Scalable Spatial Query Processing for Location-aware Mobile Services

KwangJin Park[1], MoonBae Song[1], Ki-Sik Kong[1], Chong-Sun Hwang[1], Kwang-Sik Chung[2], and SoonYoung Jung[3]

[1] Dept. of Computer Science and Engineering, Korea University
5-1, Anam-dong, Seongbuk-Ku, Seoul 136-701, Korea
{kjpark, mbsong, kskong, hwang}@disys.korea.ac.kr
[2] Dept. of Computer Science and Engineering, Korea National Open University
kchung0825@knou.ac.kr
[3] Dept. of Computer Science Education, Korea University
jsy@comedu.korea.ac.kr

Abstract. Location-aware mobile services(LAMSs) are characterized by a large number of objects and a large number of queries. Moreover, with a large candidate data set, answering LAMSs via scanning through the whole data set becomes extremely expensive. In broadcast-based services, any number of clients can monitor the broadcast channel and retrieve the data as it arrives through the broadcast channel. Thus, a wireless broadcast system capable of answering LAMSs queries is considered a promising solution because it can serve a virtually unlimited number of users within its coverage. Furthermore, if the data is properly organized to cater to the needs of the clients, such a scheme makes effective use of the low wireless bandwidth, and is ideal for achieving maximal scalability. In this paper, we address the issues of supporting spatial queries of location-aware data via wireless data broadcast. A linear data broadcast based on their location is proposed to answer spatial queries on air. Comprehensive experimentation shows that the proposed scheme is highly scalable and is more efficient in terms of both tuning time and access latency in comparison to other techniques.

1 Introduction

Data dissemination is the delivery of data from the server to a larger set of clients. This is characterized by an inherent asymmetry in the communications. Any number of clients can monitor the broadcast channel and retrieve the data as it arrives on the broadcast channel. If the data is properly organized to cater to the needs of the clients, such a scheme makes effective use of the low wireless bandwidth and is ideal for achieving maximal scalability. Two key requirements for data access in wireless environments are **the conservation of power** and **the minimization of the client waiting time**.

With a large candidates data set, answering LAMSs via scanning through the whole data set becomes extremely expensive. Thus, index structures and related search algorithms have been proposed to provide efficient processing of LAMSs queries. Air indexing is one of techniques used to address this issue, and which operates by interleaving indexing information among the broadcast data items. By first accessing the broadcast index, the mobile client is able to predict the arrival time of the desired data. Thus, it can stay in the power save mode most of time, and tune into the broadcast channel only when the requested data arrives [2][3]. Air indexing techniques can be evaluated in terms of the following two factors: First, *Access Time(AT)*: The average time elapsed from the moment a client issues a query to the moment when the required data item is received by the client. Second, *Tuning Time(TT)*: The amount of time spent by a client listening to the channel. Then, the *Access Time* consists of two separate components, namely: *Probe Wait*: The average duration for getting to the next index segment. *Bcast Wait*: The average duration from the moment the index segment is encountered to the moment when the required data item is downloaded.

There are several indexing techniques which have been developed, such as the distributed indexing approach [3], the signature approach [4], and the hybrid approach [5]. In a recent paper [8], we proposed the concept of data sorting for broadcasting called BBS(Broadcast based Location Dependent Data Delivery Scheme). A preliminary simulation-based results showed that BBS is significantly reduce the AT. In this paper, we attempt to reduce both the TT and AT in the wireless mobile computing environment. After pointing out the limitations of the existing indexing schemes, we present various schemes that can overcome these problems. To the best of our knowledge, this is the first work in which NN query processing without on index segment is proposed and this technique provides the best access time. In this paper, we assume a geometric location model, i.e., a location is specified as a two-dimensional coordinate and the broadcasted data objects are static, such as restaurants, hospitals, and hotels. The mobile clients can identify their locations using systems such as the Global Positioning System(GPS).

