An Approach For Spatial Query Processing In Location Based Services For Mobile Environment

Dhikhi T[#1], Kapilya G[#2], Vanathi R[#3]
[#]Department of Computer Science and Engineering, Saveetha School of Engineering,
Saveetha University, Thandalam, Chennai
[1]dhikhi@gmail.com
[2]kapilya@gmail.com
[3]vanathi.be@gmail.com

***Abstract---* The paper is proposed to provide users of mobile devices personalized services tailored to their current location. The main research in the paper concerns with the spatial query processing and mobile location based service delivery. The paper relays on geocoding service which performs as a geocoder by determining a geographic position, such as a place name, street address or postal code. The geocoding process is done by giving geographic positions with geographic co-ordinates latitude and longitude, other points and features where the features can be mapped and entered into Geographic Information Systems. The validity region of a query is returned to the user along with the query result in which validity region indicates the geographic area within which the result of the query remains valid. Spatial data occupies a spatial extent, which stores location and boundary in the spatial database.**

Keywords - Spatial mining, LBS, R* tree, Nearest Neighbor Query processing, Caching.

I. Introduction

Location-based queries are of interest in plenty of applications [18], geographical information systems can be an example. The approach is proposed here that leverages short-range, ad hoc networks to share information in a peer-to-peer (P2P) manner among mobile clients to answer location-based nearest neighbor queries. The technique can reduce the access traffic to remote servers by a significant amount with minimal caching at peers. For example, if two mobile hosts are close to each other, the result sets of their queries for a specific object type may overlap significantly. Through mobile cooperative caching [5] of the result sets, the mobile clients can efficiently share the query results. An efficient implementation of spatial selections is an important requirement for good overall performance of complete geographic database system. Location-based services (LBS) provide users of mobile devices personalized services tailored to their current location. The paper deals with the spatial query processing technique in a broadcast environment followed by location-based spatial queries processing in a demand access mode. The research issues on twochallenges like Scheduling Spatial Queries Processing, Monitoring Continuous Queries on Moving Objects.

In LBSs the result of a query is only valid for a particular user location because users are moving frequently; new queries need to be sent to the server whenever there is a location change that creates high network transfer and server processing cost. To overcome this issue, the concept of validity regioncan be used. The validity region is a rectangle area that shows the geographic area within which the result of the query remains valid and is given back to the user together with the result. Then checks for a new query that is issued by verifying whether it is the part of the validity region. The utilization of validity regiondecreases significantly the amount of new queries issued to the server and thus the communication via the wireless channel.

II. RELATED WORK

The related work pertinent to our approach can broadly be classified into two areas, namely nearest neighbor query processing and cache management in mobile environments.

A. Nearest Neighbor Query Processing

Thefundamental operations performed on spatial data sets is to find the nearest neighbors of a query object – either one (1NN) or a specific number k (kNN) [19].R trees [8]and their derivatives have been a prevalent method to index spatial data and increase query performance. Roussopoulos et al. [13] proposed a branch-and-bound algorithm to find the nearest neighbors that searches an R-tree in a depth-first mode. The best-first NN algorithm suggested by Gisli et al. [9] retains a heap with the entries of the nodes visited so far and it always expands the first entry in the heap. With the emergence of mobile devices attention has focused on the problem of continuously finding k nearest neighbors for moving query points (k-NNMP)[14]. A naive approach might be to continuously issue kNN queries along the route of a moving object (multi-step search). This solution results in repeated server accesses and nearest neighbor computations and is therefore inefficient.

One method to reduce the computational complexity is to sample the trajectory instead of treating it as a continuous curve. In the early work, nearest neighbor searches were based on the Euclidean distance between the query object and the sites of concern. However, in many applications target cannot move freely in space but are constrained by a network (e.g., cars on roads, trains on tracks). Therefore, in a realistic environment the nearest neighbor computation must be based on the spatialnetwork distance, which is more expensive to work out. A number of techniques have been suggested to manage the complexity of this problem [11, 12].

