Constant Supervision of Location Based Spatial Queries in Wireless Broadcast MediumUsing Peer-To-Peer Sharing

SIRISHA MADHURI T.
M. Tech Student,
Department of Computer Science & Engineering
Sri Venkateswara University College of Engineering
Tirupati, India

DR. A. RAMA MOHAN REDDY
Professor,
Department of Computer Science & Engineering
Sri Venkateswara University College of Engineering
Tirupati, India

Abstract:
Location based spatial queries (LBSQ's) are the queries that are totally dependent on the location of the probers. Wireless data broadcast is an assuring approach for information airing that influences the computational capacities of the Personal Digital Assistants (PDAs) that tend to elevate the scalability of the system. In this paper we present a contemporary query processing like spatial query procedure that while retaining high scalability and a high definiteness manages to lessen the latency regularly. Our mode is based on peer-to-peer sharing which empower us to deal with queries without delay at a PDA prober by using query outcomes cached in its adjoining mobile peers.

Keywords: *Air Indexes, Broadcast Channel, Location based Spatial Queries, Peer-To-Peer sharing, Probers, Spatial Databases*

---

1. Introduction

Spatial query processing is becoming an integral part of many new applications. Recently, there has been a growing Interest in the use of location-based spatial queries [2] (LBSQs), which refer to a set of spatial queries that retrieve information based on mobile users' current locations. MOBILE devices with computational, storage, and wireless communication Capabilities (such as PDAs) are becoming increasingly popular. At the same time, the technology behind positioning systems is constantly enabling the integration of low-cost GPS devices in any portable unit. Consider, for instance, a user (mobile client) in an unfamiliar city, who would like to know the 10 closest restaurants. This is an instance of a k nearest neighbor (kNN) query[3], where the query point is the current location[2] of the client and the set of data objects contains the city restaurants. Alternatively, the user may ask for all restaurants located within a certain distance, i.e., within 200 meters. This is an instance of a range query.

Continuous supervision of multiple queries over arbitrarily moving objects [5]. In this setting, there is a central server that monitors the locations of both objects and queries. The task of the server is to report and continuously update the query results as the clients and the objects move. The processing load at the server side increases with the number of queries. In applications involving numerous clients, the server may be conquered by their queries or take excessively long time to answer them. To avoid this problem we have a wireless broadcast which is an assuring approach for information airing that influences the computational capacities of the Personal Digital Assistants (PDAs) that tend to elevate the scalability of the system. The data objects are continuously broadcast by the server, interleaved with some indexing information. The clients utilize the broadcast index, called air index, to tune in the channel only during the transmission of the relevant data and process their queries locally. Thus, the server load is independent of number of clients.

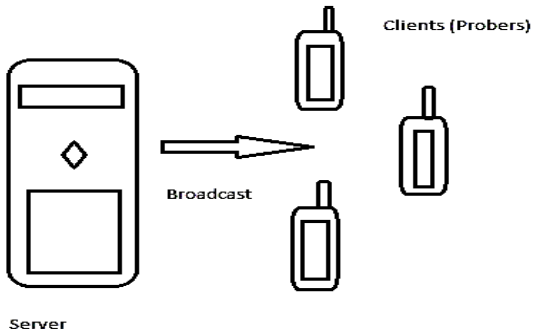

Fig. 1. Example of Query Processing

In the Wireless Broadcasting channel[1] we have the interleaving model which contains the number of broadcasting channels with data segments[1] within them. Each data segment again has the indexing information which contains the data of the related LBSQ. For applications where the objects are moving, the server broadcasts a dirty grid in the beginning of each cycle .The dirty grid [1] indicates the regions of the data space that have received updates since the last cycle.

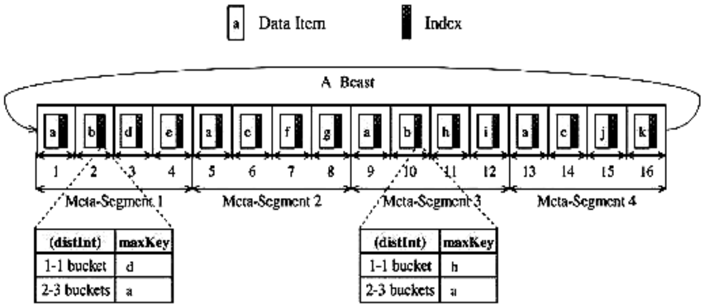

Fig. 2. Wireless Broadcast Channel
[Courtesy: www.sciencedirect.com]

Novel query processing techniques must be devised to handle these new challenges.