The remainder of the paper is organized as follows: Section 2 provides background information. Section 3 describes the proposed algorithms. A performance evaluation is presented in section 4. Finally, section 5 concludes this paper.

2 Background

With the advent of high speed wireless networks and portable devices, data requests based on the location of mobile clients have increased in number. However, there are several challenges to be met in the development of LAMSs [1], such as the constraints associated with the mobile environment and the difficulty of taking the user's movement into account. Hence, various techniques have been proposed to overcome these difficulties.

2.1 Hilbert-Curve

In [9], a linear index structure is proposed based on the Hilbert curve, in order to enable the linear broadcasting of objects in a multi-dimensional space. The Hilbert curve needs to allocate a sufficient number of bits to represent the index values, in order to guarantee that each of the points in the original space has a distinct value. If k is the number of bits used for a coordinate in the i-th dimension of the targeted m-dimensional space and n is the number of bits assigned to represent the coordinates, then a total of $\sum_{i=1}^{m} k$ bits need to be allocated to represent the coordinates and the expected time for the conversion is $O(n^2)$. Besides, to perform k-NN query processing using the Hilbert curve, more than two broadcast periods are needed.

2.2 Wireless Data Broadcast

The use of broadcasting methods that properly interleave index information and data on the broadcast channel can significantly improve not only the energy efficiency, but also the AT. Let N be the number of data objects and C be the download time for the required records. There are two parameters that need to be optimized in the one dimensional space of the AT and the TT [2][3]:

- Optimal Latency: The best AT is obtained when no index is broadcast along with the data. In this case, the size of the entire Bcast is minimized, and the average AT is $\frac{N}{2} + C$. However, in this case, the worst value of the average TT is obtained, since it is equal to $\frac{N}{2} + C$.
- Optimal Tune: This parameter allows the best TT to be obtained, while simultaneously increasing the AT. The server broadcasts the index at the beginning of each broadcast. In this case, the average TT is $1 + l + C$, where l is the number of levels in the multileveled index tree. In this case, the probe wait is equal to $\frac{N+index}{2}$ and the Bcast wait is equal to $\frac{N+index}{2} + C$. Since the Latency is the sum of the Probe Wait and the Bcast Wait, the average AT is equal to $(\frac{N+index}{2})\times$ 2+C= N + index + C.

3 Proposed Algorithms

In this section, we first introduce the broadcast-based LDIS scheme(BBS) [8]. Then, we describe algorithm for selective tuning, namely Exponential Sequence Scheme(ESS).

3.1 Sequential Data Broadcast

In a recent paper [8], we have proposed the concept of data sorting for broadcasting called BBS(Broadcast-based Location Dependent Data Delivery Scheme). In the BBS method, the server periodically broadcasts the IDs and coordinates of the data objects, without an index segment, to the clients, and these broadcasted data objects are sorted sequentially, according to the location of the data objects, before being sent to the clients. The BBS provides the fastest access time, since no index is broadcasted along with the data and thus, the size of the entire broadcast cycle is minimized. A preliminary simulation-based results showed that BBS is significantly reduce the AT.

A simple sequential broadcast can be generated by linearizing the two dimensional coordinates in two different ways: i.e., horizontal broadcasting(HB) or vertical broadcasting(VB). In HB, the server broadcasts the LDD(location dependent data) in horizontal order, that is, from the leftmost coordinate to the rightmost coordinate. On the other hand, in VB, the server broadcasts the LDD in vertical order, that is, from the bottom coordinate to the top coordinate. In this paper, we assume that the server broadcasts the data objects using HB.

Notations

- O_i: broadcast data object, where $O_i \in S$
- O_{candi}: candidate for the nearest data object
- O_c: current broadcast data object (initially O_c regarded as NN), where $O_c \in S$
- q: a query point
- O_{ps}: one of the data items broadcast before O_i in the current broadcast cycle, where $O_{ps} \in S$
- O_f: the client's first tuned data item in the broadcast channel
- Data_first: the server's first broadcast data item in the current broadcast cycle.
- Flag A: if the dist(x-coordinate of O_{ps}, x-coordinate of q) is larger than dist(O_c, q), then set to 1 (initially set to 0). This flag guarantees that the client dose not miss the NN in the current broadcast channel.
- Flag B: if the dist((x-coordinate of O_c),(x-coordinate of q))$<$ dist(O_{candi}, q)), then set to 1(initially set to 0).

