Scheduling and Monitoring Spatial Queries Using Continuous KNN and Shortest Path

T.Venkat Narayana Rao* Professor HOD,C.S.E
S.Naresh Kumar** Assistant Professor,C.S.E
K.Madhuri*** Assistant Professor,C.S.E
Department of Computer Science and Engineering
Hyderabad Institute of Technology and Management [HITAM]
Hyderabad, A.P, India.

ABSTRACT

Information retrieval using wireless networks is now a promising practice in the era of mobile computing; because of the mobile networks has dynamic nature. They may not be able to handle a tremendous amount of traffic and service requests from the users due to asynchronous upstream and downstream capacity of the network which in turn may reduce the computational capability both at client and server. To enhance the computational capabilities of these mobile devices specifically when there is a request to the server for multiple spatial information having multiple entities has become a challenge to mobile computing researchers. In this paper we propose a framework that continuously broadcasts the information from server, by using a special indexing scheme for query processing that is not only efficient in processing of continuous special queries but it also reduces the energy consumption and enable the integration of low-cost GPS devices . In any portable unit we aimed to design shortest path for the client query to reach the destination using elliptical boundary and nearest neighbor methods.

Keywords- **Location-based services, Air Index, Broadcast Grid Index, Geographical Querying, Elliptical boundary.**

I. INTRODUCTION

Now a days the computational capabilities of the mobile devices in a wireless environment have been increased tremendously. On the other hand technology has been increased to provide the location based services to the mobile applicants. The location based services has grown to support a larger user community and to provide more intelligent services when client requests for any queries [1]. A continuous spatial query runs over long periods of time and requests constant reporting of its result as the data changes dynamically. Typically, the query type is a range or nearest neighbor (NN), and the assumed distance metric is the Euclidean one. In general, there are multiple queries being processed simultaneously. The query processes and the data objects move frequently and arbitrarily i.e., their velocity vectors and motion patterns are unknown. They issue location updates to a central server, which processes them and continuously reports the current query results. Consider, for example, that the queries correspond to vacant cabs, and the data objects are pedestrians that ask for a taxi.

As cabs and pedestrians move, each free taxi driver wishes to know his/her closest client. In this application the client can send multiple queries at a time to the server this is an instance of continuous NN monitoring. Spatial monitoring systems aim at minimizing the processing time at the server and/or the communication cost incurred by location updates. Due to the time-critical nature of the problem, the data are usually stored in main memory to allow fast processing [4]. In Existing System, server will continuously broadcast the spatial data to all the clients in wireless environment. At the client side client always send the queries to the server, in this process energy consumption of battery will be reduced while client continuously access the spatial data from the server. This causes the following limitations.

- While accessing the information from the server client needs to send spatial queries to server continuously. In this process load at the server side increases.
- Existing applications will not search depending on the client's location.
- Query processing is done based on the client's location.
- For getting the map from the sever, client continuously access the server and this causes power consumption high.
- No Accurate results will be obtained by previous existing applications
- Clients can query only on static data.

To avoid all the above constraints we proposed a system which will enhance the following capabilities:-

- Accurate results would be obtained.
- Addresses continuous queries.
- Considers even moving objects.
- Querying over static data (snapshot data).
- Querying over dynamic data (continuous queries).
- Uses grids for mapping the objects so that changes can be continuously updated.
- Query results can retrieved by using a key associated with the query issued.