Mobile Query Semantics: In a mobile environment, a typical LBSQ is of the following form: "find the top-three nearest hospitals." The result of the query depends on the location of its requester. Caching and sharing of query results must take into consideration the location of the query issuer.

High Workload: The database resides in a centralized server, which typically serves a large mobile user community through wireless communication. Consequently, bandwidth constraints and scalability become the most important design concerns of LBSQ algorithms.

Query Promptness And Accuracy: Due to users' mobility, answers to an LBSQ will lose their relevancy if there is a long delay in query processing or in communication. For example, answers to the query "find the top-three nearest hospitals" received after five minutes of high-speed driving will become meaningless.

The wireless environment and the communication constraints play an important role in determining the strategy for processing LBSQs. In the simplest approach, a user establishes a point-to-point communication with the server so that her queries can be answered on demand. However, this approach suffers from several drawbacks. First, it may not scale to very large systems. Second, to communicate with the server, a client must most likely use a fee-based cellular type network to achieve a reasonable operating range. And third, users must reveal their current location and send it to the server, which may be undesirable for privacy reasons. A more advanced solution is the wireless broadcast model. It can support an almost unlimited number of mobile hosts (MH) over a large geographical area with a single transmitter. With the broadcast model, mobile hosts do not submit queries instead they tune in to the broadcast channel for information which they desire. Hence, the user's location is not revealed. One of the limitations of the broadcast model is that it restricts data access to be sequential. Queries can only be fulfilled after all the required on-air data arrives. This is why in some cases; a five-minute delay to the query "find the top-three nearest hospitals" would not be unusual. Alleviating this limitation, we propose a scalable and low latency approach for processing location-based spatial queries in broadcast environments. Our approach leverages ad-hoc networks to share information among mobile clients in a peer-to-peer (P2P) manner. The rationale for our approach is based on the following observations.

- As mentioned previously, when a MH launches a nearest neighbour query, in many situations, she would prefer an approximate result that arrives with a short response time rather than an accurate result with a long latency.
- The results of spatial queries often exhibit spatial locality. For example, if two mobile hosts are close to each other, the result sets of their spatial queries may overlap significantly. Query results of a mobile peer are valuable for two reasons: *i*) they can be used to answer queries of the current mobile host directly; and *ii*) they can be used to dramatically reduce the latency for the current mobile host to retrieve on-air information.
- P2P approaches can be valuable for applications where the response time is an important concern. Through mobile cooperative caching [2] of the result sets, query results can be efficiently shared among mobile clients.

In this paper, we concentrate on two common types of spatial searches, namely, *k* nearest neighbor [KNN] queries and window queries. The contributions of our study are as follows.

a) We identify certain characteristics of LBSQs that enable the development of effective sharing methods in broadcast environments.

b) We utilize a P2P based sharing method to improve the current approaches in answering on-air *k* nearest neighbour queries and window queries.

c) Through extensive simulation experiments, we evaluate the benefits of our approach under different parameter sets.

2. Related Work

2.1 Wireless Data Broadcast

We can distinguish two approaches for mobile data access: the *on-demand access model* and the *wireless broadcast model*. For the on-demand access model, point-to-point connections are established between the server and the mobile clients, and the server processes queries which the clients submit on demand. For the wireless broadcast model, the server repeatedly broadcasts all the information in wireless channels and the clients are responsible for filtering the information. The advantage of the broadcast model is that it is a scalable approach. However, the broadcast model has a large latency, as clients have to wait for the information they need in a broadcasting cycle. If a client misses the packets which it needs, it has to wait for the next broadcast cycle. Nearly all the existing spatial access methods are for databases with random access disks. These existing techniques cannot be used in a wireless broadcast environment, where only sequential data access is supported. Zheng et al. proposed to index the spatial data on the server by a space-filling curve. The Hilbert curve is chosen for this purpose because of its superior locality. The index values of the data packets represent the order in which these data packets are broadcast.

