International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Power and Time Aware Replicated Consistent Data Dissemination to the Read-Write Mobile Clients (PTRCD)

Dr. S. Abirami[1], Dr. K. Chitra[2]

[1]Assistant Professor Department of CA & IT, Thiagarajar College (Autonomous), Madurai, Tamil Nadu, India
[2]Assistant Professor Department of Computer Science, Govt. Arts College, Melur, Madurai, Tamil Nadu, India

ABSTRACT

Though the mobile technology has tremendous growth, the limitation of mobile devices and mobile network such as limited battery power, data traffic and network disconnections are still the major research issues in mobile computing system. The author propose an enhanced index based broadcast scheme called PTRCD (Power and Time aware Replicated Consistent Data Dissemination) by adapting caching with compression mechanism for disseminating consistent replicated mobile data to the read-write mobile clients through reduced power, time consumption of mobile clients with minimum bandwidth requirement. The proposed approach is implemented with replication technique for providing high data availability and accessibility. The experimental results prove that the performance of suggested approach is better in most cases. The key features of PTRCD system are : (1) It caches only the required data of mobile clients and the index of first data item of current broadcast cycle in compressed form to reduce the tuning time, power consumption and network communication (2) Cached data are invalidated by re-caching the re-broadcasted data items from broadcast channel with no upstream communication.

Keywords : Mobile Caching, Data Compression, Replicated Database, Consistency, Data Dissemination.

I. INTRODUCTION

The technology development in mobile computing system increases the mobile users day by day. Most of the web applications are migrated to mobile applications. Especially the e-commerce applications need to reach out their data to the wide range of mobile clients. Large number of E-Commerce applications rely on broadcast based data delivery [1][2], in which the mobile database server broadcasts the database content in series of broadcast cycles to the massive amount of mobile clients. In these environment, the bandwidth of wireless channel is limited than that of wired channel as well as the up stream communication bandwidth from client to the server is very lower than down stream communication [3][4]. Whenever the server broadcast the data, the mobile clients have to receive consistent data with in their limited resources. The mobile clients may receive inconsistent data due to the following reasons: Uncontrolled execution of broadcast and update transaction, network disconnections due to the scarcity of power back up, delayed response because of data traffic and node failure in mobile network. Receiving outdated (not a fresh) data is not worth-able for the mobile clients in most cases.

Data broadcasting approach provides more benefits but the mobile clients have to wait for their interested data in broadcast channel [5]. Caching server data at mobile clients provide good solution for data broadcast system because it reduces the power consumption, communication between clients and server [6][11] but preserving consistency on cached data is an additional burden. Reducing power consumption, data access delay and bandwidth communication guarantee the consistent retrieval of data with in the dead line of mobile client transaction. The replication in broadcast system provides the high data availability and the mobile clients can get their data items independent of their location through minimum data access delay.

In active replication, the mobile clients can get the updates from any replica servers. Passive replication allows the mobile clients to get updates only from the master server. The master server can propagate the entire data updates to all replicas (content transfer) or it may propagate the description about update transaction (log transfer) [9]. Whenever the server database is updated, the updates can be propagated to all replicas immediately (active propagation). The Lazy propagation scheme propagates the updates after the commitment of update transaction. Though the replication provides high performance, enforcing consistency in replicated mobile database environment is complicated. Due to reduce the bandwidth requirement and to enforce the maximum consistency and currency, the Single master-active replication with log based lazy propagation technique is suggested in PTRCD approach.

Data broadcast system for time constrained mobile clients can be achieved by reducing data access delay and power consumption of mobile clients. In general mobile clients can operate in three modes [10], active mode (mobile device send and receive signals), doze mode (mobile device can receive signals, process of these devices at low rate), sleep mode (communication is suspended). By allowing the mobile clients to operate in doze mode, power consumption can be highly reduced. In this paper, the author propose a cache and index based algorithm for broadcasting consistent data to the read-write mobile clients in replicated mobile database system.