B.Cache Management In Mobile Environments

Caching is a key technique to improve data retrieval performance in widely distributed environments. Mobile clients retrieve information from database servers via intermediate base-stations. With the increasing deployment of new peer-to-peer wireless communication technologies there exists a new information sharing alternative known as peer-to-peer cooperative Caching. Peer-to-peer cooperative caching can bring about several distinctive benefits to a mobile system: improving access response time, reducing server workload and palliating point-to-point channel congestion. As a disadvantage, it may increase the communication overheads among mobile hosts.

The Cooperative Caching (COCA) [5] scheme investigates the effects of client activity levels, data replication, and cache size. The advantages of clustering mobile clients into groups are investigated in [4].Semantic caching [16] stores additional information such as query descriptions with the data items in the local cache. The metadata is used to determine whether a new query is fully answerable from the cache. In that case no communication with the server is required. The cached index enables the objects to be reused for common types of queries [10].

III.PROPOSED PREDICTION MODE

In this paper efficient way of processing different spatial queries are proposed thereby solving several issues in spatial query processing. This paper also helps in reducing the workload of both server and client and the result will be obtained with minimum response time and best path with minimum cost. Mobile devices can only provide a computing environment with limited CPU, memory, network bandwidth, and battery resources. As such, mobile clients must be designed to balance the utilization between these resources and the loads between the client and the server. For example, promoting more computation on the client can reduce bandwidth consumption but increase CPU load and memory consumption. Given the rapidly increasing capability of mobile devices, mobile applications must make reasonable assumptions about the client's computational capability and be able to adapt to it.

Geospatial applications will be woven into the fabric of the information society. Location based services (LBS) as a promising research field brings new opportunities and challenges to geographic information accessing, sharing and disseminating. The terminal users can access to required information – largely around their locations and surroundings through pull and push mode. However, because of big variety of mobile devices, software of LBS has not normally been available for them. JAVA development environments render a suitable method to gather requested information in an appropriate way. In this project a new architecture of LBS based on JAVA with Web Server is proposed. These applications of LBS based on JAVA may be written once, and then placed on any server platform that supports the Enterprise Java Beans specification. Spatial selections [5,6] are of great importance within the set of spatial queries and operations. An efficient implementation of spatial selections is an important requirement for good overall performance of the complete geographic database systems.

A. R Tree*

R* trees are a variant of R-trees [8] which aim to improve performance. They are also a better indexing data structures for spatial joins (map overlay structure) which are one of the most important operations in geographic and environmental databases. The R-trees versions take into consideration only the area parameter while making their decisions and optimizing the tree performance. Besides area R* trees also take into consideration the margin and overlap parameters. R* tree is the most robust method .he overlap of an entry can be defined as the sum of the overlap of the particular entry with all the objects of a node. If Ei….Ep are the entries in a node and Ek needs to be inserted, Overlap (Ek) = S [i=1 to p &&i⇔k] area (Ek n Ei)

Algorithm ChooseSubtree

CS1 Set N to be the root

CS2 If N 1s a leaf,

return N

else

If the childpointers in N point to leaves [determine the minimum convergence cost], choose the entry in N whose rectangle demands least overlap enlargement to include the new data rectangle Resolve ties by choosing the entry whose rectangle needs least area elaboration, then the entry with the rectangle of limited area if the childpointers in N do not point to leaves [determine the minimum area cost], choose the entry m N whose rectangle needs least area enlargement to include the new data rectangle Resolve ties by choosing the entry with the rectangle of smallest area

end

CS3 Set N to be the childnode indicated to by the childpointer of the chosen entry and repeat from CS2.

