Dynamic Broadcast Scheduling in DDBMS

Babu Santhalingam[#1], C.Gunasekar[#2], K.Jayakumar[#3]

[#1] Asst. Professor, Computer Science and Applications Department, SCSVMV University, Kanchipuram, India,

[#2]Research scholar, Computer Science and Applications Department, SCSVMV University, Kanchipuram, India,

[#3]Research scholar, Computer Science and Applications Department, SCSVMV University, Kanchipuram, India,

Abstract

This system will address the problem of online scheduling sequential data objects with the principle of periodicity in the context of dynamic information dissemination. Many modern information applications spread dynamically generated data objects and answer the complex query for retrieving multiple data objects. In dynamic environments, data streams need to be online processed rather than being stored and later retrieved to answer queries. Particularly, data objects are produced dynamically by the information providers, interleaved and disseminated efficiently by the broadcasting servers, and associated sequentially in the client sides. The proposed algorithm with a well-specific gain measure function prominently outperforms the FIFO schedule and is able to minimize the mean service access time to the extent close to the theoretical optimum.

Keywords: Push, broadcast, periodicity, scheduling, query processing, and data dissemination.

1. Introduction

Many emerging data-intensive applications involve a number of servers and a much larger number of clients, e.g., electronic auction and tender, information news distribution, proxy service, and Web surfing, to name a few. The traditional pull-based or client-server data dissemination model usually suffers from the scalability problem, especially in the asymmetric communication networks where the downward information flow or bandwidth capacity is larger than the upward one. Being scalable against extensive client population, among others is the push-based data broadcast model as investigated by many researches [2], [3], [4], [5]. Specifically, the central idea of data broadcasting is that the server periodically broadcasts data objects to many clients who monitor the broadcast channel and retrieve data objects without explicit requests. Substantial researches have explored the data broadcast methodologies but are mainly based on the traditional data management systems [4], [6], wherein a data object is independent, persistent, and static and is mapped to a pair of state and value in the database against a simple query/transaction [2], [5]. A simple query (or a read transaction) is processed successfully, while a client retrieves the answered data object on the broadcast channel. In practice, however, many modern information applications spread dynamically generated data objects and answer the complex query for retrieving multiple data objects. In dynamic environments, data streams need to be online processed rather than being stored and later retrieved to answer queries.

1.1 Related Works

In the existing system [1], Periodic schedule is NP-hard for information broadcasting. The data is broadcasted based on the Transaction Id. The invalidation and the multiversion techniques are presented to increase the concurrency of read-only transaction in the presence of an update. The broadcasting of disks are prefetched and cached in the system.

In the proposed system, information is broadcasted dynamically. The data is not broadcasted based on the Transaction Id. Data broadcasting such that the Admin periodically broadcasts data objects to many employees who monitor the broadcast channel and retrieve data objects without explicit requests along with the employees request if any. Dynamic Information dissemination when the broadcasting server is dealing with unrelated data.

2. System Modelling

Fig. 1 depicts the clients-providers-servers system where we formulate the problem of scheduling sequential objects. Let $S = \{ S_1; S_2; \ldots ; S_M\}$ denote the collection of M information service providers. Each S_k produces n_k uniform-length objects within a constant time interval I, i.e., object
Productionrate n_k. The broadcasting P server has $N = 1 \leq k \leq M\ n_k$ objects in an I. Assume that a slotted time model is employed: it takes one time slot to broadcast each object. The length of a broadcast cycle, denoted as L, is equal to the number of time slots used to deliver objects by a periodic broadcast schedule, denoted as P.

