An Effective Broadcast Program Generation Algorithm for Dependent Data

Jiun-Long Huang and Wen-Chih Peng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
E-mail: jlhuang@csie.nctu.edu.tw, wcpeng@csie.nctu.edu.tw

Abstract— **We explore in this paper the problem of broadcasting dependent data in multiple broadcast channels, and explicitly investigate the effect of data replication. By analyzing the model of dependent data broadcasting, we derive several theoretical properties for the average access time in a multiple channel environment. In light of the theoretical results, we develop an efficient heuristic algorithm to generate broadcast programs, which are designed for the environments with replication. Our experimental results show that the proposed algorithm is able to obtain solution broadcast programs of very high quality.**

Keywords: Data broadcast, mobile information system, mobile computing

I. Introduction

Most works on broadcast program generation in multi-channel environments were under the premise that each user requests only one data item at a time and the requests for all data items are independent [9][13]. However, in many real applications, there exists dependency among data items. Consider a Web page containing some images as an example. Once the user requests this Web page by a browser, the browser will request these images automatically after receiving this Web page. In this example, data allocation algorithms assuming independent data are not able to effectively optimize the performance of the broadcast programs. This phenomenon attracts a series of studies on solving the problem of broadcasting dependent data. Explicitly, the problems of broadcasting dependent data can be categorized by the following three properties: (1) the number of broadcast channels (single [1][10][14] or multiple channels [5][6]), (2) the constraint of the sequence of the data items (ordered [1][6][8] or unordered[1] [2][3][4][7] queries) and (3) the employment of data replication[2] (without replication or with replication).

We address in this paper the problem of the broadcast program generation for ordered queries with dependent data in multiple broadcast channels. Different from prior work, we consider the case with data replication. Explicitly, in this paper we first model the problem of broadcast program generation for ordered queries in multiple channels with replication. By analyzing the model of dependent data broadcasting, we derive several theoretical properties for the average access time in a multiple channel environment. In light of the theoretical results, we develop a heuristic which is able to efficiently generate broadcast program on multi-channel environments with data replication. The experimental results show that the data replication technique employed can lead to more efficient use of network bandwidth. To the best of our knowledge, there is no prior work on the broadcast program generation for dependent data in multiple channels with replication. This fact distinguishes this paper from others.

The rest of this paper is organized as follows. Section II presents the preliminaries of this study. Analytical models of the problem of broadcasting dependent data with ordered queries are derived in Section III. The proposed algorithm is also described in Section III. Performance evaluation on varies parameters is conducted in Section IV. Finally, Section V concludes this paper.

II. Preliminaries

A. Problem Description

Same as in [11], it is assumed that the database D contains $|D|$ data items, $D_1, D_2, \cdots, D_{|D|}$ and each data item is read-only. In addition, each data item is assumed to be of equal size s (i.e., $|D_i| = s$ for all i). From the users' perspective, a query is an indivisible request of single or multiple data items as defined below.

Definition 1: An *ordered* query $Q_i = \{D_{q^i(1)}, D_{q^i(2)}, \cdots, D_{q^i(|Q_i|)}\}$ is an *ordered*, non-empty subset of all data items where $|Q_i|$ represents the number of required data items in Q_i. Note that $1 \leq q^i(j) \leq |D|$ for all j where $1 \leq j \leq |Q_i|$, and $q^i(j) = k$ represents that the j-th accessed data items in Q_i is D_k.
The query profile is the aggregation of the access behavior of all users. Formally, we have the following definition.

Definition 2: A query profile Q consists of a set of $\langle Q_i, Pr(Q_i)\rangle$ pairs where $|Q|$ indicates the number of queries in Q. $Pr(Q_i)$ represents the probability that a query issued by users is Q_i. It is noted that $\sum_{i=1}^{|Q|} Pr(Q_i) = 1$.

[1] An example unordered query could be one issued by a mobile user for requesting multiple data items simultaneously.

[2] A broadcast program without (respectively, with) replication corresponds to the case that data items appear with equal (respectively, different) frequencies. A broadcast program without replication is also called a *flat* broadcast program.

