A Novel Air Indexing Scheme for Window Query in Non-Flat Wireless Spatial Data Broadcast

Seokjin Im, Hee Yong Youn, Jintak Choi, and Jinsong Ouyang

Abstract: **Various air indexing and data scheduling schemes for wireless broadcast of spatial data have been developed for energy efficient query processing. The existing schemes are not effective when the clients' data access patterns are skewed to some items. It is because the schemes are based on flat broadcast that does not take the popularity of the data items into consideration. In this paper, thus, we propose a data scheduling scheme letting the popular items appear more frequently on the channel, and grid-based distributed index for non-flat broadcast (GDIN) for window query processing. The proposed GDIN allows quick and energy efficient processing of window query, matching the clients' linear channel access pattern and letting the clients access only the queried data items. The simulation results show that the proposed GDIN significantly outperforms the existing schemes in terms of access time, tuning time, and energy efficiency.**

Index Terms: **Air index, grid-based distributed index (GDIN), non-flat wireless data broadcast, spatial queries.**

I. INTRODUCTION

Wireless data broadcasting is an effective way for information services supporting a massive number of mobile clients since it can simultaneously accommodate any arbitrary number of clients [1]. Here the server disseminates data via a wireless channel in buckets as the smallest logical access unit. While monitoring the channel, each client processes its own query independently and picks its desired data items on the channel. The systems maintaining spatial data items provide location dependent information services through a variety of spatial queries such as window query and k nearest neighbor (kNN) query [2]–[4].

Due to limited battery power, the clients operate in one of the two modes: Active mode for listening to the wireless channel and doze mode for saving the energy. The client's performance is characterized by access time and tuning time [1]. Access time refers to the time elapsed from the time a given query is issued to the time it is satisfied. Tuning time is the amount of time for the client to stay in active mode during data access. Tuning time can be reduced by means of air indexing in which the index information showing the broadcasting times of the items is interleaved with data items. The air indexing scheme allows each client to listen selectively to its only desired items.

The wireless data broadcast systems can be categorized by the uniformity of the broadcast frequency of data items. In a flat data broadcast, all data items are disseminated by the same frequency in a broadcast cycle. In a non-flat data broadcast, popular data items are disseminated more frequently than unpopular data items. Non-flat data broadcast, thus, reduces the average access time in case of skewed data access patterns.

For a non-flat data broadcast, various data scheduling and indexing schemes on the wireless channel have been researched to minimize the access time and energy consumption required for query processing. The existing data scheduling and indexing schemes, however, are mostly based on non-spatial data items [5]–[8], or flat data broadcast for spatial data items [2]–[4]. In this paper we focus on non-flat data broadcast of spatial data items over a single wireless channel to efficiently provide location dependent information services. Here, we propose a data scheduling scheme disseminating popular data items more frequently, and a novel indexing scheme called grid-based distributed index for non-flat broadcast (GDIN) to support energy efficient processing of window query. To the best of our knowledge, this is the first work on indexing scheme targeting non-flat broadcast of spatial data. The main objective of the proposed approach is to allow quick access to popular items and energy efficient query processing by extracting the desired items using their original coordinates. It is achieved by employing the linear table structure matching the clients' linear channel access pattern. We evaluate the performance of the proposed GDIN scheme by simulation and compare it with the existing schemes. The simulation results show that the proposed scheme significantly outperforms the existing schemes in terms of the access time, tuning time and energy conservation.

The rest of the paper is organized as follows. Section II summarizes the related works and Section III presents the proposed GDIN scheme. In Section IV the performance of GDIN is evaluated and compared. Finally, we conclude the paper in Section V.

II. THE RELATED WORKS

In this section we briefly discuss the existing works related to non-flat broadcast and air indexing of spatial data items. In [5], the authors proposed a data scheduling scheme on a wireless channel called broadcast disk, which considers nonuniform data access distribution. According to the access rate, data items are grouped into several logical disks assigned to different broadcast frequencies. Popular items' disks have higher broadcast frequencies and data broadcasting occurs by circularly taking the items from the disks according to their relative frequencies. As

Manuscript received December 04, 2008; approved for publication by Hussein Mouftah, Division III Editor, April 14, 2011.

This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2008-357-D00244).

S. Im and J. Ouyang are with the California State University at Sacramento, 6000 J Street Sacramento, CA, email: imseokjin@gmail.com, ouyangj @ece.csus.edu.

H. Y. Youn is corresponding author with the Sung Kyun Kwan University, Suwon, Korea, email: youn@ece.skku.ac.kr.

J. Choi is with the University of Incheon, Incheon, Korea, email: choi@incheon.ac.kr.

1229-2370/11/$10.00

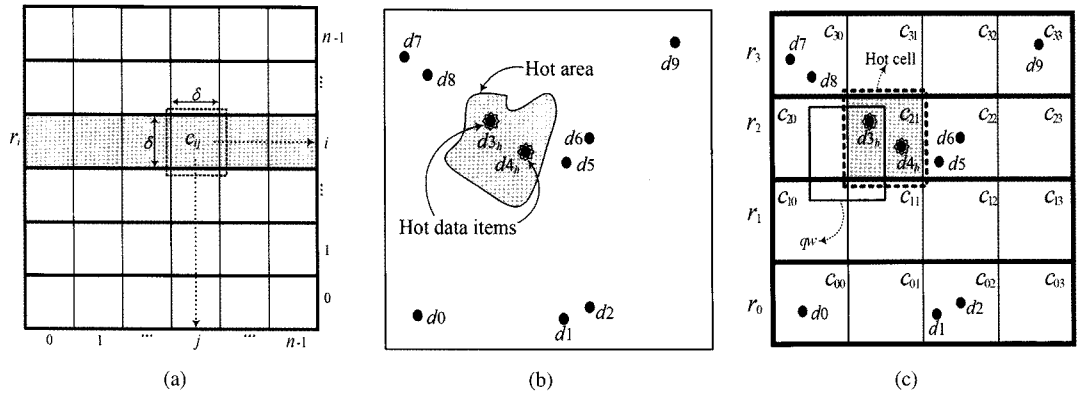


Fig. 1. The structure employed in the proposed scheme: (a) Data space D and its grid partition, (b) an example of data space D with 10 data items, and (c) 4×4 grid for Fig. 1(b).

