



A Frame Work for Spatial Query Processing on Spatial Database

T.Aruna Sri¹, Ch. Sanjay², Sudarson Jena²
Dept. of C.S.E¹, Dept. of Mechanical Engg.², Dept. of IT²
GITAM University, Hyderabad

Abstract: Spatial database is a database that is optimized to store and query data. In order to process spatial database a set of functions are needed to process spatial data types called geometry or feature. The main purpose of this framework is to adapt SQL queries and provide the best result among the available techniques of spatial query processing. Spatial objects are stored in the spatial database, which are expressed by spatial data and attribute data. Spatial data depicts the information about the location and shape of the spatial data, etc. Attribute data also expresses the non-spatial information about name and special attributes of the spatial object, etc. As the high volume of the spatial data and the complexity of spatial query and spatial objects, it is a hot topic of spatial database application to optimize the spatial query.

Keywords: Spatial data ; Spatial objects; Spatial database; Spatial Join; Spatial Index.

I. INTRODUCTION

Spatial databases are databases that store spatial data or, in other words, data related to spaces in the physical world, parts of living organisms, engineering design and many other spaces of interest. The data in a spatial database is often captured, initially, in the form of digital images, so spatial databases are sometimes called pictorial or image databases.

Spatial database system [4] may be defined as a database system that offers spatial data types in its data model and query language, and supports spatial data types in its implementation, providing at least spatial indexing and spatial join methods.

A spatial database is related to objects in space, including points, lines and polygons typical databases grasps various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types. Database systems use indexes to quickly look up values and the way that most databases index data is not optimal for spatial queries. Instead, spatial databases use a spatial index to speed up database operations.

The following query types and many more are supported by the Open Geospatial Consortium:

- Spatial Measurements: Finds the distance between points, polygon area, etc.
- Spatial Functions: Modify existing features to create new ones, for example by providing a buffer around them, intersecting features, etc.
- Spatial Predicates: Allows true/false queries such as 'is there a residence located within a mile of the area we are planning to build the landfill.
- Constructor Functions: Creates new features with an SQL query specifying the vertices (points of nodes) which can make up lines. If the first and last vertex of a line is identical the feature can also be of the type polygon (a closed line).
- Observer Functions: Queries which return specific information about a feature such as the location of the center of a circle.

As characteristic of the spatial itself, geometrical operation costs are vast in time and space. Escaping or

decreasing geometrical operation becomes a springboard for optimizing query processes. So, it is an important content for spatial query optimization to optimize query handling algorithms, and improve query handling efficiency.

II. FRAMEWORK

A. Spatial Database:

A spatial database must typically handle large collections of geometric objects, such as hundreds of thousands or millions of polygons. In particular, a spatial database must be able to retrieve a selection of objects, without scanning every object in the database. This means that spatial databases must use spatial indexing -- a numbering system used to select elements from a list -- and employ an efficient mathematical formula, or algorithm, for connecting, or joining, spatial objects from different classes together.

B. Spatial Join:

Spatial join operation is used to combine two or more datasets with respect to a spatial predicate. The predicate can be a combination of directional, distance, and topological spatial relations. In the case of non spatial joins, the joining attributes must be of the same type, but for spatial joins they can be of different types. Usually each spatial attribute is represented by its minimum bounding rectangles (MBR).

The first known technique to solve the spatial join operation was a grid based technique developed by Orenstein in 1986. The technique uses a grid to divide multidimensional spaces into smaller blocks known as pixels. A z-ordering [7] is applied to order the pixels. Each object is approximated by the pixels which intersect with its MBR. As pixels are ordered by z -ordering, each object is represented by a set of z-values, which are one-dimensional. Any one-dimensional index (e.g., B+-tree) can then be used to sort them and using sort-merge spatial join operation is done. The performance of this technique depends on the granularity of the grids.

C. R-Trees:

R-tree is an index structure [6] that was defined by Guttman in 1984. The leaf nodes of this multiway tree contains entries of the form: (I, object-id), where object-id is a pointer to a data object and I is a bounding box (or an axes-parallel minimal bounding rectangle, MBR). The data object can be of any type: point, polyline, or polygon. The internal nodes contain entries of the form: (I, child-pointer), where Child-pointer is a pointer to a child and I is the MBR of that child. The maximum number of entries in each node is called the branching factor M and is chosen to suit paging and disk I/O buffering. The Insert and Delete algorithms assure that the number of entries in each node remains between m and M, where $m \leq [M/2]$ is the minimum number of entries per node. An advantage of the R-tree is that pointers to complete objects (e.g. polygons) are stored; so the objects are never fragmented.