Query(Q_i)	$Pr(Q_i)$
$Q_1 = \{D_1, D_2, D_3\}$	30%
$Q_2 = \{D_1, D_3, D_4\}$	20%
$Q_3 = \{D_4, D_6, D_1\}$	20%
$Q_4 = \{D_1, D_2, D_5\}$	10%
$Q_5 = \{D_5, D_3, D_4, D_6\}$	10%
$Q_6 = \{D_6, D_5, D_2\}$	10%

Fig. 1. An example query profile

$D_1(1)$	$D_2(1)$	$D_1(2)$	$D_3(2)$	$D_5(1)$
$D_6(1)$	$D_3(1)$	$D_4(1)$	$D_6(2)$	$D_1(3)$

Fig. 2. An examples broadcast program with replication

We in this paper take the access time as the measurement for the quality of broadcast programs. Denote the average access time of a query Q_i as $T_{Access}(Q_i)$ and the average access time of a query profile Q as $T_{Access}(Q)$. The average access time of the query profile Q can be formulated as the following equation,

$$T_{Access}(Q) = \sum_{i=1}^{|Q|} \Big[T_{Access}(Q_i) \times Pr(Q_i) \Big]. \qquad (1)$$

III. Design of the Proposed Algorithm

Let n be the number of broadcast channels. Note that the determination of the number of broadcast channels (i.e., the value of n) is system dependent and usually involves such factors as the availability and deployment of communication channels. The problem of broadcast program generation with replication can be divided into two subproblems: (1) determining the number of replicas needed for each data item and (2) determining the placement of these replicas into the broadcast program. In this section, we first consider the determination of the number of replica for each data item in Section III-A. Then, we analyze the average access time of a broadcast program in Section III-B. Finally, we design an effective broadcast program in Section III-C according to the analytical results.

A. Determination of Number of Replicas for Each Data Item

The first subproblem indicates that the system should determine how many copies for each data item to be placed in the broadcast program. According to the property shown in [15], for data items of equal size, the total average access time will be minimized if the copies of each data item are equally spaced and for any two data items D_i and D_j,

$$\frac{n(D_i)}{n(D_j)} = \frac{\sqrt{Pr(D_i)}}{\sqrt{Pr(D_j)}}, \qquad (2)$$

where $n(D_i)$ is the number of replicas of D_i and $Pr(D_i)$ is the access frequency of D_i. $Pr(D_i)$ can be obtained by the following equation:

$$\begin{aligned} & Pr(D_i) \\ = \; & \sum_{j=1}^{|Q|} \Big(Pr(Q_j) \times \text{the no. of occurrences of } D_i \text{ in } Q_j \Big). \end{aligned}$$

Let L be the length of the broadcast program which is specified by the system designers. Since all data items should be broadcast at least once, at first each data item is assigned one slot. There will be $n \times L - |D|$ slots left. These slots are assigned to all data items according to Eq. (2). Therefore, the number of slots in the rest slots assigned to D_i (denoted as $n^{'}(D_i)$) can be calculated by the following equation,

$$n^{'}(D_i) = \frac{\sqrt{Pr(D_i)}}{\sum_{j=1}^{|D|} \sqrt{Pr(D_j)}} \times (n \times L - |D|) \, .$$

Then, $n(D_i)$ is determined as $\lceil n^{'}(D_i) \rceil + 1$.

After determining $n(D_i)$ for each data item D_i, we revise the original database by considering the number of replicas for each data item as D^* which is defined as follows,

$$D^* = \bigcup_{i=1}^{|D|} \left(\bigcup_{j=1}^{n(D_i)} \Big\{ D_i(j) \Big\} \right),$$

where $D_i(j)$ indicates the j-th copy of data item D_i. The size of the revised database D^* (denoted as $|D^*|$) is as follows,

$$|D^*| = \sum_{i=1}^{|D|} n(D_i) = n \times L.$$

After determining the number of replicas for each data item, the broadcast program with replication can be stated as follows.

Definition 3: A broadcast program P *with* replication is a *placement* of all data items in D^* into an n by L array where L is a predetermined number and $L \geq \left\lceil \frac{|D|}{n} \right\rceil$. In addition, each data item D_i will appear $n(D_i)$ times in the broadcast program P.