2.2 Spatial Queries and Cooperative Caching

We focus on two common types of spatial queries: *k* nearest neighbour queries and window queries. For nearest neighbour queries, the use of the R-tree algorithm [4] and its derivatives, and the branch-and-bound algorithms that search an R-tree in either a depth-first [7] or best-first manner [5], have been widely adopted. For window queries that find objects within a specified area, the R-tree families [8, 1] provide efficient access to disk-based databases. Basically, an R-tree group's object close to each other into a minimum bounding rectangle (MBR) and a window query only visits the MBRs that overlap with the query window. Caching is a key technique to improve data retrieval performance in widely distributed environments. With the increasing deployment of new P2P wireless communication technologies (e.g., IEEE 802.11b/g), *peer-to-peer cooperative caching* becomes an effective sharing alternative [9]. With this technique, mobile hosts communicate with neighbouring peers in an *ad-hoc* manner for information sharing, instead of relying solely on the communication with remote information sources. Peer-to-peer cooperative caching can bring about several distinctive benefits to a mobile system: improved access latency, reduced server workload, and alleviated point-to-point channel congestion.

3. Wireless Data Broadcasting and Air Indexes

We can distinguish two approaches for mobile data access the on-demand access model and the wireless broadcast model [1]. For the on-demand access model, point-to-point connections are established between the server and the mobile clients, and the server processes queries which the clients submit on demand. For the wireless broadcast model [1], the server repeatedly broadcasts all the information in wireless channels and the clients are responsible for filtering the information. The advantage of the broadcast model is that it is a scalable approach. The most common data organization method is the interleaving scheme, as shown in Fig. 2. The data objects are divided into some distinct segments and each data segment in the transmission schedule is preceded by a complete version of the index. In this way, the access latency for a client is minimized, since it may access the index immediately after the completion of the current data segment.

The main motivation behind air indexes is to minimize the power consumption at the mobile client. Although in a broadcast environment [2], the uplink transmissions are avoided, receiving all the downlink packets from the server is not energy efficient. To measure the efficiency of an indexing method, two performance metrics have been considered.

1. Time to tune
2. Access Latency

Time to tune is the total time that the client stays in the receiving mode to process the query.
Access latency is the total time elapsed from the moment the query is issued until the moment that all the corresponding objects are retrieved.

3.1. Location Based Spatial Query Processing Using Air Indexes

Here we have the exploration of possibility of broadcasting spatial data [2] together with a data partitioning index. There is another class of queries called NN Queries namely (LNN) queries, in the context of wireless broadcast systems [3]. The air indexing structure can be considered using a Hilbert curve.

Hilbert Curve : A **Hilbert curve** (also known as a **Hilbert space-filling curve**) is a continuous fractal space-filling curve first described by the German mathematician David Hilbert in 1891, as a variant of the space-filling curves discovered by Giuseppe Peano in 1890.Because it is space-filling, its Hausdorff dimension is 2.

H_n is the nth approximation to the limiting curve. The Euclidean length of H_n is $2^n - \frac{1}{2^n}$, i.e., it grows exponentially with n, while at the same time always being bounded by a square with a finite area. Here we consider the Broadcast Grid Index which is nothing but a broadcast channel.

BGI indexes the data objects with a regular grid [1], i.e., a partitioning of the data space into square cells of equal size with side-length _ (a system parameter). Each cell stores the object coordinates falling inside and maintains their total number.

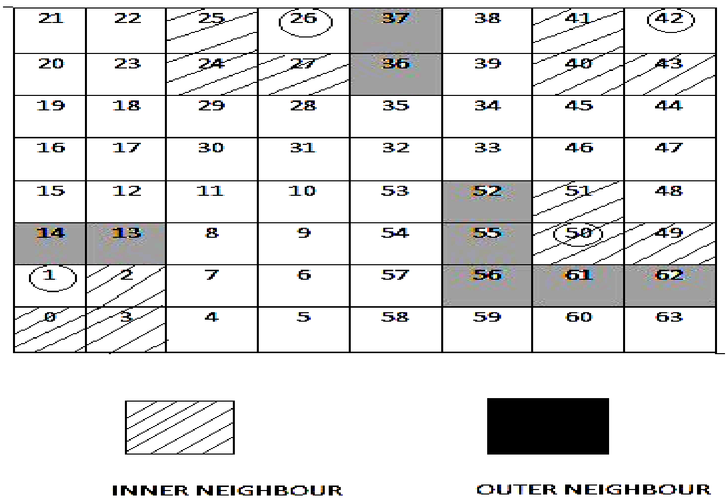

Fig. 3. Forming the Index Segment through Hilbert Curve

4. Query Processing

We hear use the KNN Query computation which runs completely at the client side. For this here an index segment[6] is taken like the following.