II. BACKGROUND

2.1 Rich Internet Applications(RIA)

Historically, there have been major shifts in the software industry. We moved from mainframes with dumb terminals to client/server technology. Users gained convenience and productivity, and mainframe systems were patronizingly labeled as legacy. With the availability of the World Wide Web industry, visionaries turned the table. Vendors and corporate IT had been eager to get rid of the complexity of client/server. Version management and technologists were sold on multi-tier computing. This time client/server was called legacy. And to rub it in, good old desktop applications were labeled "fat". Excited with server multi-threading, messaging, persistence, and similar toys. To make us feel better, we proudly called the new breed of applications "thin client". To meet the new challenges, we are entering an era of rich Internet applications (RIA), which restores the power of desktop applications. RIAs run in a virtual machine (i.e., Adobe Flash Player or Java VM) and have the potential of becoming a full-featured desktop application soon [9]. As opposed to just simply displaying web pages delivered from some server machine, RIA really runs on the client. Many of the data manipulation tasks (sorting, grouping, and filtering) are done locally like in the old client/server days. Industry analysts predict that in three or four years most newly developed projects will include RIA technologies. RIA combines the benefits of using web as a low-cost deployment model with a rich user experience that is at least as good as today's desktop applications. Since RIAs don't require the entire page to refresh in order to update their data, the response time is much faster and the network load is much lower. Let's illustrate the difference between "legacy" Web and RIA with a shopping cart example. Non-RIA Web applications are page-based. Since HTTP is a stateless protocol, when the user moves from one page to another, a Web browser doesn't remember the user's actions on the previous page. As a common treatment of this amnesia, a user state is stored on the server side in the form of the HTTP session. Consider the case of an online shopping session. It can go as follows:

- The user initiates a search for an item on Web page #1.
- The server processes this request and returns page #2 that may or may not contain the required item.
- The user adds an item to a shopping cart that takes yet another trip to the server to create the shopping cart. And store it on the server side then the server responds with page #3 so the user can either continue shopping (repeating the first three steps) or proceed to the checkout – page #4. At the checkout the server retrieves selected items from the session object and sends page #5 to the user for shipping information. The data travels back to the server for storage, and the client gets back page #6 for billing information. After that page #7 will confirm the order and completion page. This simplest online purchase consisted of seven roundtrips to the server. In a striking difference to desktop applications. A few seconds per page refresh is considered fast for a typical web application. The commonly acceptable delay is up to eight seconds. Now assume that the network server is slow. Your potential buyer went elsewhere then RIA eliminates the roundtrips and substantially improves the system performance by doing a lot more of the processing on the client than a thin client web application. To put it simply, RIA isn't a set of pages controlled by the server; they are actual applications running on the client's computer and communicating with servers primarily to process and exchange data.

2.2 Wireless Broadcast

The transmission schedule in a wireless broadcast system consists of a series of broadcast cycles. Within each cycle the data are organized into a number of indexes and data buckets. A bucket (constant size) corresponds to the smallest logical unit of information. Similar to the page concept in conventional storage systems. A single bucket may be carried into multiple network packets (i.e., the basic unit of information that is transmitted over the air). However, they are typically assumed to be of the same size (i.e., one bucket equals one packet) [2].

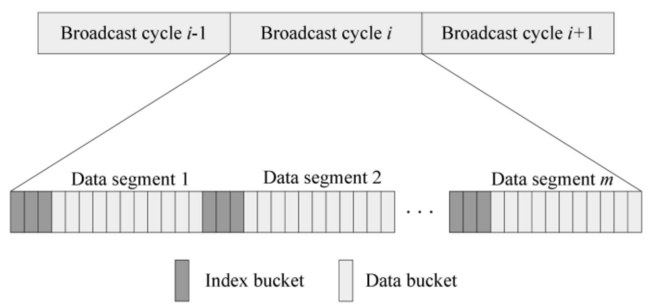

Figure 1: Cycles partitioned into segments

2.3 FLEX

Flex through which we can create an Action Script application for the ubiquitous Flash Player, a high-performance multimedia virtual machine that runs byte code files in the **SWF** format (pronounced swif). The player's JIT compiler converts the SWF byte code to native machine code for fast performance. The later facility is specific to Flex 2 available since 2006. Although early versions of Flex were out in 2004, they didn't support just in time compilation. Flex 2 applications run cross platform in a ubiquitous Flash Player 9 that's a lightweight virtual machine. The platform includes:

- An XML-based language called MXML that supports the declarative programming of GUI components targeting designers.
- The standard object-oriented programming language, Action Script 3.0, based on the latest ECMA Script specification.
- Server-side integration via Flex Data Services (FDS) giving client applications transparent access to the world of J2EE.
- Charting components, access to multimedia controls, etc.
- An Eclipse-based full-featured IDE with automated deployment, debugging, and tracing Facilities.

