

EFFICIENT ALGORITHM FOR DISSEMINATING CONSISTENT DATA TO READ-WRITE MOBILE CLIENTS

Dr.K.Chitra Manikandan¹, S.Abirami²

¹Asst.Prof.,Computer Science,
Govt. Arts & Science college for women,
Pudukottai, Tamilnadu.

²Lecturer in BCA, Dept. of Mathematics,
Thiagarajar college, Theppakulam,
Madurai-9, Tamilnadu.

ABSTRACT

Most of emerging mobile data base applications emphasis on data dissemination. Data dissemination is the delivery of data from a server to large number of mobile clients. While the data on server being broadcasted to mobile clients , update transactions from external sources may be executed concurrently. If the execution of update and broadcast transactions were overlapped, then mobile clients may observe inconsistent data. This paper proposes an index based algorithm for ensuring the consistency of read-write mobile clients. Under this algorithm mobile clients can receive consistent and current data and it allows the mobile clients to update broadcasted data.

KEYWORDS

Dynamic Database, Data Dissemination, Consistency , Serializability, push.

1. INTRODUCTION

The enormous growth in wireless communication technology give the mobile users the capacity of accessing information at anywhere . Many emerging mobile database applications involve the data dissemination to larger clients[13][15]. Examples of such dissemination based applications include stock information system, sport tickers and Ecommerce solutions. The two methods of data disseminations are push based ,pull based data disseminations[2][3].With the push based data dissemination(Data broadcast method) the server broadcasts the data in regular intervals without the request of mobile clients. In pull based data dissemination method (Client server architecture) the server broadcasts data to the mobile clients on demand.

In asymmetric communication environment downstream bandwidth capacity from server to client is higher than upstream communication from client to server. Pull based methods are not suitable for asymmetric communication environment[1][14].For Example wireless class rooms, hospital information system. In such environment the mobile client has some characteristics[4]: Frequent disconnection, Power limitation. There are many benefits of the broadcasting mode[5] .It reduces

bandwidth by allowing the clients to receive information without communicating the server. It provides faster response and it saves the battery power.

Since many data item in mobile computing system are used to record real time information system. E.g., the current traffic condition of the road, news updates,. Their values are highly dynamic .If the mobile client can't get the most updated data then accessing outdated data is undesirable. When concurrent execution of update and broadcast transactions are allowed , to maintain the freshness of the data items, the problem of data consistency must be addressed.

In this paper we propose a new serializability based concurrency control algorithm to ensure the consistency and currency for Read-Write mobile clients. The following summarizes the contributions of this paper .

- 1.The algorithm broadcasts the consistent data to read write mobile clients.
2. Size of control information is smaller.
- 3.The algorithm broadcasts the same ordered set of updated items to all the mobile clients.
4. It broadcasts the data items along with status information for newly updated data items.
5. It ensures the consistency over multiple broadcast cycles.
6. The algorithm broadcasts the current and consistent data to the mobile clients who were all connected to the network at any time.

The organization of remaining part of the paper is as follows. Section 2 describes the system model. Section 3 reviews related work on consistent data delivery. Section 4 discuss the data inconsistency problem. New algorithm is proposed in section 5.Conclusion and future work of this paper is in section 6.