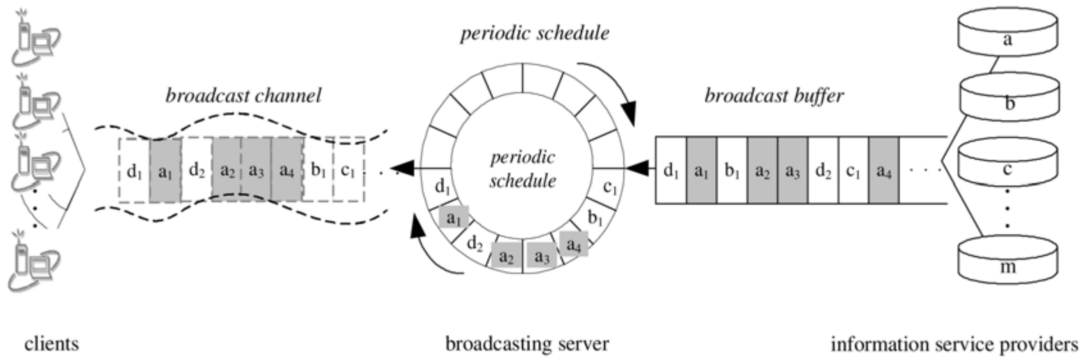

Fig. 1. Dynamic data dissemination in a clients-providers-servers system.

3. Modules

3.1 Client Module

This module gives a set of requests to the server, which responds to the request but always processed and responded by the scheduled optimizer.

3.1.1 Pseudo code

Place the request

While (server on-line) forward Request to Server

End While

3.2 Broadcasting server

The Server broadcasts the information been sent and received and consists of a queue follows Request Response Model.

3.1.2 Pseudo code

While (on Request) Process Request

Call Schedule Optimizer

Generate Response

End While

3.3 Schedule optimizer

The Schedule optimizer is embedded inside with the broadcast server which decides the schedules of sending and receiving the data. The algorithm used in this module is Deterministic Scheduling algorithm.

3.1.3 Pseudo code

While (on call)

Evaluate Bandwidth Evaluate Type of Request

Evaluate State of Database Schedule Broadcast Return result to Server

End While

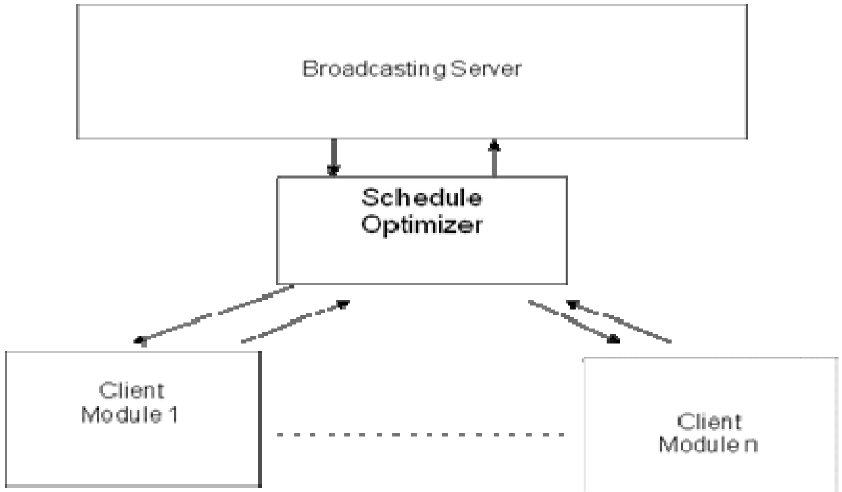

Fig.2 Functional Block Diagram

4. Online Scheduling With Optimal Approximation

In light of the above quantitative analyses and results, this section develops a deterministic algorithm for the broadcast schedule optimization and the assessment of mean service access.

4.1 FIFO Strategy

Let an infinite input sequence Pin arrange data objects in chronological order. Given an input queue Q (to tolerate traffic variations), by the FIFO strategy, the broadcasting server always selects the " head object" in Q as the next broadcast object into the output sequence Pout. Thus, the slot position distribution in Pout is the same as that in Pin, and the object sequential order ok of each Sk is preserved in Pout.

4.2 Deterministic Strategy

A deterministic strategy is to process directed distance adjustment and deterministic object selection. While scheduling dynamic sequential objects, the broadcasting server adjusts the directed distance of two consecutive objects from the same information service. For the arrival of an object $d^k{}_i$, the server will decide whether it is profitable to broadcast $d^k{}_i$ directly at the moment or defer its broadcast and keep it in the output buffer. In this way, the server will adjust the distance dd^k_{i-1}, i from the preceding object d^k_{i-1} to $d^k{}_i$. On the other hand, while objects in buffer are considered, an effective object selection process is required to determine the next broadcast object in each time slot.