The Flex 2 platform is easily extendable and integrates well with server side Java, ColdFusion, PHP, Ruby, ASP. The upcoming release of Adobe Apollo will allow the creation of a desktop application based on Flash Player, Flex, PDF, and HTML. The SWF file format is open, and there are third party open source products that offer tools for creating RIAs delivered by Flash Player like Open Laszlo from Laszlo Systems. As opposed to the last version, Flex 2 offers a way to create RIAs without incurring hefty licensing.

2.4 Open Plug ELIPS Studio

ELIPS Studio is leading the next generation of mobile application development environments. Now software developers and creative designers can quickly develop connected or non-connected applications. And deploy them on multiple mobile platforms. ELIPS Studio is equally valuable for experienced mobile application developers who want extra productivity, and for Adobe Flash / Flex and web developers who want to reuse their RIA / RDA assets and create rich mobile applications. Until now, application developers needed in depth expertise in all the mobile operating systems if they wanted to address a global market of over 4 billion phone users. And not just every operating system. They also had to understand the differences between the individual models from each manufacturer. ELIPS Studio cuts through the complexity of the fragmented mobile market by providing a single development toolkit for the world's most popular handsets. It includes a sophisticated simulator, and the code it generates is compact and rapid enough to run even on low-specification, mass-market handsets.

III. IMPLEMENATAION

3.1 Client Server networking using broadcast grid index method

Client/server architecture can be considered as a network environment that exchanges information between a server machine and a client machine where client has some queries that has to be shared by server. So in this system the client can send multiple queries at a time to the server. Server will continuously receive the spatial queries from the client side. Because there are more than one client can access the same server at time. In this situation server may react slowly because it has to process lot of queries and filtering. Mobile networks are dynamic nature they may not able to handle a tremendous amount of traffic and service requests from the users due to asynchronous upstream and downstream capacity of the network which in turn may reduces the computational capability both at client and server.
To avoid this we used BGI (Broadcast Grid Index Method). BGI is suitable for both static and dynamic data [1] [2].

1) Whenever the client issues the query, it broadcasts the query to the server and goes to sleep mode for the next index.
2) Then index is been traversed and determined by the client about the query to be satisfied by the data objects at the server.
3) The client wakes up receives the content.
4) When client retrieved the content (map) from the server an index is been generated to that place.
5) If the client wants to access some other interest on same place again by using the index we can easily store the map without communicating with the server again and again.

3.2 Working with Static Objects

Working with the static objects is nothing but we can access the application by sitting in one place any location. For example if client want to access application from Hyderabad and he needs hospitals in Mumbai he can access the application very easily.

3.3 Working with dynamic Objects

Working with dynamic objects client will move one place to another place like roads, rivers. In travelling client can access the application and from that place As shown in figure 2 the data objects are moving objects like taxis that issue location updates to a central server using unicast uplink messages. The server processes the location updates and continuously broadcasts the object information along with an up-to-date index using a wireless network. The interested clients (e.g., mobile devices) listen to the broadcast channel and process their queries locally [1].