Most of the existing cache based mobile data dissemination approaches cache the data on some active mobile clients, they cache the data depend on access frequency and also on-demand data broadcast scheme is applied for cache miss. Though the existing approaches provide maximum cache hit rate, caching consistent data on mobile clients from special kinds of database through minimum size of control information and network communication are not considered. The following section-2 describes the related work and their limitations. Section-3 presents our cache based power and time aware system model. The different phases of the proposed algorithm are described in section-4. Conclusion and future enhancements specified in section -5.

II. Related Work

Cache management architecture is implemented using three protocols in [12] to save the power consumption of mobile clients and to simplify cache management. The proposed cache replacement algorithm caches the server data on some clients based on weight vector. Bandwidth communication, cpu speed, access delay, hit ratio are used to calculate the weight vector. Based on weight vector and threshold value, the server data items would be cached into some active clients. The second algorithm (cache discovery) is addressed to discover the requested item from the catch. Whenever the data item is requested by the mobile clients, it will be retrieved either from the local cache (if it is available) or from the active mobile client by broadcasting the data requests to the active clients. Mobile clients can get the updated data through time stamp mechanism. Whenever the cache is full, the cache replacement algorithm replaces the cached data based on access rate, distance or size of data. Though these algorithms reduce the bandwidth communication, they could not provide consisted data broadcast of dynamic database.

To increase the cache hit, Absolute Value Interval (AVI) method [13] is proposed based on access frequency. This algorithm has two phases. The first is for cache invalidation and the second phase performs the broadcast skipping process. The proposed AVI method increases the cache hit of highly dynamic databases using temporal properties of data (Access frequency, AVI). Here, each data item would be grouped based on their update rate. Each data item is assigned with time frame (which includes data value, AVI value, update time stamp). Data would be chosen for broadcast depends on time frame. In static AVI approach, broadcast bandwidth is fixed and it is not so in dynamic. The duplicated broadcasting has been avoided in the broadcast skipping method. Since this protocol is not in need of invalidation messages, it saves network bandwidth. But the consideration of temporal properties alone is not efficient in all cases.

To reduce the rate of cache miss and to save the power consumption of mobile clients, the proposed algorithm in [14] allows the clients to wake up only during IR period. This algorithm broadcast the ID list of data would be broadcasted in next cycle then broadcast the data in the ID list. It follows the stateless approach. To satisfy the time constrained mobile transactions, the server allows the priority requests. These types of requests would be broadcasted immediately instead of waiting for next IR. The modified UIR method is proposed to maintain the consistency on dynamic databases. UIR (updated IR) contains the updated data after the last IR was broadcasted. It suggests to broadcasts the hot data items on one channel and broadcasting the cold data along with UIR using another channel. Still these approaches retain the consistency with the minimum impact on mobile clients but they are in need of additional communication bandwidth for broadcasting IR and UIR.

Extended OUFO is proposed in [15] to preserve consistency and to reduce the access delay of mobile clients. The requested data items on the cache of mobile clients are matched with the broadcasted data items. If so then mobile transaction can read that data. When the server broadcast the already read data items whenever the mobile client waiting for another data then the mobile transaction would be re-broadcasted. It increases the response time and restart overhead. To avoid these, invalidation with OUFO is proposed. Here, cached data is validated by comparing the time stamp of accessed data with that of in IR. When the timestamp of IR is greater than time stamp of the data item on cache then the mobile transaction has to be restarted. This approach ever delivers the most updated data items to the mobile clients but broadcasting IR and frequent rebroadcast may occupy more network bandwidth.

Consistent data dissemination for time constrained mobile transaction is addressed in [16]. The proposed extended multi version broadcasting (MV) algorithms reduce the data access time. Though the MV broadcasting approach is suitable for frequently disconnected mobile clients, it increases the length of broadcast transaction. The first proposed algorithm enforces the consistency by allowing smaller broadcast cycles. It provides increased data currency but it is not suitable for larger databases. The second performs the cache invalidation by comparing the broadcast cycle number of first version of data with the cycle number of latest version. Whenever the client transaction has received the requested data equal to the cycle number on the cache, then the mobile transaction would be committed but they utilize more network bandwidth. The two other schemes LVO and AR are proposed to reduce the broadcast overhead. These proposed schemes are fit for time constrained mobile clients but they are not suitable for databases of larger data items.
Two algorithms proposed in [17] reduce the number of dead line missing transactions. Request proposition method (RP) calculates the access scores of data items for a transaction with multiple data request based on EWMA (Exponentially Weighted Averaging). It is calculated by considering number of data items requested for a transactions and the number of transactions that are accessed the data item. So it reduces the number of deadline missing transactions. EQSD (Equal Stack Data items) broadcast the data depend upon their deadline. Closest deadline data item values will get higher priority for broadcast. Data deadline is calculated with the following properties, data arrival, dead line time of a transaction and the number of data items required by a transaction. Still it serves data delivery to the time aware transactions there is no logic for data consistency.