2. ARCHIRECTURAL MODEL

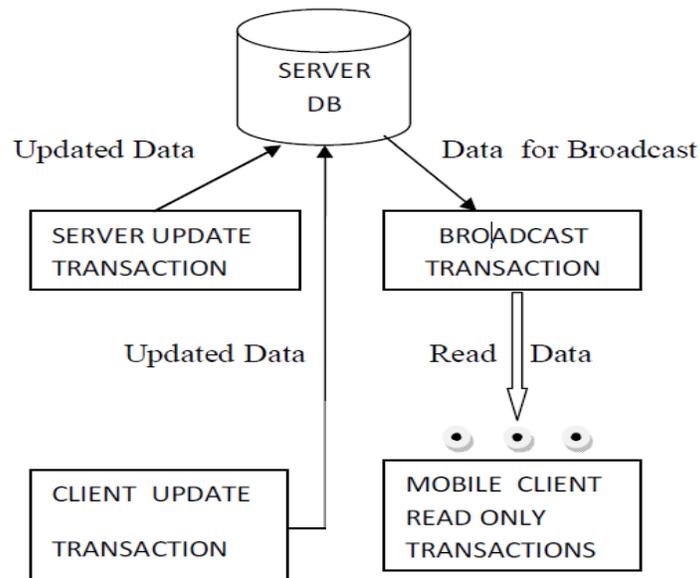


Figure 1: Our System Model

As seen in Figure(1), our model consists of four transactions: the Broadcast transactions, Mobile read only transactions, Mobile update transactions and server update transaction. A mobile computing system consists of an information server and large number of mobile clients connected via wireless network. Information server maintains the highly sensitive data items. For example last traded price of stock ,location of moving object, etc., To maintain the validity of data items server update transactions must be installed immediately.

In our architecture model, it is assumed that update transactions(UT) are short and they are generated either from external sources or from mobile clients. The information server periodically broadcasts the data items .Each period of data broadcasting is called as broad cast cycle. The length of broadcast cycle can be fixed or variable. The broadcasts process of data items in multiple cycles is modeled as broadcast transactions(BT).

Mobile clients could be able to issue read only and update transactions to the server. A mobile read only transaction is assumed to access one data item from the broadcast list. Mobile update transaction is assumed to update one fresh data item from the current broadcasted data items.

2.1 Assumptions

The following summarizes the assumptions of this paper,

- All update transactions processed only on information server.
- Consistency among the update transactions(server, client) resolved through 2PL algorithm.
- A mobile transaction can issue either read only or update request.
- Each data item is stored on data base server with row index.
- All mobile clients can get the same ordered set of updated data items.
- For all update date transactions , the proposed algorithm broadcasts the updated data items with the status information on current cycle itself.
- A mobile update transaction can update only the currently broadcasted data items.
- Authenticated mobile clients alone can issue update transactions.

3. RELATED WORK

In [6] they suggested OUFO algorithm for data consistency of read only mobile clients with minimum delay. Data conflicts between update and broadcast transactions is resolved by comparing the set of data to be broadcasted with the set of updated data items. This algorithm re broadcasted the overlapped data items. To reduce the waiting time of mobile clients, set of broadcasted data items are compared with data item set requested by the mobile client. If there is a match then mobile client read the data item. Otherwise the broadcasted data items may be in already read set. Then the mobile transaction is re started . To reduce restart overhead OUFO is suggested with catching and IR methods. But this approach suffer from the overhead of generation of IR.

In [7] they proposed BUC algorithm for inconsistency problem of read write mobile transactions. This approach modeled with three component Is(Information server),BRS(broadcasts server),Mu(Mobile unit) . BRS receives current broadcast cycle list of (index , updated cycle) for all scheduled data, original data from IS. To ensure the consistency mobile clients are allowed to read the data item only if the span value of each read set data item is lesser than or equal to span

window. Otherwise mobile transaction is re started. It maintains the consistency by allowing the update transactions of mobile clients only if $update\ cycle_{IS}$ is equivalent to $update\ cycle_x$. This algorithm is simple. But it suffers from excessive transmission of broadcast cycles.

In [8] they proposed broadcast disk environment where all updates executed at server, clients access them in a read only fashion. They enforce the consistency through IR Invalidation scheme and Caching mechanism. For cache management LIX algorithm was proposed. Whenever the server page is updated, server broadcasts IR and propagation List. Then client removes the affected page from the cache specified by IR. But this approach would require client and server to communicate for every read.

The broadcast concurrency control with forward validation (BCC_FV) mechanism for data consistency is proposed in [9]. At the server side data conflict is detected by comparing Write set and Read set of active transaction. There is only one type of data conflict between committed update transaction and the active transaction: Write-read conflict. If so, read only transaction is aborted. But there is an overhead of excessive transaction restart. They proposed another protocol called BCC_TI. It assigns time stamps of every update transaction. During each broadcast cycle, server broadcasts write set along with time stamps. The mobile read only transaction is allowed if Lower bound timestamp of read only transaction \geq time stamp of update transaction. Otherwise it is aborted. Here the mobile client can't get the same order of updated data.

In [10] they proposed F-Matrix & R-Matrix algorithms. For each broadcast cycle it broadcasts control matrix ($n*n$) along with the data items. Where n is the number of database items. Mobile client has to check the data conflicts before reading the data from broadcast cycle. To ensure that no live transaction writes on to any data in read set. Otherwise client has to restart its transaction. It allows the clients to update the broadcasted data items. Due to the broadcast of control matrix it increases the bandwidth requirement.

