Ramamohanarao Kotagiri
P. Radha Krishna
Mukesh Mohania
Ekawit Nantajeewarawat (Eds.)

LNCS 4443

Advances in Databases: Concepts, Systems and Applications

12th International Conference on Database Systems for Advanced Applications, DASFAA 2007
Bangkok, Thailand, April 2007, Proceedings

Springer

Proc. of Int'l Conf. on Database Systems for Advanced Applications (DASFAA 2007), pp. 884-890 (Apr. 2007).

On Query Processing Considering Energy Consumption for Broadcast Database Systems

Shinya Kitajima[1], Jing Cai[1], Tsutomu Terada[2], Takahiro Hara[1], and Shojiro Nishio[1]

[1] Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{kitajima.shinya, cai, hara, nishio}@ist.osaka-u.ac.jp
[2] Cybercommunity Division, Cybermedia Center, Osaka University
5-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
tsutomu@cmc.osaka-u.ac.jp

Abstract. In recent years, there has been an increasing interest in the broadcast database system where the server periodically broadcasts contents in a database to mobile clients such as portable computers and PDAs. There are three query processing methods in the broadcast database system, while each method consumes different amount of power for query processing. In this paper, we propose a new query processing method which dynamically chooses an appropriate method among the three query processing methods by considering energy consumption.

Keywords: data broadcast, broadcast database system, query processing, power consumption.

1 Introduction

The recent evolution of wireless communication technologies has led to an increasing interest in broadcast information systems in which data is disseminated via the broadcast. In such systems, a server broadcasts various data periodically via the broadband channel, while clients pick out and store necessary data. There are many studies for improving the performance of broadcast information systems[1,2,3]. Most of them deal with broadcast data as data items simply, and do not address the performance improvement by considering contents and characteristics of broadcast data.

In this paper, we assume a broadcast system that the server periodically broadcasts contents in a database and clients issue queries to retrieve data from the database. We call such a system *broadcast database system.* There are three basic query processing methods in this system. However, the performance of each method changes according to the system situation such as query frequency.

In [5], we proposed a query processing method which chooses the method with the least response time among these three methods. In this paper, based on the method in [5], we propose a new query processing method which considers the energy consumption of mobile clients when choosing a query processing method.

R. Kotagiri et al. (Eds.): DASFAA 2007, LNCS 4443, pp. 884–890, 2007.

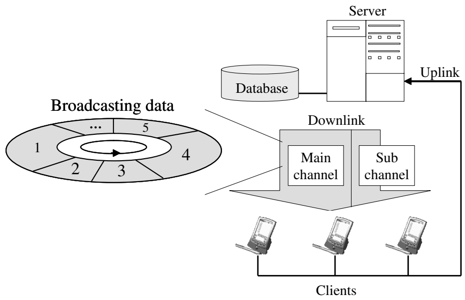

Fig. 1. Broadcast database system

Furthermore, the simulation evaluation confirms that the proposed method improves the lifetimes of clients with a low battery power left.

The remainder of this paper is organized as follows. Section 2 describes the outline of a broadcast database system and introduces three basic query processing methods and the traditional method in the broadcast database system. Section 3 explains our method in details. Section 4 evaluates the performance of our method. Finally, we conclude the paper in Section 5.

2 Broadcast Database System

Fig. 1 illustrates the concept of a broadcast database system. In this system, the server broadcasts contents in a relational database via the broadcast channel and processes queries from clients. Clients issue queries to retrieve the necessary data from the database. Clients have a small storage, low power resource, and low CPU capability, such as a PDA. The broadcast channel from the server to clients is divided into two channels: a broadband *main channel* to disseminate the contents in a database repeatedly, and a narrowband *sub channel* to disseminate the other data. Moreover, there is a narrowband uplink channel from clients to the server. Clients use the uplink channel to send queries to the server.

2.1 Assumed Environment

We assume that our method is used for disseminating information to many and unspecified users. For example, for an information service in a shopping center, the server broadcasts data in the database including advertising information, shop information, and goods information in the shopping center, while thousands of users receive the broadcast information and retrieve the necessary information. The broadcast data contains maps of shops and images of goods.

The users occasionally issue queries to the server to retrieve the information, such as a natural join operation "I want the image of item A, and the map to the shop selling the item". We assume several minutes' delay for receiving query result is acceptable for clients. On the other hand, users set the deadline of the response time to each query. When users cannot receive query result by the time of deadline, the query fails. Although users stay in shopping center for a certain time, the battery is limited and the users who run out of the battery cannot receive the service.

2.2 Query Processing Methods

In the broadcast database system, there are three basic query processing methods and one adaptive method as follows.