On-demand based cache replacement algorithm MIN-SAUD (Minimum Stretch Integrated with Access rate, Update frequencies, cache validation Delay) is proposed in [18]. It allows the mobile clients to issue their requests on uplink channel and the server broadcast it through broadcast channel. Then broadcasted data items are compared with the cached data item. If there is a catch hit, validity of cached data will be verified by receiving next IR. If there is a catch miss or invalid, pull request will be given to server. Whenever the cache is full the proposed cache replacement algorithm ejects the data based on the probability of lower access rate, data arrival delay, and update frequency, larger size of data and the cost of invalidation. This scheme reduces the delay in executing mobile client's transaction but this type of on demand broadcast mechanisms are in need of additional communication cost.

III. System Model

The proposed PTRCD approach is aimed to meet the following,

- Consistent & ordered data dissemination to the read-write mobile clients
- To provide high data availability and accessibility
- Save power and time consumption of mobile clients in data dissemination
- Reduce communication bandwidth between servers and mobile clients
- To handle node failure in replicated database environment.

To provide high data availability and quick response, the proposed algorithm employs on replication technique. In PTRCD system, the master server is responsible for executing update transactions (either from external environment or from mobile clients) then it propagates update log transaction to all replica servers. Replica servers focus on broadcasting consistent data. On client side, caching mechanism is used to reduce power consumption, network communication. The time constrained mobile transaction requests would be executed by the replica servers instantly. The following Figure.1 shows the proposed system model.

In PTRCD, consistency is ensured among MS (Master Server) and RS (Replica Server) through synchronization mechanism and the consistency between replica servers and the mobile clients (MC) is achieved through serialization and queuing mechanisms. The proposed approach suggests the cache size as one half of the size of a broadcast cycle. The mobile clients can specify their data of interest (DI) on cache. Our algorithm caches the newly updated data on mobile clients in compressed form with no upstream communication.