a result, the clients can quickly access popular items. In [6], the authors proposed an optimal broadcast scheduling approach considering nonuniform data access pattern of the clients. Also, time-critical scheduling of data items was proposed in [7]. Recently, an air indexing scheme called MHash was proposed for energy-efficient query processing and optimized the access time for non-flat broadcast [8]. These works are for non-flat broadcast of non-spatial data.

Wireless data broadcast of spatial data researched so far has been based on flat broadcast. Various indexing schemes of spatial data have been proposed for energy efficient query processing in wireless data broadcast environments. In [2], the authors propose Hilbert curve index (HCI) to efficiently support spatial queries such as window query and kNN query, which employs B+ tree constructed on Hilbert curve (HC). It is obtained with the integer numbers representing the locations of the data items. HCI matches well the clients' linear channel access pattern. However, it causes the clients to consume excessive energy for numerous candidate items to extract the queried items. In [3], the distributed spatial index (DSI) over HC was proposed, which adopts a distributed table to allow a quick start of query processing. It shows improved performances compared to HCI. However, it still causes the clients large energy consumption because of the same reason as HCI. In [4], the authors proposed cell-based distributed index (CEDI) for window query with two-level tables indexing the grid space. It enables the clients to process the given query energy efficiently by letting them listen to only the queried data items using their original coordinates.

The existing schemes for spatial data mentioned above are for flat broadcast. As a result, they cause long access time under nonuniform distribution of data access. In this paper we develop a scheme for window query targeting non-flat broadcast of spatial data.

III. GRID-BASED DISTRIBUTED INDEX FOR NON-FLAT BROADCAST

In this section we present the GDIN, which is based on grid partition. The goal of GDIN is to efficiently support the window query with query window qw in terms of the tuning and access time in non-flat broadcast environments. Here, qw is a rectangle defined by (LL_x, LL_y) and (UR_x, UR_y), the coordinates of the lower-left and upper-right corner, respectively.

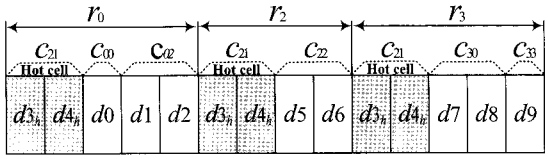


Fig. 2. The scheduling of data items on the channel for the example of Fig. 1(c).

In order to achieve the goal, we partition data space D into $n \times n$ grid for effective data scheduling on the wireless channel. Fig. 1(a) shows data space D and its $n \times n$ grid partition with δ as the length of all sides of the cells. As shown in Fig. 1(a), we label the ith row of the grid as r_i and the cell of the ith row and jth column of the grid as c_{ij}, here $0 \leq i, j \leq n-1$.

A. Data Scheduling on the Channel

In order to consider the popularity of spatial data in non-flat broadcast, we define the area queried more frequently in D as the hot area. We call a data item in the hot area as a hot data item, denoted d_h, and a cell containing the hot data items as a hot cell. For example, Fig. 1(b) shows data space D with 10 data items from d_0 to d_9, in which hot area is defined as shown. The two data items, d_3 and d_4, in the hot area are defined as hot data items and denoted as $d3_h$ and $d4_h$. Fig. 1(c) shows a 4×4 grid laid upon data space D in Fig. 1(b). Here, cell c_{21} containing the two hot data items is defined as the hot cell.

The data items are scheduled in row major order on the wireless channel. For the scheduling, all the cells with data items are placed in row major order. Then, for disseminating hot data items more frequently than the other data items, all hot cells are placed in the beginning of the transmission of every row. This makes it possible for the clients to access hot data items quickly. Fig. 2, for example, shows the scheduling of the data items of Fig. 1(c). As a result, in an even data distribution with all rows containing data items, hot cells are replicated n times, and the clients can access hot items within $1/2n$ time of the broadcast cycle in average.

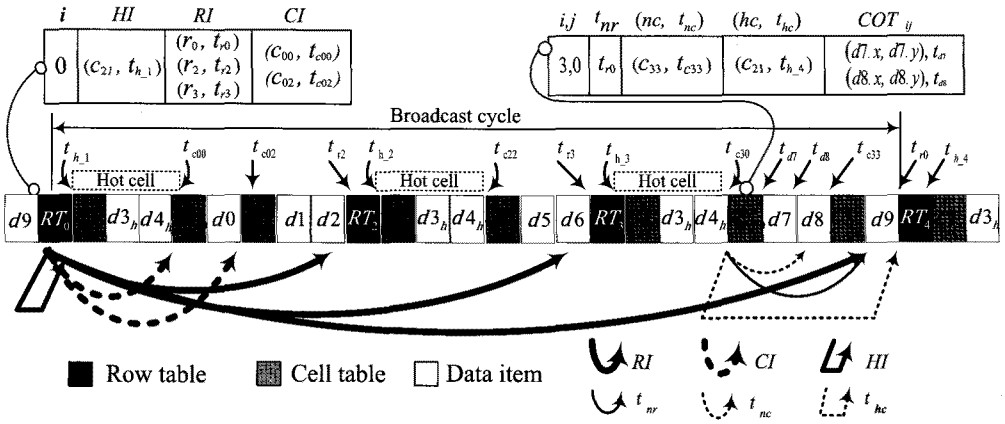


Fig. 3. The structure of broadcast channel with GDIN for the example of Fig. 1(c).

