows and Table 4 reduced those 6 rows into 2 rows by using relational database to nested Integrated Inverted Index relational query.

Using nI^3, we propose spatial big data which can reduce disk space and redundancy when assembling of multiple similar databases from different sources.

The below graph shows the big data using spatial attributes of geographic object types, represented by shape in Figure 1, have intrinsic spatial relationships (e.g. close, far, contains, intersects). Because of these relationships real world entities can affect the behaviour of other features in the neighbourhood. This makes spatial relationships be the main characteristic of geographic data to be considered for data mining and knowledge discovery. It is also the main characteristic which differs geographic/spatial data mining from non-spatial data mining. The rectangular shape in the Figure 1, containing a relationship between city, street and near objects (eg., restaurants, cinema theatres, hospitals, gas stations, consultancy etc.,). The shape containing geographic data will be stored in relational databases, which shown in Table 1, Table 5 and Table 6. The Table 5 and Table 6, both store the information of the relational database surrounding around 3km from the city MH and PHI respectively. From the city (MH) the surrounded 3km have the services of other city (PHI) and vice versa. In the below tables the similar tuples are shown as intersected tuples around 3km near to cities.

Table 5: Relational Database around 3km from a City (MH)

City	Street	Near Services
MH	Ave	Restaurant 2
MH	Ave	Restaurant 3
MH	Ave	Restaurant 4
MH	Ave	Restaurant 5
MH	Str Ave	Gas station 1
MH	Str Ave	Gas station 2
PHI	Pennsylvania Ave	Consultancy 1
PHI	Pennsylvania Ave	Consultancy 2

Table 6: Relational Database around 3km from City (PHI)

City	Street	Near Services
PHI	Oak ave	Hotel 1

Table 6: Contd.,

PHI	Oak ave	Hospital 1
PHI	Oak ave	Hospital 2
PHI	Pennsylvania ave	Consultancy 1
PHI	Pennsylvania ave	Consultancy 1
MH	Elm str Ave	Gas station 1
MH	Elm str Ave	Gas station 2

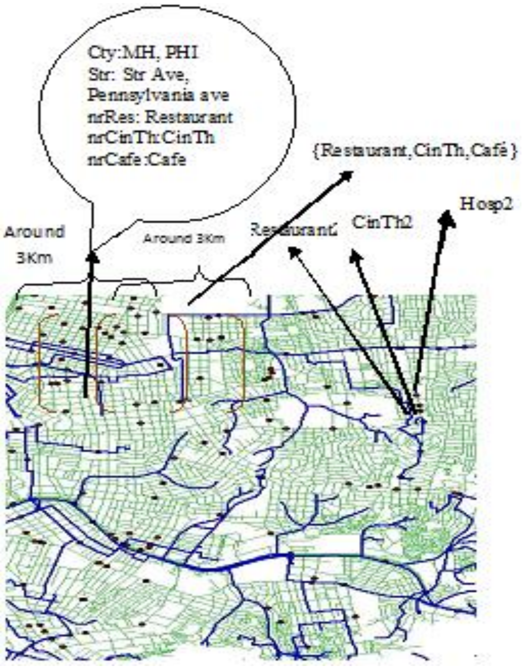

Figure 1: Geometric Data Using MAP Representation

METHODOLOGY

The query conversion is in relational database to nested integrated inverted index database to reduce redundancy and disk space in spatial big data.

Query processed using conversion of Relational database into Nested Integrated Inverted Index (nI^3) database.

Consider the relation (U={Ci, St, NS}, R) in Table 1.

Nest$_c$(U, R)=({Ci, St, NS' }, R$_x$), where NS' is a multi-valued data of NS, Rx is a nested relation which is shown in Table 4.

Suppose we have the relation schema 1st Normal Form Relations with the attributes City(Ci), Street(St) and Near Services (NS). The purpose of this databases is to record the information of near services for a particular city and street.