4.3 Gain Measure Function

In our design, four gain measure functions, GMF-1, GMF-2, GMF-3, and GMF-4, are presented for the deterministic object selection as follows:

$G1(d^k_i) = -q_k . n_k$

$G2(d^k_i) = q_k.L - q_k (dd^k_{i+1,i+2} + ... + dd^k_{nk,1} + ... + dd^k_{i-1,i})$,

$G3(d^k_i) = q_k . (L/n_k - dd^k_i\text{-}1,i)^2 / (L/n_k)^2$

$G4(d^k_i) = G3(d^k_i) * \sum_{1\leq i\leq nk-1}(dd^k_{i,i+1} - L/n_{k-} (L/n_k - dd^k_{i-1,i}))$.

GMF-1 and GMF-2 are responsive to the theoretical spirit of broadcast periodicity. The factors n_k and L are primary ones for computing gain in the deterministic procedure. In GMF-3, the normalized $| L/n_k - dd^k_i\text{-}1,i |$ reflects the potential gain of a candidate object distinctly, regardless of other n_k -1 objects in the schedule. GMF-4 further takes account of its n_k -1 neighbouring objects temporally.

4.4 Deterministic Strategy against Web Content Traffic

This section examines deterministic strategies against practical patterns of Web traffic workload. The understanding of Web traffic characteristics [7], [8], [9] suggests that 1) the use of a Zipf-like distribution with its θ in the range of 0.75 to 0.90 traces out a trend toward less variability in popularity across files on Web servers [8], [10] and [3]) the Pareto distribution with its α value in the range of 1 to 1.5 can fit well with the distributions of file and transfer (or response) sizes and the number of embedded references among Web pages [9], [11]. In conformity to the proposition of scheduling sequential objects, we assume that the delivery of each Web object contains b broadcast items of embedded reference objects or file segments with a smaller and uniform-length item size. Accordingly, the simulation takes θ=0.85 and α=1.1 to generate data sets of Web workload. Fig. 3 illustrates the performance results against Web data workload in the cases of b in the range of 2 to 10. The deterministic strategies manifest their functions to reduce T(t). All their effects can be obviously improved by the increment of buffer size. Particularly, DSMF-1 and DSMF-2 approach to the theoretical lower bound by ≤ 0.1 and ≤ 0.035 normalized differences after the buffer queue is enlarged to contain 60 percent and 80 percent of the data workload.