B. Indexing Tables

GDIN employs tables for indexing rows, cells and data items on the wireless channel. The tables are organized in two levels as follows: The upper level row table, RT_i of r_i, containing the information on the rows and the cells of r_i having data items and hot cells; the lower level cell table, CT_{ij} of c_{ij}, containing the information on the data items in c_{ij}. The row table and cell table are not made if the row or cell is empty such as r_1 and c_{01} of Fig. 1(c).

RT_i is constructed as follows:

$$RT_i = \langle i, HI, RI, CI \rangle, \text{for } 0 \le i \le n-1. \quad (1)$$

- Hot cell information (HI): $\{< c_{ab}, t_{ab} > \mid 0 \le a,\ b \le n-1\}$ where c_{ab} is the hot cell containing hot data items and t_{ab} is the time when CT_{ab} of c_{ab} is broadcast on the wireless channel.
- Row information (RI): $\{< r_a, t_a > \mid 0 \le a \le n-1\}$ where r_a is the row having data items and t_a is the time when RT_a of r_a is broadcast on the wireless channel.
- Cell information (CI): $\{< c_{ib}, t_{ib} > \mid 0 \le b \le n-1\}$ where c_{ib} is the non-empty cell belonging to r_i and t_{ib} is the time when CT_{ib} of c_{ib} is broadcast on the wireless channel.

HI differentiates the proposed indexing scheme from the existing ones. It enables the access to hot data items within $1/2n$ of the length of a broadcast cycle under the assumption of even distribution of data items over the entire area. With RI and CI, the clients can directly access their desired row or cell without referring to other tables and get the information on empty rows and cells. RI and CI provide the clients with multiple access paths using the links leading to other row tables and cell tables, which results in quick data access. For example, Fig. 3 shows RT_0 of r_0 and we explain RT_0 in detail in subsection III-C.

CT_{ij} is constructed as follows

$$CT_{ij} = \langle (i,j),\ t_{nr}, (nc, t_{nc}), (hc, t_{hc}), COT_{ij} \rangle \quad \text{for } 0 \le i,j \le n-1. \quad (2)$$

- t_{nr} is the broadcasting time for the first row table appearing immediately after c_{ij} on the wireless channel.
- (nc, t_{nc}): nc is the first regular cell appearing immediately after c_{ij} on the wireless channel and t_{nc} is the broadcasting time for the cell table of nc.
- (hc, t_{hc}): hc is the first hot cell appearing immediately after c_{ij} on the wireless channel and t_{hc} is the broadcasting time for the cell table of hc.
- COT_{ij} (coordinate table): $((d_x, d_y), t_d) | (d_x, d_y)$ is the coordinates of data item, d,belonging to c_{ij} and t_d is the time when d is broadcast on the wireless channel.

t_{nr} and t_{nc} enable the clients to successively access the row table and cell table. t_{hc} forms a chain of broadcast time of hot cells to allow the clients to successively access the hot cells. COT_{ij} enables the clients to extract the items of c_{ij} in the query window. For example, Fig. 3 shows CT_{30} of c_{30} and we explain CT_{30} in detail in subsection III-C.

C. Broadcast Channel Structure

With the proposed GDIN, the wireless broadcast channel is structured as follows. Using the data scheduling mentioned in subsection III-A, row table RT_i is followed by hot cells and cells in r_i. Also, cell table CT_{ij} is followed by the data items within that cell c_{ij}.

Fig. 3, for example, shows the structure of the wireless broadcast channel based on the data scheduling shown in Fig. 2. In Fig. 3, RT_0 is followed by hot cell c_{21} and cell c_{00} and c_{02} in r_0. CT_{00} is followed by data item d_0 in c_{00}.

Fig. 3 shows RT_0 of r_0 shown in Fig. 1(c). HI of RT_0 holds the information of hot cell c_{21} and the link of HI denoted by a thick straight line lets the clients access c_{21} directly. RI of RT_0 holds the information of all rows with data items, i.e., r_0, r_2 and r_3. The links of RI denoted by thick curvy lines let them access other row tables, RT_2 and RT_3. CI of RT_0 holds the information of cells with data items in r_0, i.e., c_{00} and c_{02}. The links of CI denoted by thick dotted curvy lines let them access c_{00} and c_{02}.

Also, Fig. 3 shows CT_{30} of c_{30} shown in Fig. 1(c). Using t_{nr} of CT_{30} represented by a thin curvy line, the clients can access RT_0, the row table appearing immediately after CT_{30} on the channel. Also, they can access CT_{33} successively using t_{nc} represented by a thin dotted curvy line and access CT_{21} of hot cell c_{21} using t_{hc} represented by a thin dotted straight line.

Table 1. The parameters used in the simulation.