D. Spatial Query Processing:

A spatial join involves two steps. Filter Step and Refine are shown in figure 1.

- Tuples whose MBRs overlaps with a query region MBR are determined. This step is not computationally expensive since at most four computations are required to determine whether two rectangles intersect. This step is known as filter step.
- The tuples that passed the filter step are fed to the refinement step, where the exact spatial representation is used and a spatial predicate is checked on these spatial representations. The refinement step is computationally expensive, but the number of tuples it processes in this step is less, due to the initial filter step.

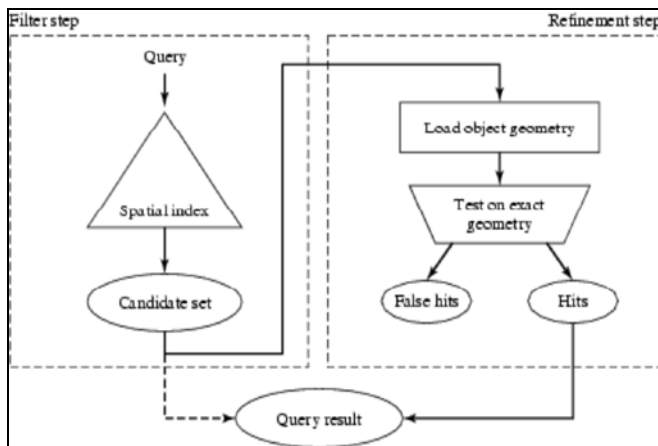


Figure 1 Filter and Refine step

III. ALGORITHM

The main steps of our spatial-join algorithm, starting from the roots of the two R-trees, Spatial-join descends down the tree level by level. At each level, some Convergence Pair Node CPN instances are taken one by one from (Breadth first search Queue)SJQ to check all the child nodes of the two nodes in an CPN instance in pair. If a pair of child nodes overlaps, they first must be share-locked; then they are saved as a CPN instance to SJQ for checking overlap at the next level. When all the child nodes of the two parent nodes in a CPN instance have been checked, the two share-locked parent nodes must be share-unlocked immediately to make the other broke operations recover to

continue if these operations try to lock one of the two parent nodes before.

The procedure of taking a CPN instance, checking pairs of child nodes, share-locking two overlap child nodes, saving the two overlap child nodes as a CPN instance to SJQ, and releasing the two locked parent nodes is repeated until SJQ is empty. When the both nodes in a CPN instance are leaf nodes, the children of the two leaf nodes are spatial objects. If two objects belonging to two different R-trees overlap, both objects are returned to users rather than saving to SJQ. A pointer variable "Current" is used to point to a CPN instance currently being processed. A procedure "mqueue" is used to add a CPN structure to SJQ. A function "get-queue" is used to get a CPN instance form SJQ.

```

Algorithm Spatial-join(A,B)
//Give two R-trees A and B, find some pairs
of objects in A and B, respectively, where each pair
of objects has an overlap condition.//
Input: two R-trees A and B
Output: some pairs of overlap objects
Begin
SJQ: a queue of OPN;
Current: a pointer of CPN;
reset SJQ;
share-lock(A);
share-lock(B);
mqueue(A, 0, B, 0, SJQ);
While SJQ is not empty, do
Current <- get-queue (SJQ);
Check-intersect (Current .α,Current .A,Current .β, Current .B);
end while;
end Spatial-join.
    
```

Figure 2 Spatial Join Algorithm

IV. METHODOLOGY

GUI involves selecting the spatial datasets which are stored at backend using postgresql. To store the spatial data sets we have to import shape files using postgis. we can view those files using shape viewer All spatial data has an extensions .shp .We have an GUI interface which uses java script and we have used java swings to implement the interaction of client and server. spatial GUI has shown in Figure 3 given below .

Components of this Framework used are:

- Java Swings- jdk1.6.0_22, jre6.
- Post Gis-PostGIS Shape file and DBF Loader.
- Postgre sql- postgresql-8.4-701.jdbc4.jar
- J free charts