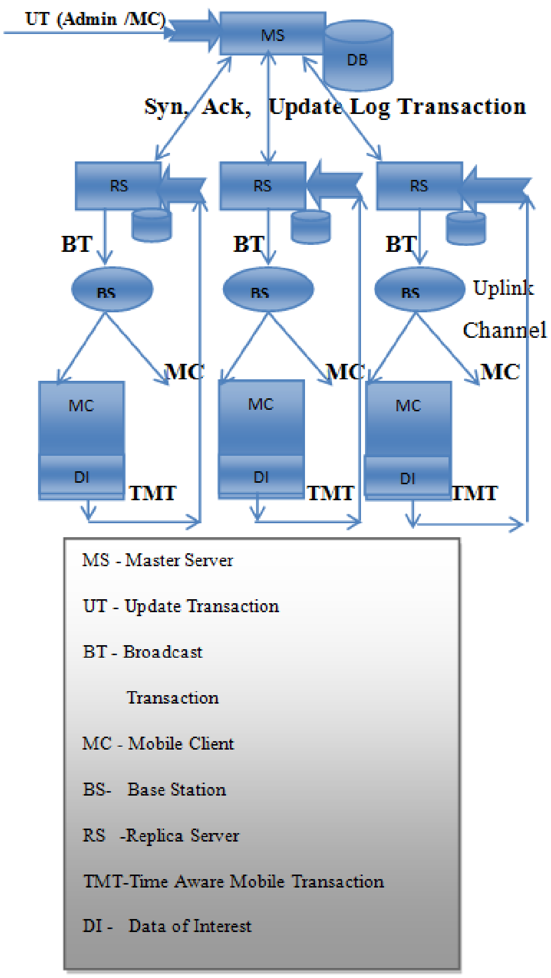

Figure 1. PTRD System model

3.1 Properties of the proposed Algorithm

The PTRCD approach is designed with the following properties,

- To save the tuning time and power consumption of mobile clients, we cache the data request of individual mobile client by let them in doze mode
- Whenever the data items are updated on MS, they will be blocked from broadcast transaction to preserve consistency
- By re-caching the updated data items immediately, the mobile clients can receive current data through lower communication bandwidth
- MS guarantees the consistency among update transactions using queuing mechanism
- All the RSs broadcast consistent data in the same order to the mobile clients using Syn (Synchronization) and Ack (Acknowledgement) signals
- In PTRCD approach, time aware mobile transactions are satisfied by allowing the mobile clients to issue their data request on uplink channel. Then the replica servers would broadcast the requested data immediately
- To reduce broadcast transaction length, we eliminate duplicated data rebroadcast by using serialization criterion

IV. Algorithm

The PTRCD approach emphasis on replication technique with caching and compression mechanism to disseminate consistent data through minimum power consumption. To satisfy the time constrained mobile transactions, the proposed system allows the mobile clients to issue the on-demand requests to the replica servers. We have designed our algorithm with four phases,

1. Update Master server phase
2. Broadcast Replica server phase
3. Mobile clients phase
4. Time aware mobile Transaction phase.

A) Update Master Server Phase

This phase is responsible for executing update requests either from external environment or from mobile clients and it has to propagate the updates to all replica servers (RS) instantly. In PTRCD, the database stored on MS can be temporal, non-temporal or dynamic database. Based on the above criteria, we have designed this phase with two sub phases.

Case 1: Read-only mobile clients

In this case, the master server is updated only by the administrator and it may have temporal, non-temporal or dynamic data on database. To avoid preemptive execution of update transactions, update transaction will be queued then MS would retrieve the update transactions from the queue sequentially. Whenever MS retrieves the update request from the queue, the master server process the update transaction in two different ways based on the kind of database.

a) When the database has dynamic & non-temporal data (value of the data item not depends on time) then MS executes every update request from queue sequentially and does the following,

- Send Syn1 to all RSs along with the indices of data items to be updated
- After receiving Ack1 from all available RSs, execute update transaction
- Send update log transaction to all RSs and receive Ack2
- Send Syn2 to all RSs for re-broadcasting updated data items
- If the Ack3 has received, MS would retrieve the next update transaction from the queue.

Example

When non-temporal & dynamic database is stored on MS data base, it would be updated only by the external device frequently. The weather value of particular area is not highly depends on time. i.e., executing these update transaction after some delay would not affect the system. Here all update transactions would be queued then the master server executes the queued requests one by one by blocking the data items (to be updated) from broadcast transaction to sustain the data consistency.

b) When the database has temporal & dynamic data then MS retrieves an update transaction based on application specific condition. When the master server retrieves an update transaction from queue, all other update transactions for the same indices would be cancelled. Then master server does the following,

- MS retrieve an update transaction in FIFO order for the update of data[d1,d2,d3]
- Cancels all other update transactions for the update of any data items which are in current update list
- Then MS does the process as that of case 1.a.

Example

When the stock trading database is stored on master server, it would be updated only by the administrator. The stock value of particular entity is highly dynamic and time dependent. Update transactions should be executed and broadcasted immediately with out any delay. Whenever the stock value of an entity is in current update transaction, all other update transactions for the same entity would be cancelled in order to retain consistency.

Case 2: Read-Write Mobile Clients

In this case, database on the master server can be updated by mobile clients. The database may have temporal data or non temporal data.

a) When the database has non-temporal data and if the mobile clients issue the update transaction requests, then the entire request will be queued. MS will retrieve each and every update request of mobile clients sequentially. Then MS does the process as that of **1.a**.