Parameter	Description	Setting
$DataSize$	The size of one data item	1024 Bytes
$IDSize$	The size of the identifier representing row/ grid cell number	8 Bytes
$FloatSize$	The size of one floating point number	8 Bytes
$BucketSize$	The size of one bucket	64 Bytes
$WindowRatio$	The ratio of one side of query window to one side of the test area	0.01, 0.04, 0.07, 0.1
N	The number of data items	5922, 16000
N_{hot}	The number of hot data items	$0.1N$
N_q	The number of window queries issued for one simulation condition	100,000
R_{hot}	The portion of queries of N_q accessing hot data items	0.5~1.0
$Bandwidth$	The bandwidth of the wireless broadcast channel	1 Mbps

With COT_{ij} of CT_{30}, the clients can filter out the data items contained in a given query window among d_7 and d_8.

D. Window Query Processing

With the proposed GDIN, each client processes independently its own window query with the query window qw by following 2 steps: (Step 1) determine Q, which is the set of the cells overlapped with the given qw. (Step 2) while accessing the cells in Q using RT_i and CT_{ij} on the channel, extract the data items within qw from each accessed cell and then download them from the wireless channel.

- Step 1. Each client determines Q as

$$Q=\left\{c_{ij}\middle|\left\lfloor\frac{LL_y}{\delta}\right\rfloor\le i\le\left\lfloor\frac{UR_y}{\delta}\right\rfloor \text{ and } \left\lfloor\frac{LL_x}{\delta}\right\rfloor\le j\le\left\lfloor\frac{UR_x}{\delta}\right\rfloor\right\}$$

where (LL_x, LL_y) and (UR_x, UR_y) are the coordinates of the lower-left and upper-right corner of qw, respectively, and δ is the length of all sides of the cells as mentioned earlier. For example, Q is determined as $\{c_{10}, c_{11}, c_{20}, c_{21}\}$ for qw shown in Fig.1(c).

- Step 2. Using RT_i and CT_{ij}, each client executes 3 tasks to obtain its own query results as follows.

(Task 1) Determine t_n using RT_i or CT_{ij} accessed currently. Here, t_n is the broadcast time for the row or cell table to be accessed for Q. t_n guides the client to the first cell among the ones within Q appearing on the channel immediately after RT_i or CT_{ij} accessed currently. The procedure for this with RT_i or CT_{ij} is as follows.

With RT_i of r_i,
$$\begin{cases} t_n \leftarrow \infty \\ \text{for all } < c_{ab}, t_{ab} > \text{ in } HI \text{ of } RT_i \\ \text{ if } \{(c_{ab} \text{ in } Q) \text{ and } (t_n > t_{ab})\} \text{ then } t_n \leftarrow t_{ab} \\ \text{for all } < c_{ib}, t_{ib} > \text{ in } CI \text{ of } RT_i \\ \text{ if } \{(c_{ib} \text{ in } Q) \text{ and } (t_n > t_{ib})\} \text{ then } t_n \leftarrow t_{ib} \\ \text{for all } < r_a, t_a > \text{ in } RI \text{ of } RT_i \\ \text{ if } \{(r_a \text{ overlapped with } qw) \text{ and } (t_n > t_a)\} \\ \text{then } t_n \leftarrow t_a. \end{cases}$$

With CT_{ij} of c_{ij},
$$\begin{cases} t_n \leftarrow \infty \\ \text{if } \{(hc \text{ in } Q) \text{ and } (t_n > t_{hc})\} \text{ then } t_n \leftarrow t_{hc} \\ \text{if } \{(c_{ij} \text{ and } nc \text{ are in the same row }) \text{ and } (t_n > t_{nc})\} \\ \text{then } t_n \leftarrow t_{nc} \text{ else } \ t_n \leftarrow t_{nr}. \end{cases}$$

(Task 2) Remove empty cells and the already accessed cell from Q. When the client accesses RT_i, it removes all the cells in each empty row using RI of RT_i. Then, it removes the empty cells of r_i using CI of RT_i, where row r_i has data items. Also when the client accesses CT_{ij} of c_{ij} in Q, it removes the accessed cell c_{ij} from Q. The procedure for this with RT_i or CT_{ij} is as follows.

With RT_i of r_i,
$$\begin{cases} \text{for all row } r_a \ (0 \le a \le n-1) \\ \text{ if } (r_a \text{ not in } RI), \text{ remove all cells of } r_a \text{ from } Q \\ \text{for all cell } c_{ib} \text{ in } r_i (0 \le b \le n-1) \\ \text{ if } (c_{ib} \text{ not in } CI), \text{ remove } c_{ib} \text{ from } Q. \end{cases}$$

With CT_{ij} of c_{ij}, remove c_{ij} from Q.

For example, when the client accesses RT_2, using RI of RT_2 it removes c_{10} and c_{11} of the empty row r_1 in Fig. 1(c) from Q. Also using CI of RT_2 it removes from Q the empty cell c_{20} of r_2 in Fig. 1(c). When the client accesses CT_{21} of c_{21}, it removes the accessed cell c_{21} from Q.

(Task 3) Extract the broadcasting time of the data items within qw and download them. When the client accesses CT_{ij} of $c_{ij} (\in Q)$, it extracts the data items within qw from all data items in c_{ij} using their coordinates included in COT_{ij}. Then it puts t_d for each of the extracted items into $dpQueue$ from COT_{ij}. Here, $dpQueue$ is a queue keeping the broadcasting time. Then, the client downloads them from the channel at the time in $dpQueue$. For example, when the client accesses CT_{21} on the channel, using COT_{21} of CT_{21} it extracts $d3_h$ within qw shown in Fig. 1(c). It puts t_{d3_h}, the broadcast time of $d3_h$ on the channel, into $dpQueue$ and then downloads $d3_h$ from the channel at t_{d3_h}.

