Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 13 Issue 2 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

An Efficient Concurrency Control Technique for Mobile Database Environment

By Md. Anisur Rahman

Dhaka University of Engineering & Technology Gazipur, Bangladesh

Abstract - Day by day, wireless networking technology and mobile computing devices are becoming more popular for their mobility as well as great functionality. Now it is an extremely growing demand to process mobile transactions in mobile databases that allow mobile users to access and operate data anytime and anywhere, irrespective of their physical positions. Information is shared among multiple clients and can be modified by each client independently. However, for the assurance of timely access and correct results in concurrent mobile transactions, concurrency control techniques (CCT) happen to be very difficult. Due to the properties of Mobile databases e.g. inadequate bandwidth, small processing capability, unreliable communication, mobility etc. existing mobile database CCTs cannot employ effectively. With the client-server model, applying common classic pessimistic techniques of concurrency control (like 2PL) in mobile database leads to long duration Blocking and increasing waiting time of transactions. Because of high rate of aborting transactions, optimistic techniques aren`t appropriate in mobile database as well. This paper discusses the issues that need to be addressed when designing a CCT technique for Mobile databases, analyses the existing scheme of CCT and justify their performance limitations. A modified optimistic concurrency control scheme is proposed which is based on the number of data items cached, amount of execution time and current load of the database server. Experimental results show performance benefits, such as increase in average response time and decrease in waiting time of the transactions.

Keywords : mobile database, optimistic concurrency control, mobile host, fixed host, base station, client-server.

GJCST-C Classification : H.2.4

AN EFFICIENT CONCURRENCY CONTROL TECHNIQUE FOR MOBILE DATABASE ENVIRONMENT

Strictly as per the compliance and regulations of:

An Efficient Concurrency Control Technique for Mobile Database Environment

Md. Anisur Rahman

Abstract - Day by day, wireless networking technology and mobile computing devices are becoming more popular for their mobility as well as great functionality. Now it is an extremely growing demand to process mobile transactions in mobile databases that allow mobile users to access and operate data anytime and anywhere, irrespective of their physical positions. Information is shared among multiple clients and can be modified by each client independently. However, for the assurance of timely access and correct results in concurrent mobile transactions, concurrency control techniques (CCT) happen to be very difficult. Due to the properties of Mobile databases e.g. inadequate bandwidth, small processing capability, unreliable communication, mobility etc. existing mobile database CCTs cannot employ effectively. With the client-server model, applying common classic pessimistic techniques of concurrency control (like 2PL) in mobile database leads to long duration Blocking and increasing waiting time of transactions. Because of high rate of aborting transactions, optimistic techniques aren`t appropriate in mobile database as well. This paper discusses the issues that need to be addressed when designing a CCT technique for Mobile databases, analyses the existing scheme of CCT and justify their performance limitations. A modified optimistic concurrency control scheme is proposed which is based on the number of data items cached, amount of execution time and current load of the database server. Experimental results show performance benefits, such as increase in average response time and decrease in waiting time of the transactions.

Keywords : mobile database, optimistic concurrency control, mobile host, fixed host, base station, client-server.

I. Introduction

In the modern decade, Because of having enhanced processing capability, easy and swift access to information as well as reduction in prices, mobile devices have achieved vast use in data processing services like mobile banking, traffic control, e-commerce, money transfers etc.

In these application and communication process of mobile devices client-server, peer to peer, and ad-hoc architectures are most recognized architectures of mobile database system. Fig. 1 displays client-server architecture. Mobile clients/Host (MH) e.g. Palmtops, Laptops, PDA's, Cellular phones etc. and fix host (FH) e.g. Terminals, desktops, servers etc and mobile base stations (BS) are three most important elements of this model. In client-server model, mobile clients are connected to fixed host through mobile base station. At present, using wireless 802.11 protocols, it is possible to have a synchronized and fast connection between mobile client and server. But, properties of mobile devices still impose some limitations upon assessment and data processing in mobile environment. Therefore, for constant and continuous processing, required data are copied partially from server to mobile client.