Example

When the polling database is stored on database of MS and it is frequently updated by mobile clients. The mobile client's update transactions are queued. Then MS executes all the update transaction one by one. Then MS does the process as that of **1.a.**

b) When the master server stores the highly dynamic data and it would be updated by the mobile clients and their update requests will be queued on server. Then MS retrieves an update transaction from the queue, it does the following,

- MS consider only one update transaction for each group of update requests based on application specific conditions. Other requests for the same indices would be cancelled.
- MS does the process as that of case 1.b.

Example

When the bidding and auctioning database is stored on Master server, it will be frequently updated by clients rather than administrator. Whenever the bidding value of an item is updated by more mobile clients, they will be queued. Then the MS would consider an update transaction of a mobile client with highest bidding value for execution. All other requests on queue for the any one of the indices in update list will be cancelled. Then the server does the process in 1.(b).

B) Broadcast Replica Server Phase

This phase is responsible for broadcasting consistent data to the mobile clients. This phase process the following data sets,

BD → Set of data to be broadcasted.
BI → Set of indices (position of data items on data base) of data to be broadcasted.
UD → Set of updated data
LI → Set of indices of data to be updated and blocked for broadcast

In this phase, replica servers broadcast the data (which are not blocked) with their indices in series of broadcast cycles. For each broadcast cycle, the PTRCD approach broadcast the index of the first data item of current broadcast cycle to know about the arrival time of required data from doze mode itself.

- Whenever the replica server received any Syn1 signals from MS then RSs set the LI, block them from broadcast transaction then send Ack1 signal to MS
- If there is any update log transaction, then all RSs execute update log transaction and set UD. RSs send Ack2 to MS for signaling about its completion
- When RSs received any Syn2, then RSs compare every index of updated data in LI with the index of data to be broadcasted (BI), if there is any overlap then RSs broadcast that data item, index and remove it from BD to avoid duplicated broadcast. Otherwise, updated data items may already broadcasted in previous cycles then RS simply rebroadcast that data item, index. At the end of execution of update log, RSs unblock the data in LI then send Ack3 to MS to indicate the completion of data rebroadcast.

C) Broadcasting Algorithm of RS

The following algorithm shows the process of replica servers in PTRCD approach.

```
While(true){
 For each Bcycle set  BD,BI

        If ( TMT request on Queue){
         For each index I in Queue{
             Broadcast D[I] where  I  not in LI
             If I in BI
             Remove I ,D from BI, BD     } }

         If (RS received Syn1){
         Set LI=set of Indices to be updated by MS
         Block the data on LI from broadcast,
         Send Ack1 to MS }

        If (Update log Transaction Signal){
        Set UD, execute Update Log Transaction
        Send Ack2 to MS}

          If (RS received Syn2{
         For each  x  in LI{
          If (x   є  BI) // x to be Broadcasted
            in current Bcycle{
          Broadcast D[x]  from UD,LI
          Remove D[x]  from BD,BI
         //To avoide duplicated broadcast
                              } //If
        Else  // x may already broadcasted
                          previous Bcycles
        Rebroadcast D[x]  from UD,LI
     }//for
     LI={}
     Send Ack3 to MS
  }//if
Else  //No Update Log
If (X is the first index of  current Bcycle) then
     Broadcast  X from BI.
Broadcast D[x]   from  BD,BI    where  x  not in LI

}//while
```

D) Mobile Client Phase

This phase is responsible for caching broadcasted data item on the local cache of mobile clients. Each client can register the data request on their cache. They would be compressed with null value then saved. We have designed the cache size as half of a broadcast cycle size. Whenever the cache is full, LRU (Least Recently Used) algorithm is used for cache replacement. If the index of broadcasted data item is matched with the cached index then the data item will be compressed and cached to replace the null value or old value immediately. By using cache, the mobile clients can read the current data. i.e, the cached data would be decompressed when the mobile clients read the cache. When the mobile clients registered their data requirements, the client side applications decompress the cached content on their local memory. The decompressed index would be compared with broadcasted index, if so then its data item is compared with decompressed data item, if they are not same (updated data item or new data item) then the broadcasted data item would be compressed and re-cached. The following figure.2 shows the cache management architecture of PTRCD approach.