The algorithm for processing a given window query is provided in Algorithm 1. The algorithm shows the procedures for the 2 steps and 3 tasks contained within step 2 mentioned above. Step 1 is executed in line 1 and step 2 is performed by the while-loop from line 3 to 19, respectively. The while-loop is repeated until all the cells in Q are accessed as shown in line 16 and then Result is returned in line 20.

In step 2 with CT_{ij}, the client executes the 3 tasks using from line 4 to 11. Task 1 is executed by the function $doTask1WithCT(\cdot)$ in line 5, which returns t_n. Task 2 is executed in line 7 and task 3 in lines 8 and 9, respectively. The function $doTask3WithCT(\cdot)$ in line 8 returns the broadcasting time for each data item within qw.

Algorithm 1 *WindowQuery*

Input: qw, a query window
Output: Result, a set of data items belonging to qw
1: $Q \leftarrow$ the grid cells overlapped with qw; // Step 1
2: Table $\leftarrow$ the indexing table firstly encountered after tuning-in;
3: **while true do** // Step 2
4: **if** (Table is CT_{ij} of c_{ij}) **then**
5: $t_n \leftarrow doTask1WithCT()$; // Task 1
6: **if** ($c_{ij} \in Q$) **then**
7: Remove c_{ij} from Q; // Task 2
8: $dpQueue \leftarrow doTask3WithCT()$; // Task 3
9: Result $\leftarrow$ all data items after accessing at the times in $dpQueue$;
10: **end if**;
11: **end if**;
12: **if** (Table is RT_i of r_i) **then**
13: $t_n \leftarrow doTask1WithRT()$; // Task 1
14: remove all the empty cells from Q with RI and CI; // Task 2
15: **end if**;
16: **if**($Q == \phi$) **then break**;
17: **else** switch to the doze mode and wake up at t_n;
18: **end if**;
19: **end while**;
20: **return** Result ;

With RT_i, the client executes task 1 and task 2 using from lines 12 to 15. Task 1 is executed by the function $doTask1WithRT(\cdot)$ in line 13, which returns t_n, and task 2 in line 14, respectively.

IV. PERFORMANCE EVALUATION

In this section we evaluate the proposed GDIN in terms of the access time, tuning time and energy consumption. Then, we compare it with the existing indexing schemes for spatial data items, i.e., HCI, DSI, and CEDI.

A. Simulation Environments

For this evaluation, we implemented the testbed with the discrete time simulation package SimJava [9]. The testbed consists of a broadcast server, a single wireless broadcast channel and a mobile client. This is because all the clients are independent of each other, i.e., each client processes independently its own query and downloads its desired data items from the wireless broadcast channel. As the network model used in the testbed, we modeled direct-band network operated by Microsoft, which is an FM radio-based broadcast network using FM radio subcarrier frequencies [11]. We assumed that the modeled network has 1 Mbps transmission rate of the wireless broadcast channel.

The simulations are conducted on two datasets, i.e., uniformly distributed dataset (UNIFORM) of 16,000 items, and real dataset (REAL) of 5922 cities of Greece. The parameters used in the simulation are shown in Table 1. Here, R_{hot} indicates the skewness of the queries issued to hot data items and $N_q R_{\text{hot}}$ is the number of queries accessing hot data items among N_q queries. For each simulation run, the average access time and tuning time are obtained in units of buckets after N_q queries are executed.

Due to space limitation, we report the simulation results with $n = 16$ for GDIN and CEDI as the default value. For fair comparisons with HCI, we set the number of replication of index information on the wireless channel to the value minimizing the access time with HCI. For the same reason, the exponential base of DSI is set to 2.

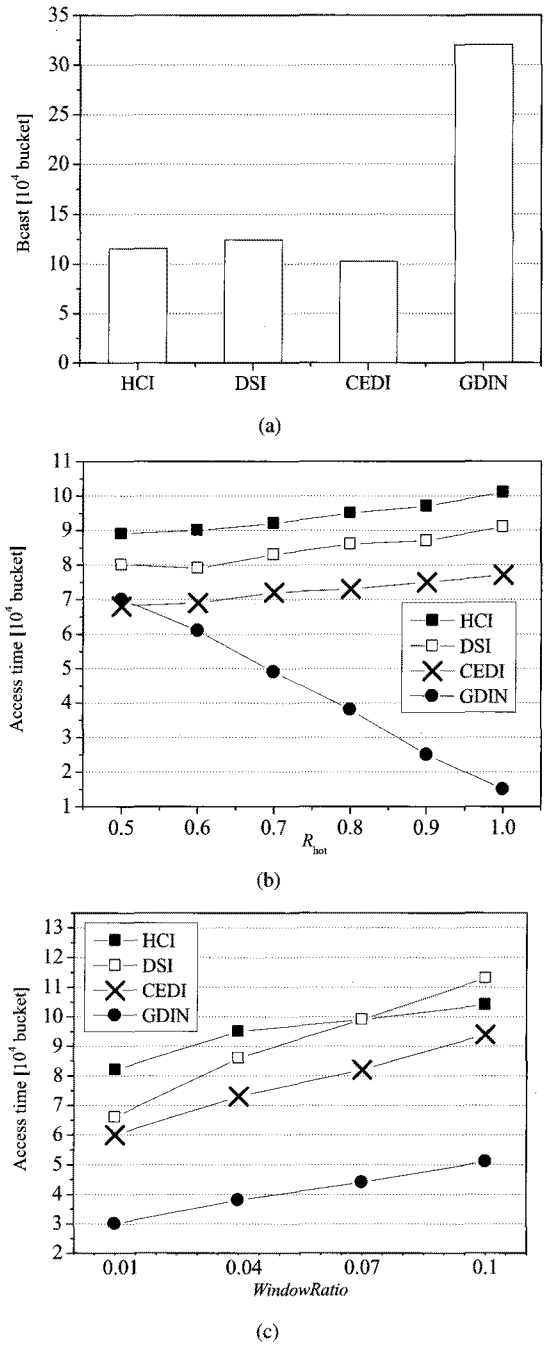


Fig. 4. The comparison of the length of a broadcast cycle and the access time for REAL dataset: (a) The comparison of the length of a broadcast cycle, (b) the access time with different R_{hot}, and (c) the access time with different window ratio.