In [11] whenever the data conflict occurred, update transactions are aborted on server. To broadcast the consistent data, broadcast transaction reads the entire database (Long lived BRT). While the broadcast transaction is being processed at server, then there can be two types of update transaction. First the update transaction serialized before broadcast. Second is the update transaction serialized after broadcast. The second type of update transaction need to be aborted for the broadcast of fresh data items. But it is not suitable for dynamic databases.

Multi version data broadcast method in [12] increases the consistency and it provides the clients to access the multiple server states. It broadcasts the multiple versions of each data item using broadcast disk organization. In multiple clustering approach all versions of each data item broadcasted sequentially. With the over flow approach older versions of data items stored separately and they are broadcasted at the end of cycle. These approaches are useful at frequently disconnection environment. But they require high bandwidth.

In [16] we ensure consistency through the usage of Indexing mechanism with serializability criteria. Whenever the update transaction is executed after broadcast transaction data conflict may occur. IOBS algorithm detects and resolve the data inconsistency by rebroadcasting the newly updated data with status information as New. The algorithm broadcasts the ordered updated read set with status information to all mobile clients. This algorithm preserves consistency only for read only mobile clients.

4. INCONSISTENT DATA DELIVERY

Since the execution of broadcast transaction is performed concurrently with update transaction, mobile clients may observe inconsistent data. In this paper we adopt serializability as the consistency criterion.. Serializability is the global property for consistent database system. If the serialization graph is acyclic the schedule is serializable.

We have list out some conditions[6] where the inconsistency may occur. we have assumed that A,B and C are data items.

1.The server update transactions executed after broadcast transaction. i.e., if the schedule is (BT1A,UT1A,UT2B,BT2B)

2.Data conflicts due to the more update transactions.
(BT1A,UT1A,B,UT2B,UT3C,A,BTC)

3.If the mobile transaction may not be able to find all the required data item in a single broadcast cycle.
(BT1A,UT1A,BT2B,UT1A,UT2B,BT3A)

4.The data conflicts from mobile update transaction requests.

The data inconsistency on first condition is resolved by rebroadcasting the updated data items immediately on the same cycle. Second and third conditions are resolved by rebroadcasting the updated data items for every update transaction. The fourth condition is resolved separately on server using index searching method.

5. CONSISTENCY ALGORITHM

Data inconsistency can be identified when we compare update transactions with broadcast transactions. The basic principle of this algorithm is to ensure the data consistency whenever the above conditions (Sec 4) met. This algorithm provides guarantee for the broadcast of ordered updated data items to all the mobile clients, who were all tuned into the channel.

5.1 Details of the Consistency algorithm

The server execute this algorithm indefinitely along with the broadcast scheduling algorithm. The server broadcast the data items in regular intervals . Such interval is called as broadcast cycle. The server has to maintain the consistency of data for two types of update transactions. One is update transactions of server, second is update transactions of mobile clients. To implement the above requirements this algorithm designed with two phases: server update phase, client update phase.

5.2 Server update phase

The server update phase divided into three processes (Commit, Updation , Broadcast).

5.2.1 Updation process

During the execution of an update transaction ,this process blocks broadcast transaction. In order to deliver the same ordered set of updated data items it keeps the updated data items and their indices

on UPD , UPI for every update transaction.

5.2.2 Commit phase

This process is executed after the completion of updating process for every update transaction. It resolves the data conflicts by comparing the index set of updated data(UPI) with the index of data to be broadcasted(BRI).If there is any index of UPI is in index set of BRI then it removes index from the BRI to avoid duplicated broadcast and it rebroadcasts the updated data items, Index , new status information. Otherwise updated data items may be broadcasted in previous cycles , this phase rebroadcasts the list of information (newly updated value, index, status information).

5.2.3 Broadcast process

If there is no update transaction request to the data base, this phase broadcast the data item, index from the current broadcast list specified by broadcast scheduling algorithm.