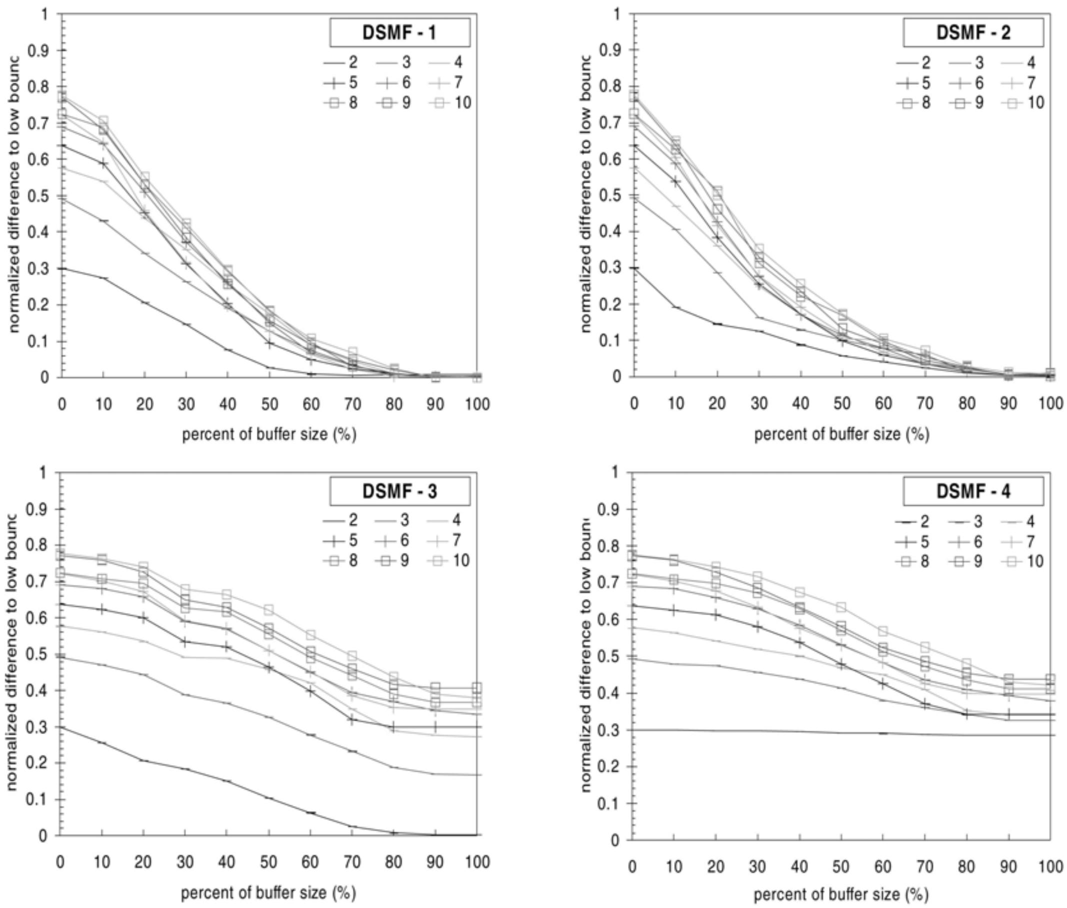

Fig. 3. Deterministic strategies against Web content traffic with α=1.1 and θ=0.85 (baseline: M=100, b=2; 3; . . . ;10, and B=0;10; . . . ;100 percent).

DSMF-3 can attain similar performance when the object production rate is smaller. Its T(t), however, goes higher than DSMF-1's and DSMF-2's T(t)s when the object production rate increases. Relatively, DSMF-4 is worse than the others in that its T(t)declines slowly. It can be explained that the occasion of slow decline is attributed to the accumulated distance of normalized difference.

Conclusion and Future Work

The paper addressed the problem of online scheduling sequential data objects with the principle of periodicity in the context of dynamic information dissemination. We have devised an efficient deterministic strategy and examined several gain measure functions for dynamic generation of a periodic broadcast schedule. The result of simulation have shown that the proposed algorithm with a well-specific gain measure function prominently outperforms the FIFO schedule and is able to minimize the mean service access time to the extent close to the theoretical optimum. The implementation can be applied for broadcasting and can far look into all types of online processing issues which involve higher bandwidth. This can also be added with the modules of transaction where data persistence is also taken into consideration.

References

[1] "Online Scheduling Sequential Objects with Periodicity for Dynamic Information Dissemination", Chih-Lin Hu and Ming-Syan Chen, "IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 21, NO. 2, FEBRUARY 2009"

[2] S. Acharya et al., "Broadcast Disks: Data Management for Asymmetric Communications Environments," Proc. ACM SIGMOD '95, pp. 199-210, May 1995.

[3] M.H. Ammar and J.W. Wong, "On the Optimality of Cyclic Transmission in Teletext Systems," IEEE Trans. Comm., vol. 35, no. 11, pp. 1159-1170, 1987.

[4] D. Barbara´, "Mobile Computing and Database-A Survey," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 108-117, Jan./ Feb. 1999.

[5] T. Imielinski and B.R. Badrinath, "Mobile Wireless Computing: Challenges in Data Management," Comm. ACM, vol. 37, no. 10, pp. 18-28, 1994.

[6] K.-L. Tan and B.C. Ooi, Data Dissemination in Wireless Computing Environments. Kluwer Academic Publishers, 2000.

[7] M. Arlitt and C. Williamson, "Internet Web Servers: Workload Characterization and Performance Implications," IEEE/ACM Trans. Networking, vol. 5, no. 5, pp. 631-645, 1997.

[8] B. Krishnamurthy and J. Rexford, Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffice Measurement. Addison-Wesley, 2001.

[9] X. Tang, J. Xu, and S.T. Chanson, Web Content Delivery. Springer, 2005.

[10] L. Breslau et al., "Web Caching and Zipf-Like Distributions: Evidence and Implications," Proc. IEEE INFOCOM '99, pp. 126-134, Mar. 1999.

[11] P. Barford and M. Crovella, "Generating Representative Web Workloads for Network and Server Performance Evaluations," ACM SIGMETRICS Performance Evaluation Rev., vol. 26, no. 1, pp. 151-160, 1998.