B. Comparison of Bcast and Access time

In this section, we compare the proposed GDIN with the existing schemes in terms of Bcast, the length of a broadcast cycle, and the access time.

Fig. 4(a) shows Bcast of the compared indexing schemes for REAL dataset. Bcast of GDIN is much longer than other schemes because GDIN is for non-flat broadcast scheme, in which hot data items are replicated n times while other schemes are for flat broadcast. Bcast of GDIN for UNIFORM dataset also has the same trend with that for REAL dataset. The longer

Bcast of GDIN by a single broadcast channel may cause the access time for unpopular data items to be excessively long. However, GDIN targets accessing popular data items quickly and reducing the average access time of unpopular and popular data items in the skewed data access patterns. We show that GDIN outperforms other schemes in the average access time on various skewed data access patterns as shown in the following simulation results.

From now on, we compare the access time. Fig. 4(b) shows the access time for REAL dataset with various R_{hot} values. Here, $WindowRatio$ is 0.04. Observe from the figure that GDIN outperforms the other schemes if R_{hot} is greater than 0.5. As R_{hot} increases, i.e., the client's data access pattern is more skewed to hot items, the access time of GDIN decreases. This is because GDIN allows the client to access hot data items quickly by replicating them n times on the channel using the indexing information on hot cells. With UNIFORM dataset, the access time of GDIN shows the same trend as with REAL dataset which is about 52%, 44%, and 40% of that of CEDI, DSI, and HCI, respectively, when R_{hot} is 0.8.

Fig. 4(c) shows the access time for REAL dataset with various $WindowRatio$ values when R_{hot} is 0.8. It shows that the proposed GDIN also enables the client to access hot items more rapidly than other schemes when the data access pattern is skewed regardless of $WindowRatio$ values. With UNIFORM dataset, the access time of GDIN is about 43%, 39%, and 44% of that of CEDI, DSI, and HCI, respectively, for $WindowRatio$ of 0.1.

C. Comparison of Tuning Time

In this section, GDIN is compared with other schemes in terms of the tuning time. Fig. 5(a) shows the tuning time for REAL dataset with various R_{hot} values when $WindowRatio$ is 0.04. Observe that the tuning time of GDIN is much shorter than that of DSI and HCI, and almost same as CEDI. This is because GDIN and CEDI make the client access only the target data items residing in the given qw by extracting them using their original coordinates before accessing them, while HCI and DSI let the client extract them after making accesses to excessive candidate data items.

The tuning time of GDIN increases as R_{hot} increases since the number of data items in the given qw belonging to hot area becomes larger than that of data items in the qw belonging to non-hot area. Note that we set the area dense with data items as hot area. However, with UNIFORM dataset, the tuning time of GIDN is almost constant since the number of data items in the given qw are same regardless of the position of the qw. With UNIFORM dataset, the tuning time of GDIN is about 99%, 47%, and 3% of that of CEDI, DSI, and HCI for $WindowRatio = 0.04$ and R_{hot} of 0.8.

In Fig. 5(b), we compare the tuning time of GDIN with those of other schemes for various $WindowRatio$ values when R_{hot} is 0.8 for REAL dataset. GDIN has almost the same tuning time as CEDI but shorter than that of HCI and DSI due to the reason mentioned above. With UNIFORM dataset, the tuning time of GDIN is about 99%, 65%, and 15% of that of CEDI, DSI, and HCI for $WindowRatio = 0.1$ and R_{hot} of 0.8.

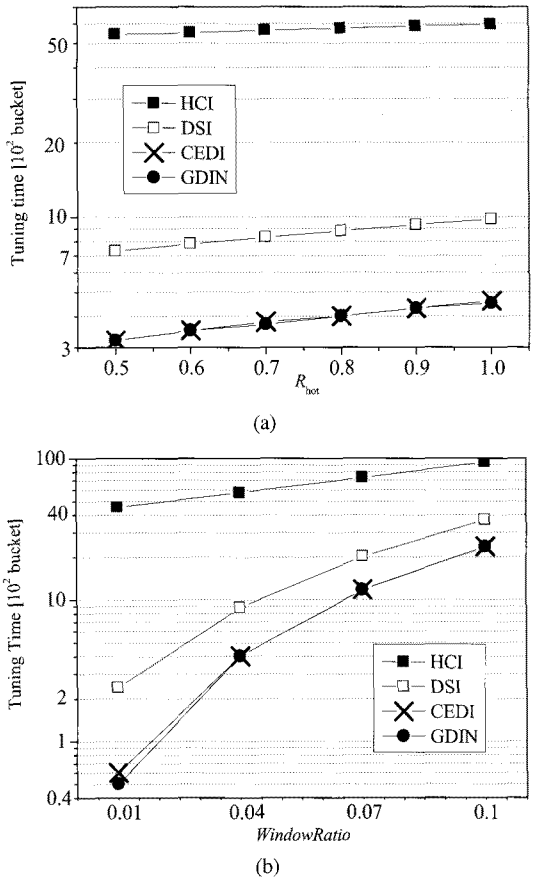


Fig. 5. The comparison of the tuning time for REAL dataset: (a) The tuning time with different R_{hot} and (b) the tuning time with different window ratio.