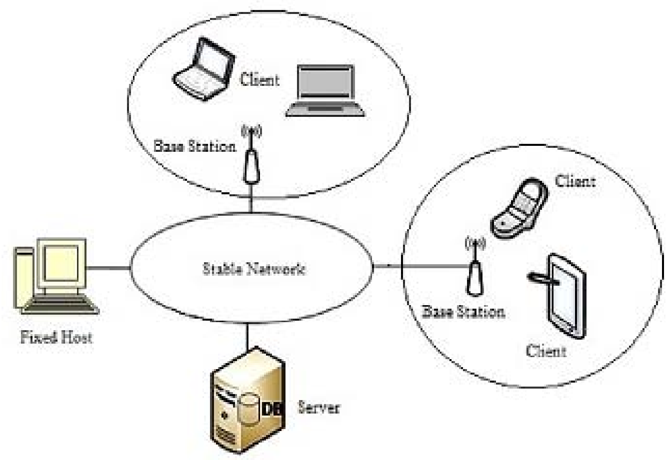

Figure 1 : Mobile Database Environment

In a mobile database environment, each node has an area of influence called cell, only within which others can receive its transmissions. In a distributed client-server mobile database system, not only clients but also servers are mobile, wireless and battery-powered. Mobile database systems are of interest because, it can be easily deployed in a short time, and end users can access and operate data anytime and anywhere. Examples of mobile database applications include law enforcement operations, automated battlefield applications, natural disaster recovery situations where the communication infrastructures have been destroyed, self-organizing sensor networks for data collecting, and interactive lectures or conferences for data exchange without pre-installed infrastructures.

However, the flexibility and convenience in Mobile Databases raise new issues when performing concurrency control (CC). CC is the activity of preventing transactions from destroying the consistency

Author : Department of CSE, Dhaka University of Engineering & Technology Gazipur, Bangladesh.
E-mail : ansiur.rahman.duet@gmail.com

of databases while allowing them to run concurrently, so that the throughput and resource utilization of database systems are improved and waiting time of concurrent transactions is reduced. When designing a CC algorithm the factors that should be taken into considerations are mobility, low bandwidth, multi-hop communication, limited battery power, limited storage, frequent disconnections, wireless communication delay, less processing power, frequent disconnections and unbounded disconnection time etc. When the execution is prolonged, the probability of conflicts with other transactions becomes higher and, consequently, transactions are likely blocked if a pessimistic CC method applies or restarted if an optimistic CC method is in use. The existing CC techniques for traditional mobile network databases, in which only clients are mobile and battery-powered, cannot be directly, because of a number of factors related to its architecture, availability and sharing of hardware and software resources, distribution of data, and mobile client's processing capability.

In this paper we proposed a new scheme that is reduces number of abortion as well as minimize the connection time of client and server by enhancing concurrency.

II. Related Works

Due to the boundaries, restrictions and specific properties of mobile devices cause traditional concurrency control techniques e.g. 2PL to exhibit a reduced amount of effectiveness in mobile database. A variety of researches have done in this field. These researches introduce new techniques of concurrency control or adapt the existing techniques of concurrency control with the requirements of mobile environment.

Because of the inherent limitations of mobile devices, constant and uninterrupted connection between client and server is not achievable over the transaction period. In locking-based pessimistic protocols, continuity of connection of mobile client with server is compulsory. If at the period of transaction, the connection is interrupted, the possibility of infinite blocking of transactions and deadlock is appeared. So, based on time out, many researchers have been done regarding methods of non-exclusive locks. In these techniques, if transaction doesn't end within the expected time, locks get caught by transaction will be free.