For choosing the best non-leaf node, alternative methods did not outperform Guttman's original algorithm For the leaf nodes, minimizing the overlap performed slightly better. In this version, the cpu cost of determining the overlap is quadratic in the number of entries, because for each entry the overlap with all other entries of the node has to be calculated However, for large node sizes we can reduce the number of entries for which the calculation has to be done since for very distant rectangles the probability to yield the minimum overlap is very small Thus, m order to reduce the cpu cost, this part of the algorithm can be altered as follows [determine nearly minimum overlap cost] Sort the rectangles m N in increasing order of their area enlargement needed to include the new data rectangle

IV.EXPERIMENTAL EVALUATION

In our work, the client will query the spatial details of the airline. The data collection mainly relies on the airport destinations of the world and their connecting routes. The users can specify the starting place and the destination. Based on the details provided the spatial query processing will be done on the airway spatial data and the most suitable route for the users will be plotted in the users mobile. So when all these details are provided, then a list of flight available matching the given criteria will be listed to the users.

From the list of flight listed for the users, they can choose a flight and check for the available seats in the flight. So if the number of seats requested is available then, their personal information will be obtained for further processing, they have to provide with the name, age and sex of all the passengers traveling. If all the seats are not confirmed, then the number of passengers in the waiting list will also be displayed. So if they wish they can continue giving the personal information. In the rest of this section we describe the datasets and present the experimental results.