To ease the following discussion, we utilize the function $f(i,j)$ to represent the placement of the the j-th copy of D_i in the broadcast program. Figure 2 shows an example broadcast program with replication. In this example, the value of $f(3,2)$ is equal to 4 which indicates that the broadcast order of the second copy of D_3 is 4. Let the function $DIST(i,j,k,l)$ represent the distance between the j-th copy of D_i and the l-th copy of D_k in a broadcast program with replication. $DIST(i,j,k,l)$ is defined as follows.

$$\begin{aligned} & DIST(i,j,k,l) \\ = \; & \begin{cases} 0, & \text{if } i = k; \\ f(k,l) - f(i,j) - 1, & \text{if } i \neq k \text{ and} \\ & f(k,l) > f(i,j); \\ L - f(i,j) + f(k,l) - 1, & \text{if } i \neq k \text{ and} \\ & f(k,l) \leq f(i,j). \end{cases} \end{aligned}$$

B. Analytical Model

The access time of an arbitrary query Q_i can be decomposed into three parts: *startup time*, *waiting time* and *retrieval time*. When a mobile user submits a query Q_i, the mobile device should wait until the system starts to broadcast the first required data item of Q_i (i.e., $D_{q^i(1)}$). This time interval is called the startup time. The waiting time is defined as the summation of the time intervals between the moment that the mobile device completes the retrieval of the data item $D_{q^i(j)}$ and the moment that mobile device starts to retrieve the next data item $D_{q^i(j+1)}$. The retrieval time is the aggregated time while the mobile device indeed reads data items from broadcast channels. It is noted that the retrieval time of a query is proportion to the number of data items required by the query.

By the definition of the average access time, the average access time of the query Q_i (denoted as $T_{Access}(Q_i)$) can be formulated as

$$T_{Access}(Q_i) = T_{Startup}(Q_i) + T_{Waiting}(Q_i) + T_{Retr.}(Q_i).$$

It is obvious that the average access time of a broadcast program is equal to the summation of the average startup, waiting and retrieval time of a broadcast program. Now, we discuss the derivations of these three components in the following subsections.

1) The Derivation of Average Startup Time: We first derive the average startup time of a broadcast program ($T_{Startup}(Q)$). Without loss of generality, we assume that the user submits a query on the m-th broadcast cycle. We also assume that the offset between the start time of the m-th broadcast cycle and the moment of the mobile user submits a query is a uniform distribution over $(0, L)$. To simplify the further derivation, we define a series $a(j)$, $j = 1, 2, \cdots, n(D_{q^i(1)})$, to represent the *sorted* placements of all copies of $D_{q^i(1)}$, and b to represent $n(D_{q^i(1)})$. Then, we have the following lemma. Therefore, $T_{Startup}(Q_i)$ can be formulated as follows.

Lemma 1: In a broadcast program with replication, the average startup time of Q_i can be formulated as

$$\begin{aligned} T_{Startup}(Q_i) \quad = \quad & \frac{s}{L \times B} \times \left\{ \sum_{i=1}^{b-1} \frac{\Big[a(i+1) - a(i)\Big]^2}{2} + \right. \\ & \left. \frac{\Big[L - a(b) + a(1)\Big]^2}{2} \right\}. \end{aligned}$$

2) The Derivation of Average Waiting Time: To facilitate the derivation of the average waiting time of a query ($T_{Waiting}(Q_i)$), we first define $p^i(j)$ to be the probability that the user retrieves the j-th copy of $D_{q^i(1)}$ as the first required data item of Q_i. Thus, we have the following lemma. Due to the limitation of space, the proofs of all lemmas are omitted.

Lemma 2: In a broadcast program with replication, $p^i(j)$ can be formulated as follows:

$$p^i(j) = \begin{cases} \frac{L - a(b) + a(1)}{L}, & \text{if } j = 1, \\ \frac{a(j) - a(j-1)}{L}, & \text{otherwise.} \end{cases}$$

To facilitate the calculation of $T_{Waiting}(Q_i)$, table Distance Table (abbreviated as table DT) for the broadcast program is constructed in advance. In table DT, $DT(i, j, k).ID = m$ indicates that the distance from the j-th copy of D_i to the m-th copy of D_k is the shortest among all copies of D_k, and let $DT(i, j, k).Length$ be the shortest distance among the distances from the j-th copy of D_i and to all copies of D_k. Therefore, $DT(i, j, k).ID$ and $DT(i, j, k).Length$ can be formulated as below.

$$DT(i, j, k).ID = m,$$

$$DT(i, j, k).Length = DIST(i, j, k, m), \text{ where}$$

$$DIST(i, j, k, m) = \min_{l=1,2,\cdots,n(D_k)} \Big\{ DIST(i, j, k, l) \Big\}.$$

Finally, $T_{Waiting}(Q_i)$ can be obtained by table DT and $p^i(j)$. For a query Q_i, since $p^i(j)$ is dependent of the allocation of the replicas of $D_{q^i(1)}$, we have to minimize $DT(i, j, k, m)$ when minimizing $T_{Waiting}(Q_i)$.