There are two cases in which the clients tune to the broadcast channel.

Case 1: the client could not identify the NN in the current broadcast cycle, since it was not able to determine whether or not the desired data item was broadcasted before it tuned to the broadcast channel(see Fig 1).

Lemma 1: While Flag A=0 and $O_f \neq$ Data_first, the client could not identify the NN in the current broadcast cycle.

Proof: Let the client begin to tune at time T_i. At T_i, O_c=O_i and O_{candi} is O_i(see Fig 1). At T_{i+1}, O_c=O_{i+1} and O_{candi} is O_i, since dist(O_i, q)$<$dist(O_{i+1}, q). At T_{i+2}, O_c=O_{i+2} and O_{candi} is O_{i+2}, since dist(O_{i+2}, q)$<$ dist(O_i, q) and dist(O_{i+2}, q)$<$ dist(O_{i+1}, q). At T_{i+3}, O_c=O_{i+3} and O_{candi} is O_{i+2}, since dist(O_{i+2}, q)$<$ dist(O_{i+3}, q). Then, the client stops tuning to the broadcast channel, since dist(O_{i+2}, q)$<$dist(x-coordinate of O_{i+3}, x-coordinate of q). However, the client could not guarantee O_{i+2} as the NN, since it missed the O_{i-n},

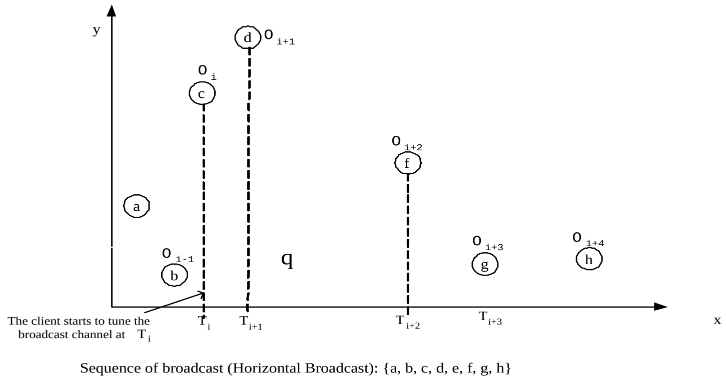


Fig. 1. The client could not identify the NN in the current broadcast cycle

such as O_{i-1} data item, and one of them could also be O_{candi}. Thus, if Flag A=0 and Flag B=1, then the client stops tuning to the broadcast channel and switches to doze mode, until the Data_first of the next period arrives. □

Case 2: the client is able to identify the NN from the current broadcast cycle, since it is sure that the desired data item is going to appear in the current broadcast cycle(see Fig 2).

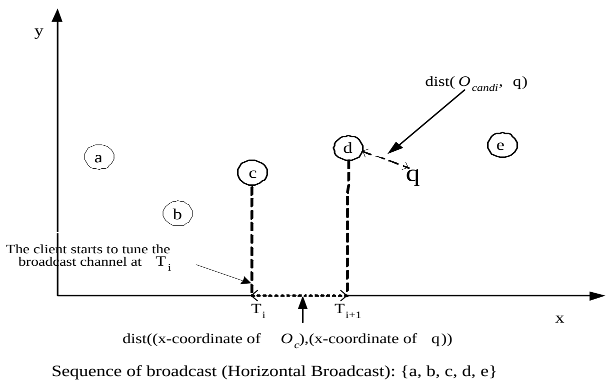


Fig. 2. The client is able to identify the NN in the current broadcast cycle

Lemma 2: If Flag A is set to 1, then the client can identify the NN from the current broadcast cycle.