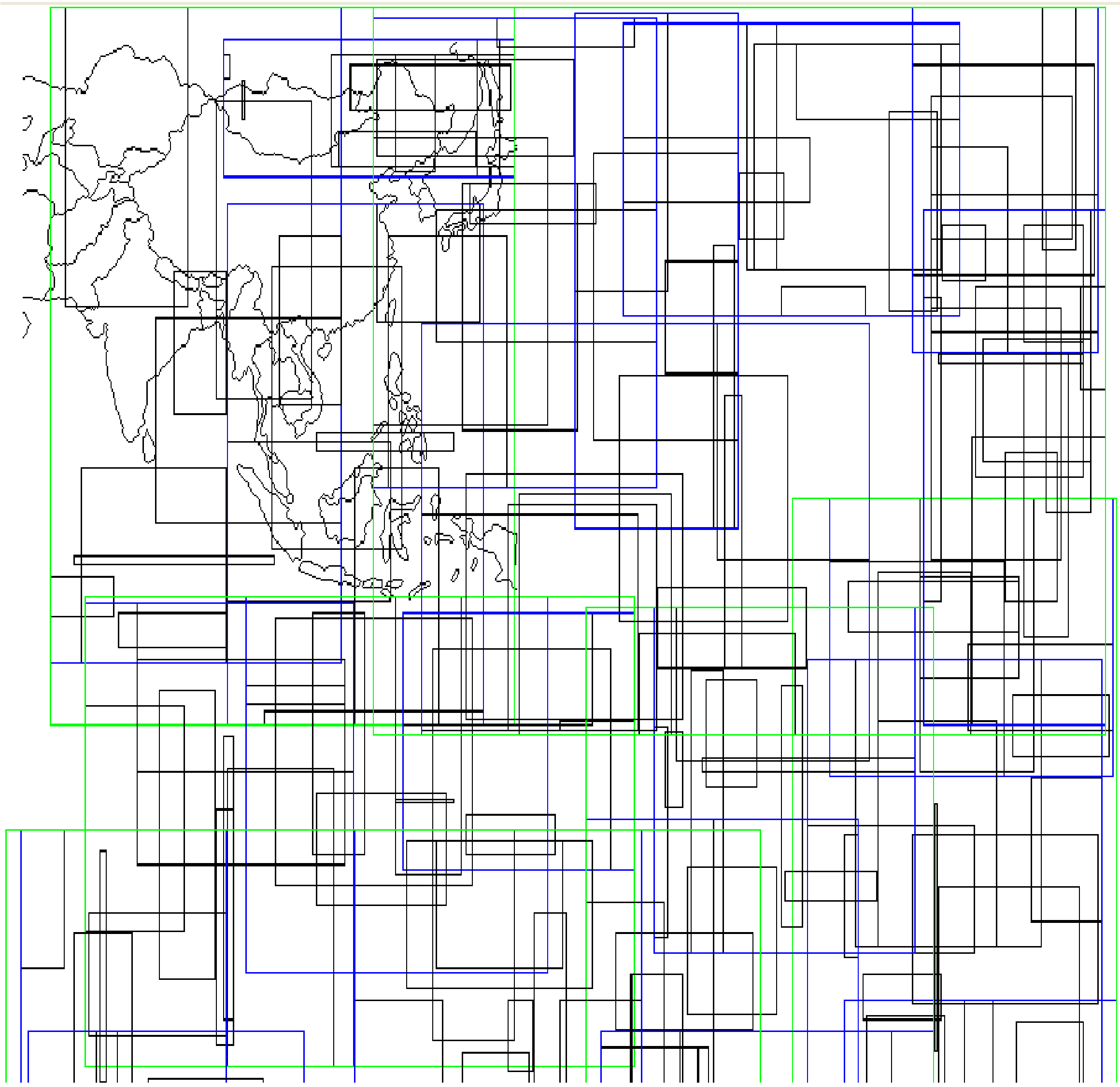

Figure 1.R*tree created

A. Datasets

The spatial database consists of all the details of the flights from the different places and their available seats and all the related necessary data. As evidence, we have the sample shown in Table 1.

B .Experimental Results

We define the different queries[3] like range query, point query, circle query, ring query, constraints query. Using R* tree [1, 7, 8] algorithm mobile clients can determine the validity of queries based on their current locations. This is made possible by server providing the validity region within the location of client in which the result remains the same, with the query result. A mobile client can select to and from cities for getting the flight details. The clients retrieve the information from the spatial database server and will be displayed in the mobile.

Figure2.Input Design

The server checks for the query issued from the client are already processed. If the server noticed that the nearest active client is having the result of the query in its cache, redirection[17] is done. One method is that the server will redirect to the nearest client having the result in its cache. The other method is that the server sends the nearest clients information that has the result in its cache to the client who issued the query. Then that query will be redirected by the client who issued the query directly to the client in which the result is cached by the information send by the server.

TABLE 1. DATASET

fnumber	destination	startingfrom	deptime	arrtime	aircrafttype	day
F001	Mumbai	Chennai	10.00	12.00	JETAIRWAYS	0
F002	Banglore	Chennai	11.00	11.45	KINGFISHER	1
F003	Jammu	Chennai	12.00	15.00	BRITISH AIRWAYS	2
F004	Delhi	Chennai	13.00	17.00	JETAIRWAYS	3
F005	Chennai	Delhi	14.00	18.00	JETAIRWAYS	4
F006	Chennai	Banglore	1.00	2.00	KINGFISHER	5
F007	Banglore	Chennai	3.00	4.00	KINGFISHER	6

We utilized R* tree for spatial indexing in the server module. The R* tree is more advance in query response time over conventional R tree algorithm by sophisticated insertion and node splitting methods that minimize a combination of overlap between bounding rectangles and the total area. Thus the client gets a most appropriate result for the query being given, that is the best path with minimum cost. Moreover through the mobile co-operative caching of result sets, mobile clients can share the query results efficiently. Figure 2 and 3 shows the input and output design of the paper. Figure 1 shows the R* tree with the different validity regions.

Figure3.Output Design

V.CONCLUSION

The main goal of the spatial query processor is to cut down the steps by preprocessing operations in the preceding steps which reduce the number of objects examined in an expensive step. Our Proposed approach addresses these challenges by providing efficient indexing for spatial query processing through J2EE distributed architecture and J2ME based client interface for mobile environment. This provides a map based interface for mutual testing of location based services and also provides less access time by caching. In future the map can be replaced with any image. It can be developed to other type of queries such as range query and context sensitive [18] in future.