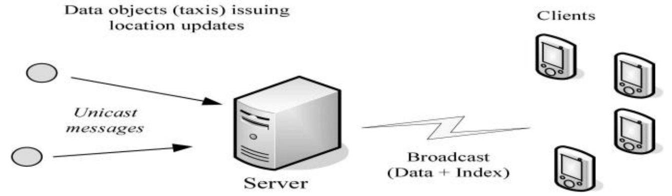

Figure 2: Broadcast Grid Index Methods

3.4 Wireless Broadcasting

The wireless Internet will allow moving clients to access their data and keeping connections active while moving from one access point to another. Making this moving seamlessly and efficient at high hand over rates while conserving quality of service profiles is a challenge. To make intelligent mobile aware applications and to achieve efficient routing management. It may be important to anticipate the change of location of users in mobile networks. In the current existing system, if you want to know about any unknown thing (places, routes, cities, rivers, hotel, theatres, etc..), simply we go for phone who knows that answer, or any search engine like Google or yahoo etc… which demands the current location of the client. The beauty of this application is that whenever a client accesses the application, the application automatically checks signal for providers. The application will catch the place of the signal and send to server. So then application will retrieve the map from the server based on latitude and longitude and also it is been used for business search i.e. which is able to provide name, address and contact number of required public or private sector.

3.5 Air Indexing

At the client side, every time it receives the spatial data and continuously accessing server for new data. So that we can have problem with battery consumption in travelling. (Old GPS devices and Mobile also). To avoid this we used Air Indexes technique [1] [4].The main motivation behind the air indexes is to minimize the power with low latency at mobile devices.

3.6 Spatial Query Processing

A spatial query is a technique using which we can provide a user friendly environment to the client for providing multiple queries at a time to the server. It is able to provide a snapshot query on static data is indexed spatial method. The main idea behind this method is to provide additional information like more nearest neighbors, mode, and range. The moving object (client) need to issue other query only after current query expires, i.e. it concentrates on single query processing and it is unable to maintain the object results. In order to broadcast the multiple queries at a time to the server we used Continuous k nearest neighbor (CKNN) [1]. Algorithm to provide accurate results, for a range queries we used R-tree [2] as the underlying index.

3.7 Continuous KNN Queries

More people focus on continuous K-Nearest Neighbor (CKNN) query processing [2]. Over moving Objects in Road Networks. A CKNN query is to find among all moving objects the K-nearest neighbors (KNNs) of a moving query object during a period of time. The main issue with existing methods is that moving objects change their locations frequently over time and if their location updates cannot be processed in time, the system runs the risk of retrieving the incorrect results of KNN. In this paper, an effective method is proposed to deal with continuous K-Nearest Neighbor query processing [6]. By considering whether a moving object is moving farther away from or getting closer to a query point, the object which is definitely not in the KNN result set is effectively excluded. Thus, we can reduce the communication cost [10]. Meanwhile we can also simplify the network distance computation between moving objects and query Comprehensive experiments are conducted and the results verify the effectiveness of the proposed algorithms.

3.7 Getting the Map

Step: 1 the interests are populated using the combo box such as theaters, hostilities, restaurants. Client need to select the any one from combo box application [Figure 1and 5].
Step: 2: After identifying the client interest from combo box, client needs to given range query to the application in miles.
Step: 3 after entering the details of all mode of interest and Range dosearch method will take inputs from the client and also catch the location of the client send all those details to server using Google API.

Step: 4 Server will process the request based on the client location by using map.getLatLngBounds () method and spatial queries it will broadcast the image [Figure 6].

3.8 Getting the Address

Step: 1 after getting the map application should mark the client interest by using createMarker() method. Marker method will take client latitude and longitude and color to filled in marker and so on values and display the markers depends on location and interest given by client [Figure 6].
marker:Marker = createMarker(latlng, result.titleNoFormatting, html, color, (j+1));

Step: 2 After placing the marker when user clicks on any marker, application will call the address using address.text=arrAddress[i].toString(); and it will be stored in data list for name, address,ciy, phone numbers using these methods [Figure 3-7].
dataList.addChild(name1);
dataList.addChild(address);
dataList.addChild(city);
dataList.addChild(numbers);

3.9 Search for different locations

Client can search from different location for static objects by search module.
Step 1: user needs to give the name of city e.g. Mumbai, India to search particular one to search in Mumbai.