Proof: Let O_c be data item 'c' in Fig 2. After the client receives data item 'd', Flag A is set to 1, which means that data item 'c' and the data items before 'c', such as 'a' and 'b', are not candidates for the NN, since the distances from the x-coordinates of 'a', 'b', and 'c' to the x-coordinate of q are longer than dist(d, q). Consequently, we can conclude that the client does not miss the NN and can find it from the current broadcast channel. □

Definition 1: If O_c is Data_first, then the client can identify the NN from the current broadcast cycle.

3.2 Energy Efficient Selective Tuning

With the BBS scheme [8], the clients can significantly reduce their access time, since this scheme eliminates the **probe wait time**(see introduction) for the clients. However, the average TT may increase, since the client has to tune to the broadcast channel until the desired data item has arrived. In the previous index schemes [3], each data item contains a pointer to the next occurrence of the index segment and, additionally, every data item contains a pointer to the next data item that has the same attribute value [3]. In our scheme, every data item contains pointers that contain the IDs, locations and arrival times of the data items that will subsequently be broadcasted. In this section, we present energy efficient scheme for the BBS environment, namely the Exponential Sequence scheme. These scheme provides the clients with the ability to perform selective tuning and help to reduce the client's tuning time.

Notations

- T_i: boundary lines of the current broadcast data object
- T: set of T_i,
- TN: nearest boundary line on the left-hand side of the q
- ON: data object of TN
- TS: safe nearest boundary line on the left-hand side of the q, where x-coordinate of $TS \leq$x-coordinate of TN

Exponential Sequence Scheme(ESS) In this section, we present a selective tuning method for use in the BBS environment. In this method, the client uses exponential pointers from each data item for the purpose of reducing energy consumption. Each data item contains the following information;

- It's ID and location information.
- Initial pointer: arrival time of the first data item to be broadcasted in the next broadcast cycle
- Forward Pointer(FP): IDs locations and arrival times of the data items that will be broadcasted at T_i. The maximal number of FP from the each data item is $\log_e N$, where e is the exponent value and N is the number of data items that will be broadcasted(e.g., if $e = 2$ and N=32, the first broadcasted data item $O1$ has the following FPs: the data items located in $T2$, $T3$, $T4$, $T5$ and $T6$(see Fig 3))

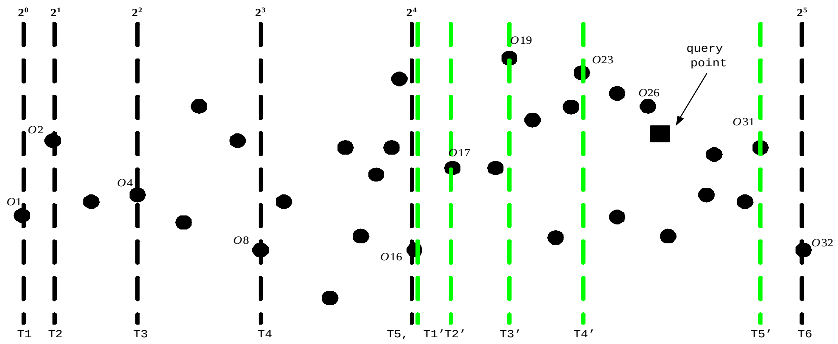


Fig. 3. An example of Exponential Sequence Scheme

The client obtains the ID, location information and FP from the first tuned data item. Then, it switches to doze mode until the desired data item appears on the broadcast channel. The client repeatedly switches between the doze and wake up modes until it finds the desired data item. Let us consider the example in Fig 3. First, the client tunes to the broadcast channel at $T1$ and obtains the following pointers ID={2,4,8,16,32} from data item $O1$. Then, it switches to doze mode until $O16$(at $T5$) has appeared on the broadcast channel, since $T5$ is the nearest boundary line on the left-hand side of the query point q up to the present time. Then, the client wakes up at $T5$, and obtains FP ID={17,19,23,31} from data item $O16$. The client again switches to doze mode until $O23$(at $T4'$) has appeared on the broadcast channel, since $T4'$ is the nearest boundary line on the left-hand side of the query point q up to the present time. Finally, the client wakes up at $T4'$ and follows the algorithm 1 and returns data item $O26$ as the NN.