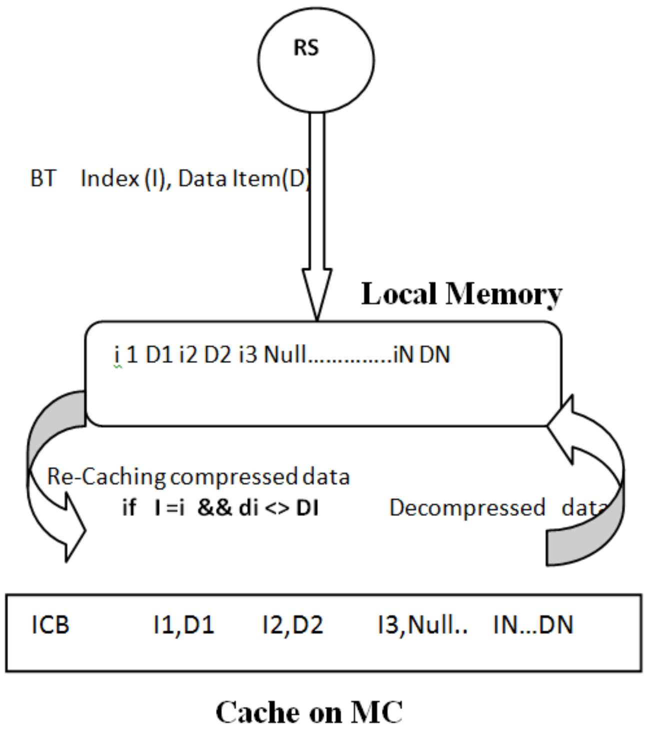

Figure 2. Cache Management of PTRCD approach

When there is a cache miss, the mobile clients can get it by knowing the arrival time of the data item. Since the index of the first data item of current bcycle is cached on mobile clients, the mobile clients can easily know about the period of particular data on broadcast channel (arrival time) even from doze mode. It highly reduces the power consumption and network communication of mobile clients. The updated data items would be cached on mobile clients through the rebroadcast of updated data items without any IR. The PTRCD approach makes use of deflator algorithm to compress and decompress the data objects and their indices to utilize the mobile cache efficiently. It includes the features of LZ77 & Huffman compression mechanisms. It effectively does the compression and decompression with out troubling throughput.

When the mobile clients wish to send the update transaction requests, they have to send through their uplink channel with the index of the data item to be modified and new data item to the master server (MS). The MS can execute or cancel the update transactions from the queue based on consistency checks.

E) Time constrained mobile transactions (TMT)

The proposed algorithm satisfies the time constrained mobile transactions by allowing the mobile clients to pull their data requests from RS. The mobile clients can send their requested data IDs to the nearest RS. They would be considered as higher priority data requests. The replica servers would broadcast the queued requests for each broadcast cycle if they were not in current update list. The proposed approach reduces the length of broadcast transaction through the smaller size of control information and by eliminating redundant broadcasting. Caching along with replication provides data reliability and quick data access to the time constrained mobile transactions.

E) Impact of Network Disconnection & Node Failure

Mobile clients are disconnected from a network voluntarily (to save the power) or due to the scarcity of network resources. Since our algorithm allows the mobile clients to cache their data requests, whenever the mobile clients disconnected from the network for a short time, they can get their data from the cache. The mobile clients can validate their cached data by tune in to the channel. As the proposed approach caches the starting index of current broadcast cycle, the cached data items (cached indices > starting index) would be invalidated in current or fore coming cycles otherwise (cached indices already broadcasted at the time of disconnection) the mobile clients may sent TMT request for them to invalidate the cached data instantly.

V. Conclusion and Future Enhancements

Simultaneous execution of broadcast and update transaction is controlled in proposed algorithm. It delivers the consistent data to the mobile clients using index based data broadcast. It provides high data availability through replica servers. It preserves the consistency using smaller size of control information to identify data inconsistencies and it saves the power consumption of mobile clients using caching mechanism. Since the algorithm rebroadcast the updated data immediately after the commitment, it is suitable for time aware mobile transaction also. But the usage of queue to disseminate consistent data may leads to delayed execution of update transactions. In future ,this algorithm can be extend to enhance performance of the broadcasting mechanism for location dependent data requests.