References [9, 10] showed a technique of locking with non-exclusive lock based on time out. In reference [10], a dynamic timer was used to solve the problem of transactions` blocking. In this method, transactions should be finished in particular time, otherwise they would abort. In reference [11], for increasing the efficiency of [10],transactions which don't finish in specified time and are near to final of transaction, wouldn't be aborted. They are allowed to continue execution for specified time. In methods basing on timer (time out) problems like long connection of mobile client with Server, estimated the amount of timer, and the length of transaction, do exist. Because of wrong estimation of the required time for completion of transactions, these problems could result in incorrect abortion of transactions. This suffers from the problem of frequent rollbacks due to regular expiry of the timer and wastage of computation. A protocol based on AVI is proposed in [12]. This method has still the problem of transaction blocking and computational overhead. This problem has been mentioned in reference [13]. Multi-version concurrency control based on MV2PL protocol and timestamp being introduced in [14] are in fact an extension of method [15].

In [17, 18], combination of optimistic and pessimistic is performed according to the semantic of operators.

In [17], changing two phases locking (2PL) and considering the semantic of transaction`s operators, are executed for increasing the concurrency degree of transactions. If conflicted operators are compatible semantically, then, they can choose a resource simultaneously. In this method, if compatible transactions lock a resource, there would be a resource scarcity for incompatible transaction. Moreover, there is the possibility of high rate of abortion through reconciliation process.

In reference [13] optimistic method is utilized. At the time of completion of transactions, updated data are sent to other mobile transactions using these data. This transfer is in multi-cast status. This technique reduces abortion rate of transactions.

Reference [19] puts emphasis on asymmetry of connecting bandwidth between mobile client and server. In [19] optimistic method is under consideration in which in the time of data updating, timestamp of transactions is set dynamically and data is broadcast for mobile clients.

Optimistic concurrency control based on timestamp ordering in [20] is adapted for broadcast environments. Also the introduced method in [21] is suitable for broadcast environments.

[3] OPCOT algorithm is introduced by assigned timestamp to operators and based on optimistic method according to concept of commitment ordering schedulers. So, reducing the waiting time as well as abortion rate, providing high concurrency, secure from deadlock and starvation, reducing computational are objectives to be considered in concurrency control protocol in mobile database environment.

III. Proposed Method

Some concurrency control scheme performs better in some particular database systems e.g. Time stamp based multi-version protocols is better in real time databases. So according to the demand of the mobile database, we can define some specific method for a defined type of transactions and consider its condition to set best performed rule for it. We can categorize transaction operations as:

Transactions that only Read an item (TR), Transactions that only Write an item (TW), Transactions that Read as well as write an item that is Update (TU), Transactions that inserts an item (TI), Transactions that delete an item (TD) etc.

In my proposed method, I have used Time-stamp based multi-version protocol for TR and TW. So TR always gets the most recent version of the data elements and TW just creates a new version of data element and the old version remains intact and hence possibility of occurring conflict is zero.

For the TU, writing conflict may happen. So it is essential to resolve the conflicts among the TUs effectively to maintain integrity of the data item.

In my proposed method, when a client MH requests for the data items e.g. Xj. On which the MH going to have write operation(s), Server stores an entry for that MH including the client Id, Time of the beginning of transaction execution (TB), the list of data items Xi for that client, maximum validation period PV, Rank RC.

T_B is maximum duration to execute and send cache to server which is supplied by MH for the use of server to achieve more concurrency. PV is a time limit within which MH have to perform its operation and send cache to server to commit else request will be discarded from the transaction list of the server. PV is determined by the TB and the bandwidth of the connection to that client. RC is an integer that keeps track of the commit attempt of a transaction.

As all transactions are executed locally affecting the cache of the client, client sends partial update of the transaction for a data item that has no further update operation within this transaction for the partial commitment to the server if connection between mobile client and server be available. It would increase the concurrency significantly. In this case server checks for the conflicts and resolves by commitment algorithm. If connection is not available during execution then at the end of all updates in client cache, updated cache is sent to server to be committed. All kind of update is done according to the commit algorithm described below:

When any client MH_i sends cache to server, server finds the clients MH_a, MH_b with which MHi conflicts. Server performs following operations:

If MH_i have expired maximum validation period of MH_i, Server cancels the commit operation and causes to restart the transaction of MH_i.

If Executed Time Tex of MH_i < $\sum$j Tex of MH_j, for all j X_i
Then
If RC > $\sum$j RC for all j Xi

MH_i commits; invalidating the conflicting clients causing to restart their execution immediately with updated value of the data item.