REFERENCES

[1] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles" in Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 322–331, 1990.

[2] Brinkhoff T, Horn H, Kriegel H-P, Schneider R. "A Storage and Access Architecture for Efficient Query processing in Spatial Database Systems"Proc 3rd Symposium on Large Spatial Databases, Singapore, 1993.

[3] Brinkhoff T, Horn H, Kriegel H-P, Schneider R "Efficient Spatial Query Processing in Geographic Database Systems" Institute for Computer Science, University of Munich, 1993.

[4] C.-Y. Chow, H. V. Leong, and A. T. S. Chan. "Group-based Cooperative Cache Management for Mobile Clients in a Mobile Environment".In Proceedings of the International Conference on Parallel Processing (ICPP), Montreal, Quebec, Canada, August 15-18, 2004

[5] C.-Y. Chow, H. V. Leong, and A. Chan. "Peer-to-Peer Cooperative Caching in Mobile Environmen". In Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, pages 528–533, 2004.

[6] M. Dahlin, R. Y. Wang, T. E. Anderson, and D. A. Patterson. "Cooperative caching: Using remote client memory to improve file system performance". In Proceedings of the 1st USENIX OSDI, pages 267–280, November 1994.

[7] D Greene "An Implementation and Performance Analysis of Spatial Data Access Method", Proc 5th International Conference on Data Engineering.606-615, 1989.

[8] A Guttman"R-trees a dynamic index structure for spatial searching", Proc ACM SIGMOD International Conference on Management of Data, 47-57, 1984.

[9] G. R. Hjaltason and H. Samet.'Distance Browsing in Spatial Databases'. ACM Trans. Database Syst., 24(2):265–318, 1999.

[10] H. Hu, J. Xu, W. S. Wong, B. Zheng, D. L. Lee, and W.-C. Lee."Proactive Caching for Spatial Queries in Mobile Environments". In Proceedings of the 21st International Conference on Data Engineering (ICDE), Tokyo, Japan, April 5-8, 2005.

[11] M. Kolahdouzan and C. Shahabi. "Voronoi-based k nearest neighbor search for spatial network databases". In Proceedings of the 30th International Conference on Very Large Databases (VLDB), pages 840–851, Toronto, Canada,August 31 - September 3, 2004.

[12] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. "Query Processing in Spatial Network Databases". In Proceedings of the International Conference on Very Large Databases, pages 790–801, 2003.

[13] N. Roussopoulos, S. Kelley, and F. Vincent. "Nearest Neighbor Queries". In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, pages 71–79, San Jose, CA, May 22-25, 1995.

[14] Z. Song and N. Roussopoulos. "k-Nearest Neighbor Search for Moving Query Point. In Proceedings of Advances in Spatial and Temporal Databases", 7th International Symposium (SSTD), pages 79–96, Redondo Beach, CA, July 12-15,2001.

[15] Wei-Shinn Ku, Roger Zimmermann, Chi-Ngai Wan and Haojun Wang "A P2P Simulation Model to Support Mobile, Scalable Nearest Neighbor Queries for Location-based Services" Computer Science Department, University of Southern California, Los Angeles,2006.

[16] B. Zheng, W.-C. Lee, and D. L. Lee. "On Semantic Caching and Query Scheduling for Mobile Nearest-Neighbor Search". Wireless Networks, 10(6):653–664, November 2004.

[17] Diya Thomas, " Location Dependent Query Processing in Mobile Environment ", International Conference on Advances in Computing and Communications, pp. 35-37, 2012.

[18] S. D Sharma,R.SKasana, " Mobile Database System: Role of Mobility on the Query Processing",Journal of Computer science and Information Security,pp. 211–216, 2010

[19] Mi Young Jang, Jae Woo Chang, " A New K-NN Query Processing Algorithm Enhancing Privacy Protection in Location-based Services" 11th IEEE International Conference on Computer and Information Technology pp.421-428, 2011.