Step 2: the application will take this text and it will send to server Application will call onGeoCode (result: GeocodingEvent): this method to convert this text to latitude and longitude values and will send to server.

Step 3: Server will call the onChangeArea() method and by using below code will get the new location map.
var geocode:ClientGeocoder=new ClientGeocoder();
geocode.addEventListener(GeocodingEvent.GEOCODING_SUCCESS,onGeoCode); geocode.geocode(caarr6);

Step 4: For getting the map we will follow the same as in Figure 1.

3.10 Shortest Path

The location based services involves mainly finding transportation networks which in turn focuses on shortest path to reach nearest neighbor. Motivated by scalability challenges faced in the mobile network industry, we propose adopting the wireless broadcast model for such location-dependent applications. In this model, continuously we are transmitted data on the air, while clients listen to the broadcast and process their queries locally. Although spatial problems have been considered in this environment, there exists no study on shortest path queries in road networks [3].
1. We adapt traditional shortest path algorithms (Dijstra's algorithm) to the broadcast model, and identify their weaknesses in this setting [3].
2. We present two novel methods, namely Elliptic Boundary (EB) and Nearest Region (NR), which exploit the broadcast environment's characteristics and take into account the technical limitations of mobile devices.

3.11 Getting the Directions

Ater getting the list of addresses, user need to select any address then application will call the getDirectionsDataItem (eventItem: ItemTouchTapEvent) method to get the directions to that particular address[8] [7]. The mobile client can change the mode and range to filter the radius of the mobile. The range and mode given by the client is queried and searched in the repository. The response for the specific range and mode is processed and sent back to the Mobile Client based on the radius as shown in Figure 8.

IV. OUTPUT OF THE IMPLEMETATION

Figure 3: the emulator on which the application is executed with welcomes Screen of the Application.

Figure 4: process the client current location that is latitude and longitude and show the exact location through the Google Map Server.

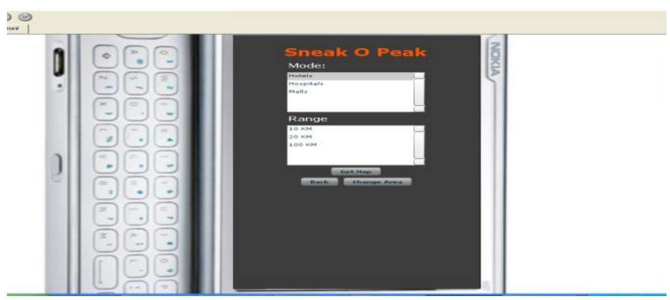

Figure 5: the client to change the Mode and Range in order to get the map based on the filter criteria selected by the client.

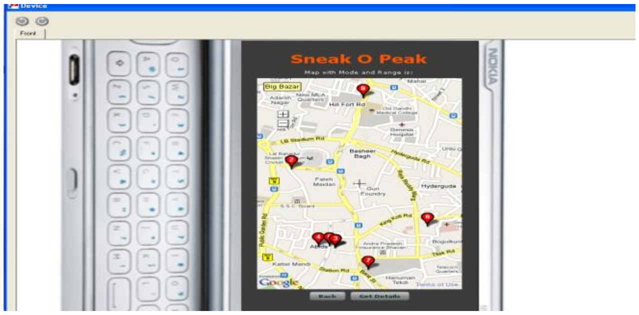

Figure 6: After filtering the range and Mode the application will provide a map with Markers for those filter values from the server. That is for Example the above screen shows the markers for Malls within 10KMs.

Figure : 7: the option get Details will allow the Mobile client to see the all the markers details such as the name, address and Phone Number

Figure 8: Output shown in other Mobile Devices

V. CONCLUSION

In this paper, we are studied regarding maps for mobile clients to reach their destination for which we are providing location based updates in a user friendly environment. A client can send multiple queries at a time to the server and process their request accordingly and server sends the response to the clients. Since more than one client can access the server at a time which may reduce the battery consumption of mobile devices. To avoid this we have used air indexing technology which makes the client to go into sleep mode after sending the query whenever response arrives, the client wakes up and receive the required result and in addition to above we have also discussed shortest path to reach nearest neighbor.