Else

MH_i aborts and restarts its execution.

Else

MH_i commits and causes to restart their execution immediately with updated value of the data item.

IV. Performance Analysis

The proposed concurrency control scheme is assessed in this section. To perform assessment, it is compared with optimistic concurrency control methods as proposed technique is based on optimistic methods. In order to assess and compare performance of proposed technique, optimistic technique is simulated as well. I have used of network simulator NS-3 [22, 23] version 3.12 to implement my protocols.

Simulation is done using C + + in application layer protocols.

A circular-shaped area (D=1000 meters) space, a fix host (server), 5 mobile base stations, and 100 mobile clients are used in simulation environment. Mobile hosts can move randomly in various directions and communicate their requirements by mobile base station to the server.

Simulation is done for getting the comparison result values for the optimistic method and proposed scheme. Since response time and waiting time of a transaction are two important parameters of performance, the graph is drawn and studied only for the two parameters. The load of the system is increased incrementally.

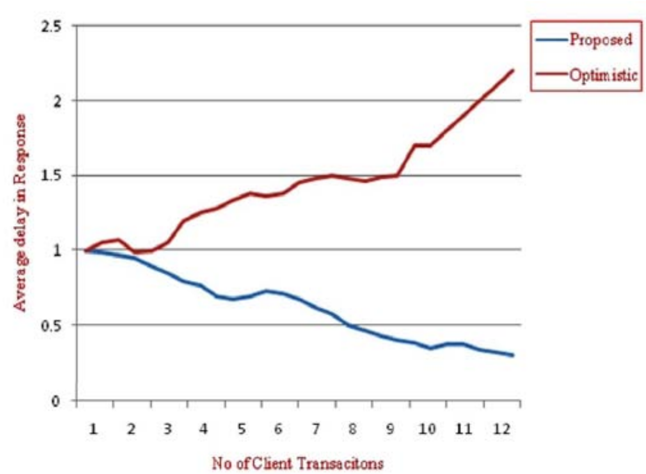

Figure 2 : Comparison of avg delay in response with number of client transactions

Global Journal of Computer Science and Technology (C) Volume XIII Issue II Version I Year 2013

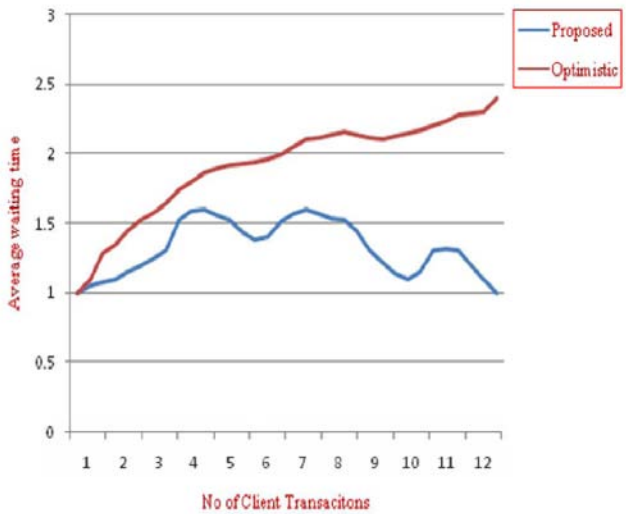

Figure 3 : comparison of avg waiting time with number of client transactions

In the figures Fig. 2 and Fig. 3 the comparison of the average delay in response as well as the avg waiting time with the increase in number of client transactions is shown. Average delay in response time is the average increase in normal transaction execution time due to concurrency. Waiting time of the transaction is the time for which temporary hiatus in execution of the transaction appear due to non-availability of shared data items. With the increase of the number of transactions avg. delay in response and waiting time increases significantly whereas proposed method performs better.