The client uses the following algorithm to perform selective tuning(This algorithm is adapted to ESS).

Algorithm 1. the client algorithm used to identify the nearest object

Input: locations of the clients and the data objects;
Output: NN;
Procedure:

1: **do**
2: read O_c
3: **for**(T)
4: **if**(T_i=T_N)
5: **then** T_i=T_{i-1}
6: **do** compare(dist(ON, q) and dist(x-coordinate of T_i, x-coordinate of q)
7: **if**(dist(ON, q)<dist(x-coordinate of q)
8: **then** TS=T_{i-1} and the client goes into doze mode until the data object of TS appears on the channel
9: **else** T_i=T_{i-1}
10: **while** find out TS
11: **else** T_i=T_{i+1}
12: **while**(x-coordinate of $T_1 \leq$x-coordinate of ON) // T_1= TN

```
if(x-coordinate of T2 ≤ x-coordinate of ON)
then
    do compare dist(O_c, q) and dist(O_p, q)
        if(dist(O_c, q) < dist(O_p, q))
            then O_c = O_candi
        else O_p = O_candi
    while dist((x-coordinate of O_c),(x-coordinate of q)) < dist(O_candi, q))
O_candi = NN
return NN
```

4 Performance Evaluations

In this section, we evaluate the performance with various kinds of parameter settings.

4.1 Energy Consumption Model

Let $\mathcal{E}$ denotes the amount of energy consumption and the "*ec*" denotes the energy coefficient which means the active-to-doze ratio, $\frac{\mathcal{E}_{ACTIVE}}{\mathcal{E}_{DOZE}}$.

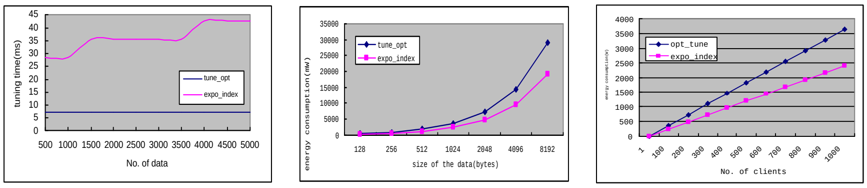
In this paper, the average energy consumption can be measured by the amount of unit energy in a given time. In order to choose reasonable coefficients, we should have some reference values. Then, the energy coefficients($\widehat{ec}$) can estimated. In our experiment, parameters *ec* is fixed with 48.61.

4.2 Experimental evaluation

This section presents the numerical results with the various environmental variables. We assume that the broadcast data objects are static, such as restaurants, hospitals and hotels. In this paper, two datasets are used in the experiments. The first data set, $\mathcal{D}1$, contains data objects randomly distributed in a square Euclidian space, while the second data set, $\mathcal{D}2$, contains the data objects of hospitals in the Southern California area, and is extracted from the data set available at [7].

4.3 Tuning Time and Energy Consumption

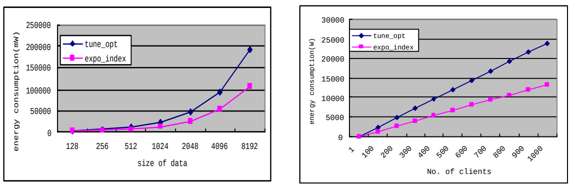
Fig 4(a) shows the tuning time as the number of data items increases. As shown in this figure, tune_opt outperforms the other schemes, since the average tuning time of tune_opt is $1 + l + C$ [3]. Fig 4(b) shows the energy consumption as the size of the data items is increased from 128 bytes to 8192 bytes in $\mathcal{D}1$. As shown in this figure, in this case, the proposed schemes outperform tune_opt, since tune_opt minimizes the tuning time but not the access time, while the proposed schemes reduce the tuning time and AT at the same time. In other words, when we estimate the energy consumption, it is necessary to consider not only the active time, but also the doze time, even if the doze time is considerably smaller