On-demand method: A client sends a query to the server through the uplink. The server processes the query and broadcasts the query result via the sub channel. In this method, the query processing is completely done by the server, no workspace is required for query processing at client, thus the energy consumption is low.

Client method: A client stores all the tables related to the query, and processes the query by itself. Query processing causes a heavy workload on the client which consumes a lot of battery power.

Collaborative method: A client sends a query to the server through the uplink. The server processes the query, attaches the query identifier to the tuples that appear in the query result, creates rules for the client to process the data, and then broadcasts the rules via the sub channel. Based on the received rules, the client receives the necessary tuples via the main channel referring to the identifiers, and reconstructs the query result by combining these tuples[4].

LRT (Least Response Time) method: The system performance, when each method is used individually, changes with the environmental conditions such as query frequency. In the LRT method[5], when the server receives a query, it calculates the response time respectively for the on-demand method, the client method, and the collaborative method, and then chooses a query processing method with the least response time.

3 ELEC Method

In the LRT method, there is a problem that the lifetimes of clients with a low battery become short. Therefore, we propose a new query processing method, called ELEC (Extended LRT considering Energy Consumption) method, that considers the battery capability of mobile clients when choosing a query processing method, and improves the lifetimes of clients which remain a low battery power.

3.1 Outline

In the ELEC method, the server basically chooses a query processing method according to the LRT method. However, the server checks the remaining battery of the client issuing query, and preferentially chooses the method with the lowest energy consumption when the remaining battery is lower than the threshold P_{TH}. The procedure of the query processing is as follows.

1. If the remaining battery of the client which currently issues a query is more than P_{TH}, the server chooses the query processing method according to the LRT method.
2. Otherwise, the server chooses the query processing method with lowest energy consumption.

The optimal P_{TH} will be changed according to the query frequency, remaining battery of clients, and so on.

3.2 Calculation of the Thresholds

For deciding the optimal threshold P_{TH}, the server duplicates last q queries while changing the tentative value of the threshold. Then, the server sets the threshold, so that the energy consumption for clients of which the remaining battery is less than the threshold, becomes least.

1. We define the remaining battery of the client which issues the last x-th query Q_x as P_x. We also define the sequence of numbers which consists of q elements as $A = (P_1, P_2, \ldots, P_q)$, and the ascending sequence of A as the sequence of nominated thresholds A'.
2. We represent the k-th element of A' as $A'[k]$. The server takes the following steps for $A'[i](i = 1, 2, \ldots, q)$.
   (2a) The server duplicates last q queries by setting $A'[i]$ as tentative threshold.
   (2b) The server calculates the energy consumption per query of which the remaining battery is less than $A'[i]$, which is represented as E_i.
   (2c) If $E_j > E_{j-1} > E_{j-2}(j = 1, 2, \ldots, q)$, the server judges the optimal threshold is less than A_j, and goes to step 3.
3. We represent the minimum of $E_i(i = 1, 2, \ldots, q)$ as E_{min_i}, while the threshold for next q queries as $A'[min\_i]$.

4 Evaluation

This section evaluates the ELEC method. Three evaluation criteria are introduced as follows.

Lifetime: The elapsed time from the user arrival to the user exit, due to either spending enough time or running out of the battery. Note that clients consume battery only by the processes related to query processing.

Success rate: The ratio of the queries of which clients could get the results to all of the queries clients issued.

Response time: The average elapsed time from the query generation to the acquirement of the query result. Note that the response time does not include the time for transmitting a query from a client to the server and the time for processing the data at the client side or the server side, since they are adequately short.

Table 1. Parameters

Parameter	Value
Simulation time[sec]	36000
Estimated staying time of users[sec]	7200
Query interval for a user[sec]	300
Deadline of response time[sec]	80
Number of tuples	10000
Size of a tuple[KByte]	10
Number of identifiers	200
Bandwidth of main/sub channel[Mbps]	10/1
Size of a processing rule[KByte]	1
Average ratio for all tuples of necessary tuples	0.003
Standard deviation for all tuples of necessary tuples	0.001
The initial remaining battery of each client[unit energy]	100 - 1000(uniform distribution)
The storage capacity of each client[MB]	1 - 100(uniform distribution)
The speed of writing/reading data[MB/s]	10/15

4.1 Simulation Environment

In the evaluation, the database schema and the query model is supposed for an information service in a shopping center as described in Section 2.1. The database consists of a shop table and a goods table. For the sake of simplicity, all tuples are supposed to be the same size. Moreover, a client only issues queries of a natural join with the shop table and the goods table.