D. Practical Implications

In this section, we show the practicality of the proposed GDIN with the amount of the average energy, E_q, consumed by the client during processing N_q queries. E_q is computed with real energy parameters by following energy model [12]

$$E_q = \varepsilon_a T_{\text{tuning}} + \varepsilon_d (T_{\text{access}} - T_{\text{tuning}}). \tag{3}$$

Here, ε_a and ε_d are the amount of energy consumed in active and doze mode per second, respectively. T_{tuning} and T_{access} are the average tuning and access time in seconds, respectively. The first term of (3) is the amount of the average energy consumed in active mode. The second term is the amount of the average energy consumed in doze mode, where, $(T_{\text{access}} - T_{\text{tuning}})$ is the average time length for the client to stay in doze mode during processing N_q queries.

For ε_a and ε_d in (3), we take two components of the client into consideration, i.e., CPU and network interface card (NIC), operating in active mode and doze mode, respectively. Table 2 shows the value of ε_a and ε_d by CPU and NIC [10]. Thus, the client consumes 25.16 mW/sec in doze mode and 1150 mW/sec in active mode, respectively.

On the simulation condition of $WindowRatio = 0.04$ and $R_{\text{hot}} = 0.8$ with REAL dataset, the tuning time by the proposed GDIN is 397 buckets, i.e., $T_{\text{tuning}} = 0.2$ sec. Therefore, the energy consumption in active mode is 0.2 sec × 1150 mW/sec, i.e.,

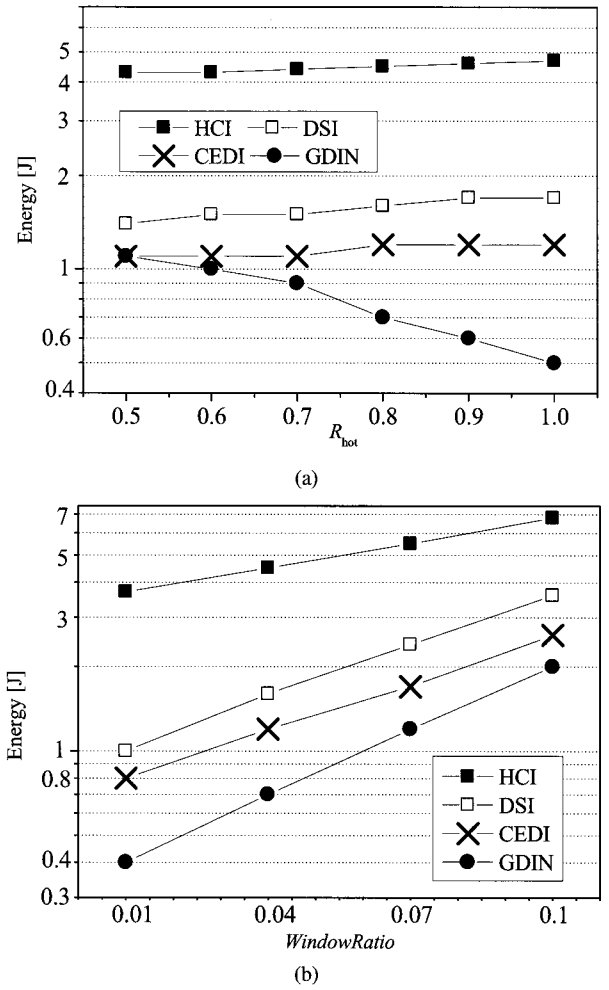


Fig. 6. The comparison of the energy consumption for REAL dataset: (a) The energy consumption with different R_{hot} and (b) the energy consumption with different window ratio.

Table 2. Energy consumption of the client (in mW/sec).

	Model	ε_d	ε_a
CPU	StrongARM SA-1100	0.16	400
NIC	RangeLAN2 7401/02	25	750

0.23 J. The access time is 37840 buckets, i.e., $T_{\text{access}} = 19.2$ sec. Therefore, the time length for the client to stay in doze mode, i.e., $(T_{\text{access}} - T_{\text{tuning}})$, is 19.2 sec − 0.2 sec = 19.0 sec. The energy consumption in doze mode is 19.0 sec × 25.16 mW/sec, i.e., 0.478 J. The total energy consumption is 0.23 J + 0.478 J, i.e., 0.708 J. On the same simulation condition and by the same computation, the energy consumptions by CEDI, DSI and HCI are 1.2 J, 1.6 J, and 4.5 J, respectively. This means that GDIN makes it possible for the client to process queries 1.63, 2.25, and 6.32 times more than by CEDI, DSI, and HCI with the same amount of energy, respectively. This shows that how practical the proposed GDIN is over the existing schemes in skewed data access pattern.

Next, we compare energy consumption of the client on various simulation conditions. Fig. 6(a) shows the amount of energy consumed by the client for REAL dataset with various R_{hot} values and $WindowRatio$ value of 0.04. It reveals that the proposed GDIN requires significantly lower energy consumption than the others in case of more skewed data access because of the shorter access and tuning time. With UNIFORM dataset, the energy consumption with GDIN is about 57%, 45%, and 14% of that with CEDI, DSI, HCI for $WindowRatio = 0.04$ and R_{hot} of 0.8. Fig. 6(b) shows the energy consumption for various $WindowRatio$ values with R_{hot} value of 0.8 for REAL dataset. The proposed GDIN outperforms the other schemes for the reason mentioned above. With UNIFORM dataset, energy consumption with GDIN is about 62%, 47%, and 21% of that of CEDI, DSI, and HCI for $WindowRatio = 0.1$ and R_{hot} of 0.8.