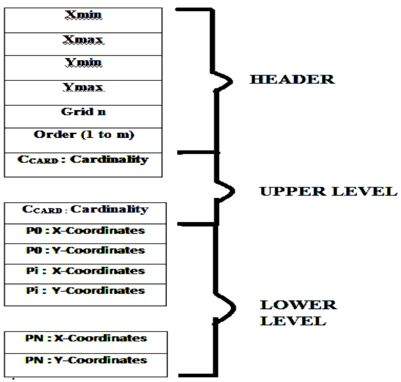

Fig. 4. An Index Segment [Courtesy: Continuous Monitoring of Spatial Queries in Wireless Broadcast Environments]

The KNN computation algorithm is shown below in fig 5.

```
KNN Computation

//Client at q goes online and listen to the first index segment
        //Step 1:The upper level is broadcast
1.best NN = φ; dmax=∞
2.for each bucket B
3.if mindist(B)<dmax //Prune upper level buckets
4.      for each cell in B
5.              for iter = 1 to n
6.              if maxdist(c) < dmax
7.              delete the kth entry of best NN
8.              insert < c, maxdist(c) > best NN: Update dmax
        //Step 2: The lower level is broadcast
9.  for each cell c with mindist(c) < dmax
10.delete all entries of c form best NN
11.for each object p in c
12.if dist(p) ≤ dmax
13.     delete the kth entry of best NN
14.     insert < p, dist(p) > into best NN: Update dmax
15.return best NN
```

Fig. 5. KNN Computation

Lines 1-8 implement step one, while lines 9-15 implement step two. The first step uses the best NN list to store the cells with the lowest maxdist values. In particular, each cell c in a considered bucket B generates up to c: card virtual entries <c; maxdist(c)> into the best NN list, where n is the cardinality of c.

In the second step, when a cell c is considered, we delete from best NN all its virtual entries (<c; max dist(c)>) and insert its actual objects p (if dist (p) _ d_{max}). Maintaining the same best NN list in both steps enhances the pruning power of the method; the max dist of a cell not transmitted so far is used to prune cells in line 9, even though its exact contents are not known.

5. Peer-To-Peer Sharing

A **peer-to-peer** (abbreviated to **P2P**) computer network is one in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to various resources. P2P networks can be set up within the home, a business, or over the Internet. Each network type requires all computers in the network to use the same or a compatible program to connect to each other and access files and other resources found on the other computer. P2P networks can be used for sharing content.

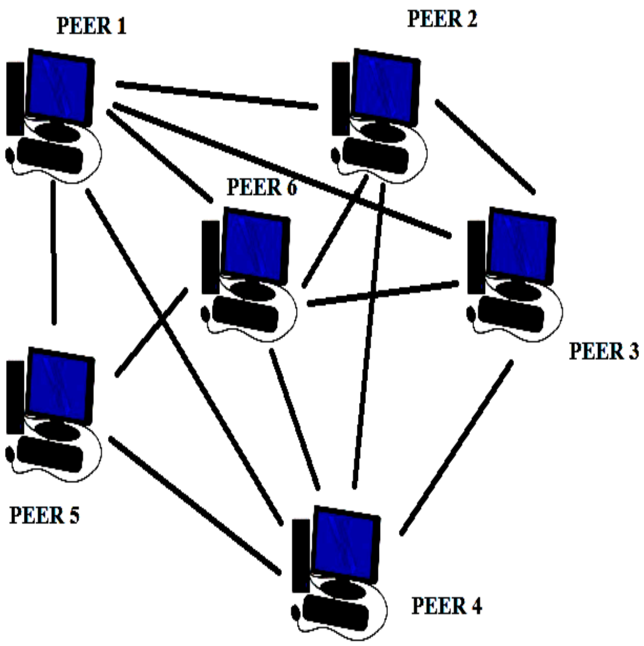

Fig. 6. Peer-To-Peer sharing network [Courtesy: www.sciencedirect.com]