VI. REFERENCES

[1]. Dr. K.Chitra Manikandan, Dr.K.Senthamarai Kannan, S.Abirami. Index Based Ordered Broadcast with Status (IOBS) Algorithm for Consistent Data Broadcast, proceeding 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.

[2]. 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.

[3]. Dr.K.Chitra Manikandan, S.Abirami,Time Stamping Method For Consistent Data Dissemination To Read Write Mobile Clients, Proceeding of International Conference on Computer and Informatics (ICCCI-2012), Coimbatore, India ,Jan10-12,2012-IEEE, ,Vol2,1 edition 01-03-2012, pp 335-339.

[4]. Dr.K.Chitra Manikandan, S.Abirami,Efficient algorithm for disseminating consistent data to read –write mobile clients, International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.2, No.4, July 2012,pp 21-25.

[5]. Liangzhong Yin, ,Guohong Cao,Adaptive Power-Aware Prefetch in Wireless Networks,IEEE Transaction on wireless communication,Vol 3,pp 1648-1658.

[6]. Yan Huang, Yann-Hang Lee,Caching Broadcasted Data for Soft Real-Time Transactions,2001

[7]. Vahideh Baradaran Sorkhbai,Hadi Shahamfar .Detecting appropriate Replication approach considering client and server parameters in mobile environment, Proceedings of IEEE International Conference on Educational Technology and Computer(ICETC),2010.

[8]. Abdelkrim Beloued, JeanMarieGilliot, MariaTeresaSegarra,Franc¸oise Andr´e. Dynamic Data Replication and Consistency in Mobile Environments,2 nd International Doctorial Symposium on Middleware ’05, December 2005.

[9]. Anita Vallur , Mohammed Atiquzzaman, A survey of data replication techniques for mobile ad hocnetwork databases, The VLDB Journal (2008),Vol 17 ,pp 1143–1164.

[10]. Le Gruenwald , Shankar M. Banik, A Power-Aware Technique to Manage Real-Time Database Transactions in Mobile Ad-Hoc Networks,12 th international workshop on computing and processing(hardware & Sofrware), 2001,pp-570-574.

[11]. Alok Madhukar ,Æ Tansel O¨ zyer , Reda Alhajj,Dynamic cache invalidation scheme for wireless mobile environments, Wireless Netw ,2009,Vol.15, pp 727–740.

[12]. G. Anandharaj,Dr. R.Anitha,A Power-Aware Low-Latency Cache Management Architecture for Mobile Computing Environments, International Journal of Computer Science and Network Security, VOL.8 No.10, October 2008.

[13]. Kam-yiu Lam, Edward Chan and Joe Chun-Hung Yuen,Broadcast Strategies to Maintain Cached Data forMobile Computing System, Proceeding of International Conference on Advances in Database Technologies - LNCS 1552 (ER '98).

[14]. Guohong Cao,Proactive Power-Aware Cache Management for Mobile Computing Systems, IEEE TRANSACTIONS ON COMPUTERS, VOL. 51, NO. 6, JUNE 2002.

[15]. Kam-Yiu Lam, Edward Chan, Hei-Wing Leung and Mei-Wai Au,Broadcasting Consistent Data to Mobile Clients with Local Cache, Proceeding of International Conference ON ADVANCES IN DATA MANAGEMENT 2000.

[16]. Hei-Wing Leung,Joe Yuen,Kam-Yiu Lam and Edward Chan,Enhanced multi-version data broadcast for time constrained mobile computing systems, Proceeding of 13 th international workshop on Database and Expert systems applications, sep 2002,pp 747-751.

[17]. Kam-yiu Lam, Edward Chan and Chun-Hung Yuen, Data Broadcast for Time-Constrained Read-Only Transactions inMobile Computing Systems, proceeding of international workshop Advance Issues of E-Commerce and Web based Information system,1999,pp-11.

[18]. Iianliang XU,Qinglong Hu,Wang-Chien Lee,Dik Lun Lee,An optimal cache replacement policy for wireless data dissemination under cache consistency, proceeding of International Conference on, Parallel Processing, 2001.