The query cited below implements the single nesting using nested Integrated Inverted Index. It takes Table1, Table5 and Table 6 as input and produce Table7 as the output. We can clearly observe the reduction of redundancy in the attribute Near Services (NS) after the Query result.

```
Select distinct Ci,St,NS[(select NS from f table
        where Ci=f.Ci and St=f.St
        group by NS)] from ftable f
        group by Ci,St;
```

Table 7: Nested Relational Database (Ni^3) for Near Services Gives Spatial Big Database

City(Ci)	Street(St)	Near Services (NS)
MH	Ave	{Restaurant1, Restaurant2, Restaurant3, Restaurant4, Restaurant5, Cinema Theatre, Cafe}
MH	Elm Str Ave	{Restaurant1, Restaurant2, Gas station1, Gas station2}
PHI	Oak Ave	{Restaurant1, Hospital1, Hospital2}
PHI	Pennsylvania Ave	{Consultancy1, Consultancy2}

EXPERIMENTAL ANALYSIS

In this section, we represented our result about the performance of our schemes for eliminating redundancies over nested Integrated Inverted Index (nI^3) database. For the experiments, we used postgres SQL on Windows 7 with 2.40 GHz Intel Core I3 CPU with 4GB of RAM.

Our experiments used Synthetic Database with a few synthetic attributes addition. There are two alternative evaluation strategies for the comparison of relational database and nested integrated inverted index database. 1. Time calculation Graph. 2. Size calculation Graph.

Time Calculation Graph: We studied the impact of reduced redundancy in the nested integrated inverted index by varying time. In this we considered number of rows whose value varied from 50 to 700 rows can be observed in Table 8 and Figure 2.

Size calculation Graph: Here we investigated how the redundancy is eliminated by considering the space occupied by the relational database and nested integrated inverted index database on the disk by varying number of rows can be seen in Table 9 and Figure 3.

Table 8: Time Calculation for Execution

No. of Rows	Relational Database (msec)	nI^3 Database (msec)
50	0.70	0.21
100	1.49	0.45
300	1.786	0.545
500	2.218	0.67
700	3.01	1.78

Table 9: Size Calculation for Execution

No. of Rows	Relational Data Size (KB)	nI^3 Database Size (KB)
50	20	10
100	44	12
300	52	23
500	70	30
700	90	45

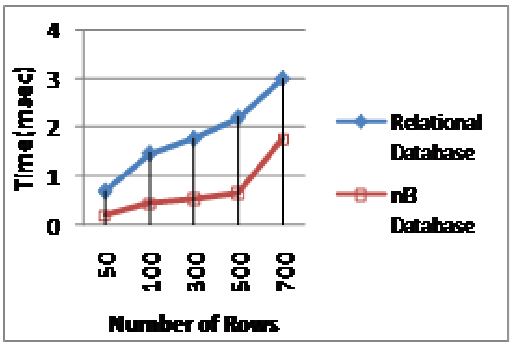

Figure 2: Time Calculation Graph

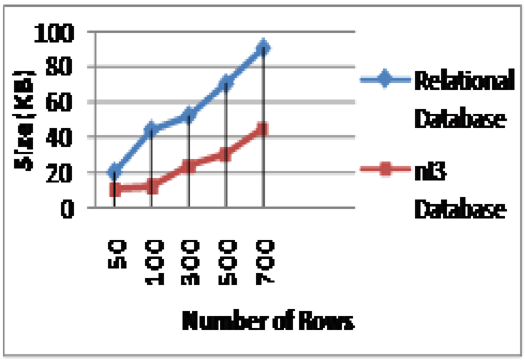

Figure 3: Size Calculation Graph

CONCLUSIONS

We presented that, the spatial big databases using nested integrated inverted index database proved to be good in eliminating redundancy on large databases. We considered a single attribute for single nesting and showed the experiments resulting in space and time saving of nested integrated inverted index databases over relational databases.