3) The Derivation of Average Retrieval Time: Since the retrieval time is the aggregation of the time when the mobile device indeed reads data items from broadcast channels, the corresponding average retrieval time of the query Q_i, $T_{Retr.}(Q_i)$, is

$$T_{Retr.}(Q_i) = |Q_i| \times \frac{s}{B}. \tag{3}$$

C. Broadcast Program Generation

According to the analytical results shown in Section III-B, we have the following observations. First, we observe from Lemma 1 that all replicas of D_{q^i} (i.e., $a(i)$s) should be placed with equal space in order to minimize $T_{Startup}(Q_i)$. Then, for each query Q_i, we have to allocate $D_{q^i(j)}$, where $j = 2, 3, \cdots, n(D_i)$, as close to replicas of $D_{q^i(j-1)}$ as possible in order to minimize $T_{Waiting}(Q_i)$.

We then design a broadcast program generation algorithm according to the above observations, and the proposed algorithm is as follows. We first calculate the number of replicas for each data item according to the method proposed in Section III-A. Then, we select all queries according to their access probabilities in descending order. After selecting a query Q_i, we check whether $D_{q^i(1)}$ has been allocated. If not, we try to allocate all replicas of $D_{q^i(1)}$ with equal space, and mark $D_{q^i(1)}$ as *allocated*. We allocate $D_{q^i(j)}$, where $j = 2, 3, \cdots, |Q_i|$ sequentially. Similarly, we skip $D_{q^i(j)}$ when it has been allocated. Otherwise, we allocate all replicas of $D_{q^i(j)}$ as close to the replicas of $D_{q^i(j-1)}$ as possible. Finally, we select next query until all queries have been selected.

IV. Performance Evaluation

For performance studies, we implement the proposed algorithm, called scheme Replication, and a query profile generator based on the approach mentioned in [12]. Since L is a user specified parameter, we define the replication factor as $\frac{L \times n}{|D|} =$

TABLE I

SYSTEM PARAMETERS

Parameters	Values
The size of each data item (s)	8K bytes
The bandwidth of each channel (B)	80K byte/sec.
The number of data items ($\|D\|$)	450
The number of channels (n)	6
The replication factor $\left(\frac{\|D^*\|}{\|D\|}\right)$	2
The number of queries ($\|Q\|$)	300
The value of fanout	15
The average query length	15
The Zipf distribution (θ)	0.5

$\frac{|D^*|}{|D|}$ to represent the degree of replication. The probability of the query Q_i issued by users is assumed to be

$$Pr(Q_i) = \frac{(\frac{1}{i})^\theta}{\sum_{j=1}^{n}(\frac{1}{j})^\theta}$$

where θ is the parameter of the Zipf distribution. Table I shows the system parameters in our experiments. In addition to scheme Replication, we also implement scheme GA-NoRep for comparison purposes. GA-NoRep is a GA-based algorithm to generate broadcast program for ordered queries in a multiple channel environment without replication [5]. The simulator and query profile generator are coded in C++.

Figure 3 shows the experimental results with the number of broadcast channels varied. We observe that the average access times of all schemes decrease as the number of channels increases. This agrees with the intuition that the increase of bandwidth will decrease the average access time. However, for all schemes, the improvement in average access time decreases as the number of broadcast channels increases. As a result, the determination of the number of broadcast channels should consider the balance between performance gain and the number of channels used. The number of broadcast channels suggested by this experiment is around 7. Scheme Replication leads to better broadcast programs than scheme GA-NoRep even when scheme GA-NoRep uses more broadcast channels. This result shows that the higher bandwidth utilization can be attained by employing data replication.