Table 1 shows the parameters used in the evaluation. The rate of necessary tuples represents the rate of tuples which are included in the query result to all tuples in the table. The user arrival intervals are given by the exponential distribution with a parameter of user arrival frequency, which is changed as shown in Fig. 2. Each user has a mobile client to access service. A user, who arrives at the shopping center, issues a query according to the query interval, with the deadline of the response time, the storage capacity, and the remaining battery. Each user exits the shopping center after elapsing the estimated staying time. We assume that the CPU capability of all clients are the same. Moreover, Mahesri et al's measurement results in [6] are used to define that the energy consumption of query processing is proportional to time t, it is presented as total sum of the energy consumption for CPU ($E_C^h = 2t$ (when CPU load is high), $E_C^l = 0.5t$ (when CPU load is low)), the energy consumption for wireless LAN ($E_W = t$), and the energy consumption for I/O ($E_I = t$ for reading, $E_O = t$ for writing).

4.2 Simulation Results

Table 2 shows the evaluation results of the success rate and the average response time of the LRT method and the ELEC method. Moreover, Fig. 3 shows the evaluation results of the average lifetime under different initial remaining battery of clients, which is classified in every 100 unit energies. The parameter q of the ELEC method is set to 50 according to the preliminary experiment.

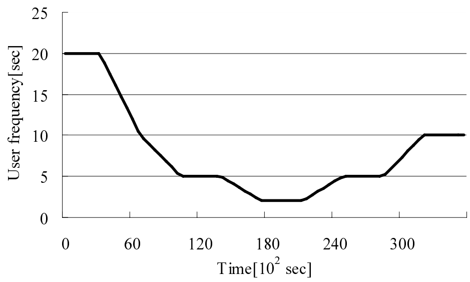

Fig. 2. Change of user arrival frequency

Table 2. Success rate and response time

Method	Success rate	Average response time
LRT	94.4%	40.4[sec]
ELEC	93.5%	41.9[sec]

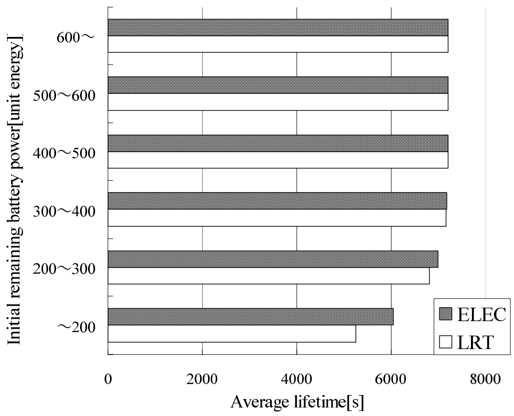

Fig. 3. Average lifetime

Table 2 and Fig. 3 show that the average lifetime of the ELEC method is much longer than that of the LRT method, though the success rate and the average response time of the ELEC method are a little worse than those of the LRT method. In the ELEC method, the lifetime of clients which have a low remaining battery is long, since the server preferentially chooses the method that has lowest energy consumption when the remaining battery of the client is low. However, the restriction of the sub channel, storage, and identifier worsens the success rate and the average response time, due to continuously choosing the query processing method with the lowest energy consumption.

5 Conclusions

In this paper, we proposed a new query processing method which dynamically chooses a query processing method by considering the energy consumption. With the proposed method, the server preferentially chooses the method with the lowest energy consumption when the remaining battery is lower than the threshold. The simulation results confirmed that the proposed method improved the lifetime of the clients which have a low remaining battery.

In future, we plan to examine a method which decides the threshold according to the CPU capability of clients.

Acknowledgments. This research was partially supported by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and Grant-in-Aid for Scientific Research (A)(17200006) and (18049050) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

1. Acharya, S., Alonso, R., Franklin, M., and Zdonik, S.: Broadcast disks: data management for asymmetric communication environments. Proc. ACM SIGMOD'95, pp. 199–210, May 1995.
2. Acharya, S., Franklin, M., and Zdonik, S.: Balancing push and pull for data broadcast. Proc. ACM SIGMOD'97, pp. 183–194, May 1997.
3. Aksoy, D., Franklin, M., and Zdonik, S.: Data staging for on-demand broadcast. Proc. VLDB'01, pp. 571–580, Sept. 2001.
4. Kashita, M., Terada, T., Hara, T., Tsukamoto, M., and Nishio, S.: A collaborative query processing method for a database broadcasting system. Proc. CIIT'02, pp. 60–66, Nov. 2002.
5. Kitajima, S., Cai, J., Terada, T., Hara, T., and Nishio, S.: A query processing mechanism based on the broadcast queue for broadcast database systems. Proc. ISWPC'06, pp. 450–455, Jan. 2006.
6. Mahesri, A., and Vardhan, V.: Power consumption breakdown on modern laptop. Proc. International Workshop on Power-Aware Computing Systems (PACS'04), Dec. 2004.