There is much more to be done. First, to apply double nesting on large databases. Second, reducing redundancy on large databases using double nesting.

ACKNOWLEDGEMENTS

We express our sincere and profound gratitude to our principal Dr. V.V.Rama Reddy, Management Members our chairman Sri P.Madhusudhana Rao, Vice-chairman Sri P.Srinivasa Rao and Secretary Sri K.Siva Rama Krishnan for their valuable support and guidance. We also thank to Prof. A. Rama Rao for their continuous support.

REFERENCES

1. Suchitra Reyya, M. Sundara Babu, Ratnam Dodda , “An Efficient Storage of Spatial Database using Nested Relational Database” International Journal of Engineering Research and Technology (IJERT), ISSN: 2278 – 0181, Vol. 1 Issue 7, September – 2012
2. Dongxiang Zhang, Kian-Lee Tan, Anthony K. H. Tung, “Scalable Top-K Spatial Keyword Search”, EDBT '13 Proceedings of the 16th International Conference on Extending Database Technology, Pages 359-370
3. Antonin Guttman, “R-trees: a dynamic index structure for spatial searching” Proceeding SIGMOD '84, Proceedings of the 1984 ACM SIGMOD international conference on Management of data pp 47 - 57
4. D. Hari Krishna, Ch. Sowjanya, P. Radhakrishna, “Quality Preferences by using H.2.4.k Spatial Databases”, IJCST Vol. 3, Issue 1, Spl. 5, Jan. - MarCh 2012Hahn, S. K., Terry, E. R., Leuschner, K., Akobundu, I. O., Okali, C. and Lal, R. (1979). Cassava improvement in Africa. Field Crops Res (2):193-226.
5. Korth, H., Roth, M.: “Query languages for Nested Relational Databases”, Nested Relations and Complex Objects in Databases, Springer, (1991).

AUTHOR'S DETAILS

Suchitra Reyya received her B.Tech degree from Jawahar Lal Nehru Technological University Hyderabad, Andhra Pradesh, India, in 2008 and the M.Tech degree from Jawahar Lal Nehru Technological University Kakinada, Andhra Pradesh, India in 2011. She was an Assistant Professor, in the Department of Computer science and Engineering, VITAM Engineering College. She was an Assistant Professor in the Department of Computer Science & Engineering, Pydah College of Engineering & Technology. Currently working as an Assistant Professor in the Department of Computer Science & Engineering, Lendi Institute of Engineering & Technology. She has 4.5 years of experience in teaching. She published 4 International Journals, two International Conferences. She is heading special interest research groups in Web technology, Oops through Java, Data mining, Data Pre-processing.

T.Sumallika received her B.Tech degree from Andhra University, Andhra Pradesh, India, in 2004 and the M.Tech degree from JNTU Kakinada, Andhra Pradesh, India, in 2011. She was an Assistant Professor, in the Department of Computer science and Engineering, AL-AMEER College of engineering and technology. She was an Assistant Professor in the Department of Computer Science & Engineering, Pydah College of Engineering & Technology. Currently working as an Assistant Professor in the Department of Computer Science & Engineering, Lendi Institute of Engineering & Technology. She has 5 years of experience in teaching. She is heading special interest research groups in Web technology, Oops through Java and Data mining.

G.V.M. Vasuki received her B.Tech degree from Jawahar Lal Nehru Technological University Hyderabad, Andhra Pradesh, India, in 2005 and the M.Tech degree from Jawahar Lal Nehru Technological University Kakinada, Andhrapradesh, India, 2012. She was an Assistant Professor, in the Department of Computer science and Engineering, GMRIT. She was an Assistant Professor in the Department of Computer Science & Engineering,JNTU Kakinada. Currently working as an Assistant Professor in the Department of Computer Science & Engineering, Lendi Institute of Engineering & Technology. She has 7 years of experience in teaching. She is heading special interest research groups in Operating Systems, Cloud Computing, and Data Analytics.