Here the tuned users will share the data content with the adjoining peers who are related to a same location based spatial queries. When the users tune into the other pee's channel he can get the required information.The main purpose of our peer-to-peer design is to decrease the access latency, as queries are answered directly by peers. Consequently, the focus of our simulation is to quantify what percentage of the client spatial query requests can be fulfilled by peers. We acquired our simulation parameters from real world data sets, for example Hospitals and Theatres datasets in Tamil Nadu state. We named the two parameter sets the *Hospital and Theatre parameter* set. For comparison purposes we blended the two real parameter sets to generate a third, synthetic data set.

6. Conclusion

This paper presented an assuring approach for decreasing the spatial query access latency by leveraging results from nearby peers in wireless broadcast environments. Our scheme allows a mobile client to locally verify whether candidate objects received from peers are indeed part of its own. A central server transmits the data along with some indexing information. The clients process their queries locally, by accessing the broadcast channel. In this setting, our target is to reduce the power consumption and access latency at the client side. Significantly by virtue of its peer-to-peer architecture, the method exhibits great scalability. The higher the mobile peer density, the more queries can be answered by peers. Therefore, the query access latency can be gradually decreased with the increase of clients, which results in the increase of scalability of the server system.

Acknowledgement

We would like to express my heart-felt gratitude and my sincere thanks to my guide Dr. A. Rama Mohan Reddy, Professor, Head of Computer Science & Engineering, SV University College of

Engineering for his great efforts of supervising and leading me to accomplish this fine work. I also want to thank and be grateful to the authors **G. SWETHA** and **G. GOPI KRISHNA** who have made my research work a fully fledged one by lending their research paper named **CONTINUOUS MONITORING OF SPATIAL QUERIES IN WIRELESS BROADCAST ENVIRONMENTS.**

References

1. Acharya, S. R. Alonso, M.J. Franklin, and S.B. Zdonik, (1995).“Broadcast Disks: Data Management for A symmetric Communication Environments,” Proc. ACM SIGMOD.
2. Cai, Y. K. A. Hua, and G. Cao “Processing Range Monitoring Queries on Heterogeneous Mobile Objects.
3. Chen, M.S. P.S. Yu, and K.-L. Wu, (1997).“Indexed Sequential Data Broadcasting in Wireless Mobile Proc. Int’l Conf. Distributed Computing Systems computing (ICDCS ’97)
4. Gedik, B. and L. Liu (2004). “Mobi Eyes: Distributed Processing of Continuously Moving Queries on Moving Objects in a Mobile System,” Proc. Int’l Conf. Extending Database Technology (EDBT ’04).
5. Guttman, A. R-Trees: (1984). A Dynamic Index Structure for Spatial Searching. In SIGMOD Conference, pages 47–57, Boston, Massachusetts, June 18-21.
6. Hjaltason, G. R. and H. Samet. Distance Browsing in Spatial Databases.
7. Imielinski, T. S. Viswanathan, and B. R. Badrinath. (1997). Data on Air: Organization and Access. IEEE Trans. Knowl. Data Eng., 9(3):353–372.
8. Roussopoulos, N. S. Kelley, and F. Vincent. (1995). Nearest Neighbour Queries. In SIGMOD Conference, pages 71–79.
9. Sellis, T. K. N. Roussopoulos, and C. Faloutsos. (1987). The R+-Tree: A Dynamic Index for Multi-Dimensional Objects. In VLDB, pages 507–518.
10. Yin, L. and G. Cao.(2006). Supporting cooperative caching in ad hoc networks. IEEE Trans. Mob. Comput., 5(1):77–89.
11. Zhang, J. M. Zhu, D. Papadias, Y. Tao, and D. L. Lee. (2003).Location-based Spatial Queries. In SIGMOD Conference, pages 443–454.
12. Zheng, B. W.-C. Lee and D. L. Lee. (2004). Spatial Queries in Wireless Broadcast Systems. Wireless Networks, 10(6):723–736.