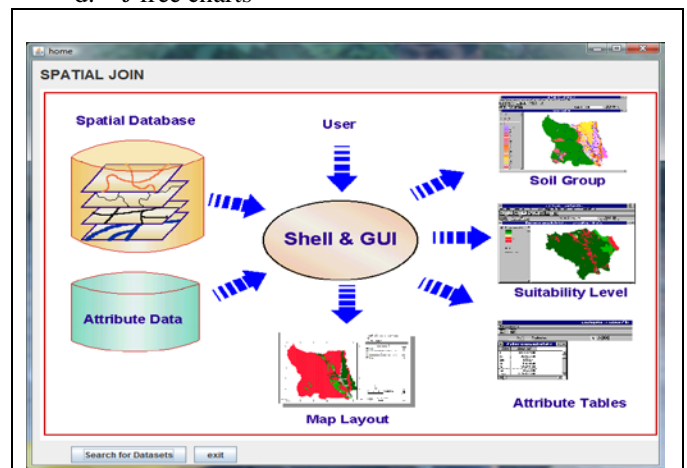


Figure 3 Spatial User Interface

Spatial data can be stored in shape files. For implementing spatial join, we use shape viewer to view data in maps. Selecting two different spatial datasets and applying spatial join algorithm on different spatial datasets as shown in Fig 4 spatial query Performance has measured by using Jcharts for selected datasets) .

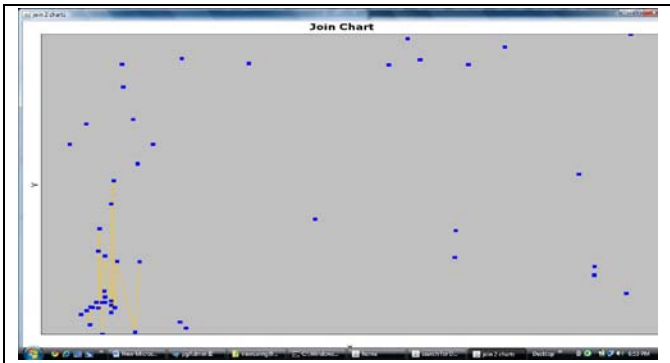


Figure 4 Spatial Join between two data sets

Spatial query Performance has measured by using Jcharts for selected datasets) .we can also compare query performance for different types of spatial data States, Places, and Railways as shown in the Fig 5.

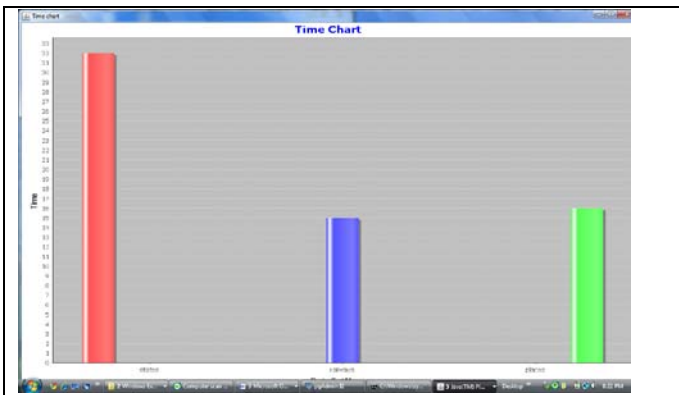


Figure 5 Spatial Query Performance

V. CONCLUSION

In this paper, we have presented a new method for expanding R-trees to better handle spatial Join algorithm are used for finding out intersecting regions (spatial join)..We can compare or improve the performance of spatial query choosing different spatial index..So that we can increase the query response time.

VI. REFERNCES

- [1]. Chang-Tien, Jing Dai, Ying Jin and Jank Mathurai, GLIP: A Concurrency Control Protocol for Clipping Indexing, vol. 21 no. 5. pp 714-728 : May 2009 .
- [2]. J. Dai, C.-T. Lu, and L.-F. Lai, A Concurrency Control Protocol for Continuously Monitoring Moving Objects, in Proc.InternationalConference on Mobile Data Management. pp 132-141: 2009.
- [3]. R. Laurini, and D. Thomson, Fundamentals of Spatial Information Systems, London: Academic Press, 1992.vol 49 pp 67-104: 2004.
- [4]. Orenstein J. A, Spatiul Query Processing in an Object-Oriented Database System, Proc. ACM SIGMOD Int. Conf. on Management of Data Washington D.C.: pp 326-333, 1986.
- [5]. Borrough P. A, Principles of Geographical Information Systems for Land Resources Assessment : Oxford niversity Press, 1986.
- [6]. Y. W. Hung, N. Jing and E. A. Rundesteiner, BFRJ global optimization of spatial joins using R-trees, Dept. of Computer Science, Worcester Polytechnic Institute, Tech. Report. WPI-CS-TR-97-5: January, 1997.
- [7]. J. M. Patel and D. J. DeWitt, Partition based spatial-merge join, Proc.ACM SIGMOD Int. Conf. on Management of Data. Washington D.C.pp 259-270: 1996.