(a) (b) (c)

Fig. 4. Tuning time and energy consumption

than the active time(i.e., we assume a ratio of $\frac{1}{48.61}$). Fig 4(c) shows the energy consumptions as the number of clients is increased in $\mathcal{D}1$. Figs 5(a) and 5(b) show the energy consumptions as the size of the data items and the number of clients are increased in $\mathcal{D}2$, respectively.



(a) (b)

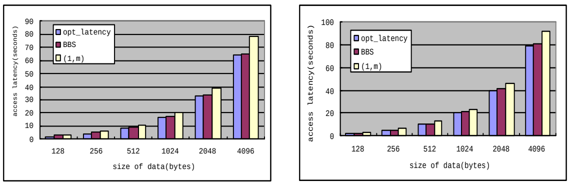
Fig. 5. Energy consumption in $\mathcal{D}2$

4.4 Access Time

In this section, we evaluate the AT. Fig. 6(a) shows the AT as the size of the data items increase in $\mathcal{D}1$. Finally, Fig. 6(b) shows the AT as the size of the data items increase in $\mathcal{D}2$. Since the clients do not need to wait and tune into the broadcast channel to receive an index segment, if they have already identified the desired objects, the BBS scheme shows increasingly lower latency compared to the other scheme.

5 Conclusion

We explored the different broadcasting and tuning methods which can be used for NN query processing. For the purpose of broadcasting in LAMSs, we present the BBS method, and for the purpose of selective tuning with the BBS method,



(a) (b)

Fig. 6. Access time in $\mathcal{D}1$ and $\mathcal{D}2$

we present the ESS method. The BBS method attempts to reduce the AT, and the ESS method attempts to conserve battery power. The experimental results show that the proposed ESS scheme significantly reduces not only the AT, but also the energy consumption, since the client does not always have to wait for the index segment.

References

1. D. L. Lee, J. Xu, and B. Zheng, "Data Management in Location-Dependent Information Services," *IEEE Pervasive Computing*, 1(3), 2002.
2. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Energy efficient indexing on air," *Proc. of SIGMOD*, 1994.
3. T. Imielinski, S. Viswanathan, and B.R.Badrinath, "Data on Air: Organization and Access," *IEEE Trans. Knowledge and Data Eng*, 9(3), 1997.
4. W. C. Lee and D. L. Lee, "Using signature techniques for information filtering in wireless and mobile environments," *Distributed and Parallel Databases*, 4(3), 1996.
5. Q. L. Hu, W.-C. Lee, and D. L. Lee, "A hybrid index technique for power efficient data broadcast," *Distributed and Parallel Databases*, 9(2), 2001.
6. B. Zheng, J. Xu, and D. L. Lee, "Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments," *IEEE Trans. Comp.*, 51(10), 2002.
7. Spatial Datasets, `http://dias.cti.gr/~ytheod/research/datasets/spatial.html`.
8. K. Park, M. Song, and C.-S. Hwang, "An Efficient Data Dissemination Schemes for Location Dependent Information Services," In *Proc. of ICDCIT*, 2004. Lecture Notes Computer Science.
9. B. Zheng, W.-C. Lee, D. L. Lee, "Search Continuous Nearest Neighbor on Air," In *Proc. The First International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, Massachusetts, August 2004.
10. M.-S. Chen, P. S. Yu, and K. L. Wu, "Optimizing index allocation for sequential data broadcasting in wireless mobile computing," *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 2003.
11. N. Roussopoulos, S. Kelley, and F. Vincent. "Nearest neighbor queries," *Proc. of SIGMOD*, 1995.