VI. FUTURE ENHANCEMENTS

Previous work on location-dependent spatial query processing for wireless broadcast systems has only considered snapshot queries over static data and moving objects. It is basically one server with more than one client object in moving position. At the same time one client can send location and queries to server, server will process the query and accept other queries from another client object and then synchronize with two clients moving objects. And also if we can make use of android then probably the mobile clients can search for even street views for their queries and also it will increase the marketing values of mobile devices.

VII. REFERENCES

[1] Kyriakos Mouratidis, Spiridon Bakiras, *Member, IEEE*, and Dimitris Papadias "Continuous Monitoring of Spatial Queries inWireless Broadcast Environments".2009

[2] Kyriakos Mouratidis School of Information Systems Singapore Management University "Continuous Monitoring of Spatial Queries"

[3] Georgios Kellaris Kyriakos Mouratidis" Shortest Path Computation on Air Indexes"

[4] Y. Cai, K.A. Hua, and G. Cao, "Processing Range-Monitoring Queries on Heterogeneous Mobile Objects," Proc. IEEE Int'l Conf. Mobile Data Management (MDM '04), 2004.

[5] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," Proc. ACM MobiCom, 2001.

[6] M.-S. Chen, P.S. Yu, and K.-L. Wu, "Indexed Sequential Data Broadcasting in Wireless Mobile Computing," Proc. Int'l Conf. Distributed Computing Systems (ICDCS '97), 1997.

[7] B. Gedik and L. Liu, "MobiEyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System," Proc. Int'l Conf. Extending Database Technology (EDBT '04),2004.

[8] B. Gedik, A. Singh, and L. Liu, "Energy Efficient Exact kNN Search in Wireless Broadcast Environments," Proc. Ann. ACM Int'l Workshop Geographic Information Systems (GIS '04), 2004.

[9] S.E. Hambrusch, C.-M. Liu, W.G. Aref, and S. Prabhakar, "Query Processing in Broadcasted Spatial Index Trees," Proc. Int'l Symp. Advances in Spatial and Temporal Databases (SSTD '01), 2001.

[10] A. Henrich, "A Distance Scan Algorithm for Spatial Access Structures," Proc. Second ACM Workshop Geographic Information Systems (GIS '94), 1994.

VIII. AUTHORS

Professor T. Venkat Narayana Rao, received B.E in Computer Technology and Engineering from Nagpur University, Nagpur, India, M.B.A (Systems), holds a M.Tech in Computer Science from Jawaharlal Nehru Technological University, Hyderabad, A.P., India and a Research Scholar in JNTU. He has 20 years of vast experience in Computer Science and Engineering areas pertaining to academics and industry related I.T issues. He is presently Professor and Head, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], JNTUH, Hydreabad, A.P, INDIA. He is nominated as an Editor and Reviewer to 25 International journals relating to Computer Science and Information Technology. He is currently working on research areas which include Digital Image Processing, Digital Watermarking, Data Mining, Network Security and other Emerging areas of Information Technology.

Mr. S.Naresh Kumar, Graduated in Computer Science and Engineering and M.Tech from Jawaharlal Nehru Technological

University, Hyderabad, A.P., India. He is presently working as Assistant Professor in Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], Hyderabad, A.P, India.

Mrs K. Madhuri Reddy Graduated in Computer Science and Engineering and M.Tech (C.S.E) from Jawaharlal Nehru

Technological University, Hyderabad, A.P., and India. He is presently working as Assistant Professor in Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management [HITAM], Hyderabad, A.P, India. Her research focuses on Mobile Computing, Data Stream Processing, and Spatiotemporal Databases.