5.3 Mobile update Phase

While the data being broadcasted to mobile clients, this process allows the clients to read the data item, status information. If the mobile client needs to update the broadcasted data item then this phase sends Index ,Old value, Updated data item to the server for commitment. This phase compares last committed data base item with Old value sent by the client. If they are same, client updation is executed. Otherwise client updation is rolled back. After the execution of this phase, server update phase would be executed automatically. We have assumed that 2PL algorithm is used to resolve the data consistency among update transactions submitted by multiple clients.

5.4 Algorithm

In this section we will state the consistency algorithm formally. First we define the following quantities.

BRD=Set of data items to be read by broadcast transaction.

BRI=Index of data items to be read by broadcast transaction.

UPD=set of updated data items of an update transaction.

UPI=set of index of updated data items of an update transaction.

SERVER ALGORITHM

```
Ser(fl=0, id, old, ud ){
    If(fl==1)
        SER2(id, old, ud);
    While (true){
        If (UT is in execution)
        { Block BRT;
          UPI+= Ordered Index of updated data items;
          UPD+= Ordered Updated data items;
        }
        else if (UPT is committed)
        {
            for each (X in UPI)
            {
                If (X e BRI)
                { //serialization order is UPT →BRT.
                  Remove data item from BRD where row index=X;
                  Broadcast data item ,X, status =New from UPD where row
                    index = X;}
                else
                { // X may already broadcasted in previous cycles.
                  Rebroadcast data item X, status= New from UPD where row
                    index =X;
                }
            }
        } //for
        UPI={ }, UPD={ }
    } //if
    else
    { //No update request
      Broadcast data item, Id from the current broadcast list.
    }
} // End while
```

SER2 Algorithm

```
SER2(id, old, ud)
{
    Get data 'p' from the data base where row_id=id;
    If(p==old) //Data to be updated is fresh
    {Update p=ud where row_id=id; Commit; }
    Else
    {No Updation;}
}
```

The SER2 algorithm resolves data inconsistency for client update transactions. If the flag(fl==1) then SER algorithm performs the process for client update transaction. If flag(fl=0) it performs the process for server update transaction from external sources. Data conflicts between the server update transaction and broadcast transaction is detected during commit process. Rebroadcast is used to resolve the data conflicts and to maintain the serializabilities order. UT→BRT.

When the update transaction of mobile client submitted to server, freshness of the data item which is to be modified is verified on mobile client update phase. If it is the fresh data then updation is allowed. After the execution of updation server update phase would detect the data inconsistency. Otherwise updation is not allowed on server. This algorithm doesn't provide the mechanism from sending updation result to mobile client.

5.5. Client side Algorithm

The server broadcasts data item, index and status information to the mobile clients . This algorithm filter the data item, status information for the mobile clients. If a mobile client is decided to update the broadcasted data item 'D' with new value then the algorithm sends old data item, index and new data item to the server for the commitment.

```
Client ( )
{
Read data item ' D' index, status form the broadcast.
Filter data item 'D', status for the mobile client.
If ( D is to be updated)
{
Old =D;
D= new value;
Call ser (1, index, old, D);
}}
```

5.6 Properties and overheads of the algorithm

5.6.1 Properties

The important properties of the algorithm are:

1. It broadcasts ordered read set to all read write mobile clients.
2. The impact of this algorithm on server is minimum.

The implementation of this algorithm is simple. All data conflict checks are performed on server. The client can access the broadcasted data without contacting server.

5.6.2 Overheads

We note that the only overhead of this algorithm are:

1. Data conflicts checks using index set of UPT and BRT.
2. The rebroadcast of conflicting data items

3.The BRT is blocked until the UPT is completed.

To speed up the client checking process , index comparison is used. The overhead of conflict checks using index is lower than conflict checks with data items. We have assumed that the length of update transaction is short. so the waiting time of broadcast transaction is minimum.

5.6.3 Disconnection

The mobile network is frequently disconnected voluntary or involuntary. The impact of disconnection on this algorithm is minimum. Because this algorithm broadcasts the data item with status information for all mobile clients. i.e., Mobile clients can get the new version of data items with the status information.

6. CONCLUSION AND FUTURE WORK

This algorithm efficiently broadcasts the current and consistent data to the mobile clients. It broadcasts the ordered updated data to all mobiles .This type of ordered delivery is use full for dynamic data bases. Broadcasting the data along with the status information is useful for the mobile clients who were tuned in to the channel currently. Since the indexing method is used ,the proposed algorithm is more suitable for broadcasting large size of data items (video, Image). Even though the algorithm retains the consistency data it has some limitations,

- 1.Detection of data conflict is complicated.
- 2.Whenever the rate of update transactions is increased then the rebroadcast rate will be increased.