V. CONCLUSION

In this paper we have addressed the problem of window queries in non-flat broadcast of spatial data. We proposed a novel indexing scheme, GDIN, considering the clients' data access patterns. GDIN enables clients to access quickly the data items especially when the query of data items is skewed. It is based on $n \times n$ grid structure employing tables in two levels: row tables carrying the information on the rows, cells, and hot data items; and cell tables carrying the information on the data items in the respective cells. Performance evaluation by simulation demonstrates that the proposed GDIN significantly outperforms the existing schemes, CEDI, HCI, and DSI, in terms of access time, tuning time, and energy consumption under the skewed data access for spatial data. As for future work, we are investigating kNN search in non-flat broadcast environments.

REFERENCES

[1] T. Imielinski, S. Viswanathan, and B. R. Bardrinath, "Data on air: Organization and access," *IEEE Trans. Knowl. Data Eng.*, vol. 9, no. 3, pp. 353–372, 1997.

[2] B. Zheng, W.-C. Lee, and D. L. Lee, "Spatial queries in wireless broadcast systems," *Wireless Netw.*, vol. 10, no. 6, pp. 723–736, Dec. 2004

[3] W. Lee and B. Zheng, "DSI: A fully distributed spatial index for location-based wireless broadcast services," in *Proc. ICDCS*, 2005.

[4] S. Im, M. Song, J. Kim, S.-W. Kang, and C.-S. Hwang, "An error-resilient cell-based distributed index for location-based wireless broadcast services," in *Proc. ACM Workshop MobiDE*, June 2006, pp. 59–66.

[5] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: Data management for asymmetric communications environments," in *Proc. ACM SIGMOD*, May 1995, pp. 199–210.

[6] N. H. Vaidya and S. Hameed, "Scheduling data broadcast in asymmetric communication environments," *ACM/Baltzer Wireless Netw.*, vol. 5, no. 3, May 1999, pp 171–182.

[7] J. Xu, X. Tang, and W. Lee, "Time-critical on-demand data broadcast: Algorithms, analysis, and performance evaluation," *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 1, pp. 3–14, Jan. 2006.

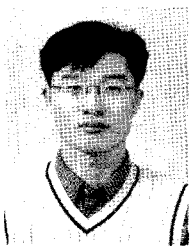
[8] Y. Yao, X. Tang, E.-P. Lim, and A. Sun, "An energy-efficient and access latency optimized indexing scheme for wireless data broadcast," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1111–1124, Aug. 2006.

[9] R. McNab and F. W. Howell, "Using java for discrete event simulation," in *Proc. UKPEW*, 1996, pp. 219–228.

[10] O.Kasten, Energyconsumption. ETHZurich, Swiss. [Online]. Available: http://www.inf.ethz.ch/personal/kasten/research/bathtub/energy_consumption

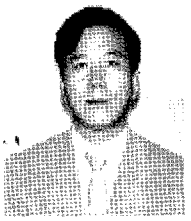
[11] MSN Direct Services. [Online]. Available: http://www.microsoft.com /industry/government/federal/doddirectband.mspx

[12] S. Im, M. Song, S.-W. Kang, J. Kim, C.-S. Hwang, S. Lee, "Energy conserving multiple data access in wireless data broadcast environments," *IEICE Trans. Commun.*, vol. E90-B, no. 9, Sept. 2007.



Seokjin Im received the B.S. degree and M.S. degree in Electronics Engineering from Kookmin University, Korea, in 1996 and 1998, respectively. He was on the Research Staff at Semiconductor Research Center of Hynix Semiconductor Corporation in Korea from 1999 to 2003. He received Ph.D. in Computer Science from Korea University, Korea, in 2007. He had been a Postdoctoral Researcher at Sungkyunkwan University, Suwon, Korea, from 2007 to 2008. He was a Visiting Scholar at California State University at Sacramento from 2008 to 2010. His major interests are ubiquitous computing, wireless data broadcast, spatial-temporal database, and mobile data processing.

Hee Yong Youn received the B.S. and M.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1977 and 1979, respectively, and the Ph.D. degree in computer engineering from the University of Massachusetts at Amherst, in 1988. From 1979 to 1984, he was on the Research Staff of Gold Star Precision Central Research Laboratories, Korea. He had been Associate Professor of Department of Computer Science and Engineering, The University of Texas at Arlington until 1999. He is presently Professor of School of Information and Communication Engineering, Sungkyunkwan University, Suwon, Korea and Director of Ubiquitous computing Technology Research Institute. His research interests include distributed and ubiquitous computing, system software and middleware, and RFID/USN.



JinTak Choi received his B.S. degree in Mathematics and his M.S. degree in Computer Science from Dongkuk University, Seoul, Korea, in 1977 and 1982, respectively. He received Ph.D. degree in Electronics from Kyunghee University, Seoul, Korea, in 1991. From 1992 to 1993, he had been a postdoctoral researcher at University of Pennsylvania. He had been a visiting scholar at California State University at Sacramento in 2005. Since 1987, he has been a Faculty Member at the Department of Computer Science of University of Incheon. His research interests include cryptography, database systems, and mobile and distributed computing.

Jinsong Ouyang received his Ph.D. degree in Computer Science from The University of New South Wales, Sydney, Australia, in 1997. From 1997 to 2001, he was doing research and development in Hewlett-Packard Company. Since 2002, he has been a Faculty Member at the Department of Computer Science of California State University Sacramento. His research interests are in the area of distributed systems focusing upon service-oriented architecture and cloud computing.