V. CONCLUSION

We explored in this paper the problem of broadcasting dependent data in multiple broadcast channels, and explicitly investigated the effect of data replication. By analyzing the model of dependent data broadcasting, we derived several theoretical properties for the average access time in a multiple channel environment. In light of the theoretical results, we developed scheme Replication to generate broadcast programs, which were designed respectively for the environments with replication. Our experimental results showed that the data replication technique employed can lead to more efficient use of network bandwidth.

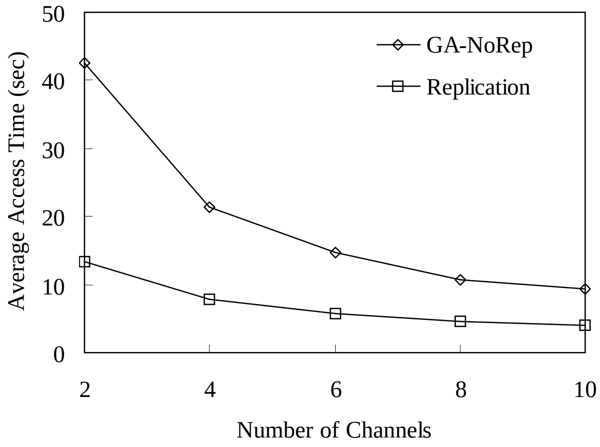

Fig. 3. The effect of the number of channels

REFERENCES

[1] Y. C. Chehadeh, A. R. Hurson, and M. Kavehrad. Object Organization on a Single Broadcast Channel in the Mobile Computing Environment. *Multimedia Tools and Applications*, 9(1):69–94, July 1999.

[2] Y. D. Chung and M. H. Kim. Effective Data Placement for Wireless Broadcast. *Distributed and Parallel Databases*, 9(2):133–150, March 2001.

[3] J.-L. Huang and M.-S. Chen. Broadcast Program Generation for Unordered Queries with Data Replication. In *Proceedings of the 8th ACM Symposium on Applied Computing*, March 2003.

[4] J.-L. Huang and M.-S. Chen. Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment . *Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment*, 16(9):1143–1156, September 2004.

[5] J.-L. Huang, M.-S. Chen, and W.-C. Peng. Broadcast Dependent Data for Ordered Queries without Replication in a Multi-Channel Mobile Environment. In *Proceedings of the 19th IEEE International Conference on Data Engineering*, March 2003.

[6] A. R. Hurson, Y. C. Chehadeh, and J. Hannan. Object Organization on Parallel Broadcast Channels in a Global Information Sharing Environment. In *Proceedings of the 19th IEEE International Performance, Computing, and Communications Conference*, pages 347–353, February 2000.

[7] G. Lee and S.-C. Lo. Broadcast Data Allocation for Efficient Access of Multiple Data Items in Mobile Environments. *ACM/Kluwer Mobile Networks and Applications*, 8(4):365–375, August 2003.

[8] G. Lee, S.-C. Lo, and A.L.P. Chen. Data Allocation on the Wireless Broadcast Channel for Efficient Query Processing. *IEEE Transactions on Computers*, 51(10), October 2002.

[9] W.-C. Lee, Q. L. Hu, and D. L. Lee. A Study on Channel Allocation for Data Dissemination in Mobile Computing Environments. *ACM/Kluwer Mobile Networks and Applications*, 4(5):117–129, May 1999.

[10] V. Liberatore. Multicast Scheduling for List Requests. In *Proceedings of IEEE INFOCOM Conference*, June 2002.

[11] C.-W. Lin, H. Hu, and D. L. Lee. Adaptive Realtime Bandwidth Allocation for Wireless Data Delivery. *ACM/Kluwer Wireless Networks*, 10:103–120, 2004.

[12] A. Nanopoulos, D. Katsaros, and Y. Manolopoulos. Effective Prediction of Web-user Accesses: A Data Mining Approach. In *Proceedings of the WEBKDD Workshop*, August 2001.

[13] W.-C. Peng and M.-S. Chen. Efficient Channel Allocation Tree Generation for Data Broadcasting in a Mobile Computing Environment. *ACM/Kluwer Wireless Networks*, 9(2):117–129, 2003.

[14] A. Si and H. V. Leong. Query Optimization for Broadcast Database. *Data and Knowledge Engineering*, 29(3):351–380, March 1999.

[15] J. W. Wong. Broadcast Delivery. *Proceedings of IEEE*, 76(12):1566–1577, December 1988.