REFERENCES

- [1] Swarup Acharya, Rafael Alonso, Michael Franklin and Stanley Zdonik, 1996," Broadcast Disks: Data Management for Asymmetric Communication Environments", The Kluwer International Series in Engineering and Computer Science, Volume 353, PP 331-361.
- [2] Swarup Acharya, Michael Franklin, Stanley Zdonik., 1997," Balancing Push and Pull for Data Broadcast", Proceedings of the 1997 ACM SIGMOD international conference on Management of data , Volume 26 Issue 2.
- [3] Pavan Deolasee, Amol Katkar Ankur Panchbudhe, Krithi Ramamritham, Prashant Shenoy, 2001,"Adaptive Push-Pull: Disseminating Dynamic Web Data", Proceedings of the 10th international conference on World Wide Web:ACM 2001, PP 265 – 274.
- [4] Daniel Barbará,1999," Mobile Computing and Database-- A Survey", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING" , VOL.11, NO. 1.
- [5] T. Imielinski, S. Viswanathan, and B.R. Badrinath, Member, IEEE,1997," Data on Air: Organization and Access", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 9, NO. 3.

- [6] Kam-Yiu Lam, Edward Chan, Hei-Wing Leung and Mei-Wai Au,2004, "Concurrency Control Strategies for Ordered Data Broadcast in Mobile Computing Systems", . Proceedings of International Conference on Management of Data (COMAD'2000), Vol. 29.
- [7] Ahmad S. Al-mogren, Margaret H. Dunham,2001," BUC, a Simple yet Efficient Concurrency Control technique for Mobile Data Broadcast Environment", Proceedings of the 12th International Workshop on Database and Expert Systems Applications-DEXA Workshops , pp. 564-569.
- [8] Swarup Acharya , Michael Franklin, Stanley Zdonik,1996," Disseminating Updates on Broadcast Disks", Proceedings of the 22nd VLDB Conference -Mumbai(Bombay), India.
- [9] Victor C. S. Lee and Kwok-Wa Lam Sang H. Son.,1999," Maintaining Data Consistency Using Timestamp Ordering in Real-Time Broadcast Environments", Proceedings of the Sixth International Conference on Real-Time Computing Systems and Applications ,pp 29-36.
- [10] Shanmugasundaram, J. Nithrakashyap, A., Sivasankaran .R. and Ramamritham, K.,1999," Efficient Concurrency Control for Broadcast Environments", Proceedings of ACM SIGMOD International.,1999,Volume 28 Issue 2.
- [11] Victor C. S. Lee, Kwok-wa Lam, Simon Wu, Eddie Chan,2001," Broadcasting Consistent Data in Mobile Computing Environments", Proceedings of the seventh Real Time Technology and Applications Symposium,IEEE computer Society Washington.
- [12] Evaggelia Pitoura, Panos K. Chrysanthis,2002," Multiversion Data Broadcast", IEEE Transactions on Computers, vol. 51, no. 10, pp. 1224-1230.
- [13] R.Varadarajan, M.Chithra, 2009,"Scheduling data reads and writes using feedback on a weakly connected environment" ,International journal of Computer Applications in Technology vol.34,no:3.
- [14] R.Varadarajan , M.Chithra, 2009,"Reducing latency through efficient channel allocation methods for multiversion data dissemination in mobile computing environment", International journal of Internet Technology and Secured transactions, vol.1, nos.3/4.
- [15] S. Krishna Mohan Rao¹, Dr A. Venugopal Reddy², Data Dissemination in Mobile Computing Environment, BVICAM'S International Journal of Information Technology, January-July, Vol.1 No.1,2009.
- [16] Dr.K.Chitra, Dr.K.Senthamarai Kannan, S.Abirami,, Index Based Ordered Broadcast with Status (IOBS) Algorithm for Consistent Data Broadcast, proceedings of First International Conference, CCSEIT 2011- Springer, Tirunelveli, Tamil Nadu, India, September 23-25, 2011, Communications in computer and Information Series , Vol. 204,1st Edition 2011, pp 275-285.