V. Conclusions

In this paper, a new concurrency control technique based on optimistic method in mobile database environment with client-server model is introduced. This method is based on Optimistic method, so there is no blocking or deadlock in transactions. Moreover, concurrency Degree of transactions is high. To reduce the rate of abortion, proposed algorithm, based on broadcast commit, applies the technique of partial update as well as assigning priority depending on some parameters. This leads to increase in the rate of commitment of transactions. There some overhead for the determining execution time of transaction which is worthless and is tolerable. There is plan to improve this method by introducing time stamp in it.

References Références Referencias

1. Joe, C.H.Y., Chan, E., Lam, K.Y., Leung, H.W, "An Adaptive AVI-based Cache Invalidation Scheme for Mobile Computing Systems", DEXA 2000, pp.155-178, IEEE Computer Society, USA.
2. Ho-Jin Choi, Byeong-Soo Jeong, "A Timestamp-Based Optimistic Concurrency Control for Handling Mobile Transactions", ICCSA 2006, LNCS 3981, pp. 796 -805, 2006.
3. K. Ali and b. Ahmad," A Concurrency Control Method Based on Commitment Ordering in Mobile Databases", International Journal of Database Management Systems (IJDMS) vol.3, no.4, November 2011.
4. Minsoo Lee, Sumi Helal, "HiCoMo: High Commit Mobile Transactions", Distributed and Parallel Databases, Vol.11, pp.73 -92, 2002, Kluwer Academic Publishers.
5. Mohammed Khaja Nizamuddin, Syed Abdul Sattar, "Adaptive Valid Period Based Concurrency Control In: Recent Without Locking in Mobile Environments" Trends in Networks and Communications, Springer CCIS, Vol.90, Part 2, pp 349-358, 2010, Springer Heidelberg.
6. Nadia Nouali, Anne Doucet, Habiba Drias, "A Two-Phase Commit Protocol for Mobile Wireless the Australasian Database Environment", Vol. 39, 16 Conference, 2005.
7. Patricia Serrano-Alvarado, Claudia Roncancio, Michel Adiba, "A Survey of Mobile Transactions", Distributed and Parallel databases, 16,193-230, 2004, Kluwer Academic Publishers.
8. Salman Abdul Moiz, Dr. Lakshmi Rajamani, "An Algorithmic approach for achieving Concurrency in Mobile Environment", INDIACom, 209-211, 2007.
9. K Vijay, "TCOT -A Timeout-Based Mobile Transaction Commitment Protocol," IEEE Transactions on Computers, vol. 51, pp. 1212-1218, 2002.
10. S.Moiz and L. Rajamani, "Single Lock Manager Approach for Achieving Concurrency Control in Mobile Environments," in 14th International Conference on High-Performance Computing, Goa, India, 2007, pp. 650-660.
11. M. Salman Abdul, "Concurrency Control Strategy to Reduce Frequent Rollbacks in Mobile Environments," in International Conference on Computational Science and Engineering, 2009, pp. 709-714.
12. S. A. Moiz and M. K. Nizamuddin, "Concurrency Control without Locking in Mobile Environments," in Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on, 2008, pp. 1336-1339.
13. D. L. R. Salman Abdul Moiz, "A Real Time Optimistic Strategy to achieve Concurrency control in Mobile Environments using on-demand multicasting," International Journal of Wireless & Mobile Networks (IJWMN), vol. 2, pp. 172-185, 2010.
14. S. Madria, M. Baseer, V. Kumar, and S. Bhowmick, "A transaction model and multiversion concurrency control for mobile database systems," Distributed

and Parallel Databases, vol. 22, pp. 165-196, 2007.
15. S. K. Madria, M. Baseer, and S. S. Bhowmick, "A Multi-version Transaction Model to Improve Data Availability in Mobile Computing," in International Conferences DOA, CoopIS and ODBASE 2002, pp. 322-338.
16. C. Sung Ho, L. JongMin, H. Chong-Sun, and L. WonGyu, "Hybrid concurrency control for mobile computing," in High Performance Computing on the Information Superhighway, 1997. HPC Asia '97, 1997, pp. 478-483.
17. A. C. Carmine Cesarano, Vincenzo Moscato, Antonio Picariello, Antonio d'Acierno, "A Hybrid Approach for Improving Concurrency of Frequently Disconnecting Transactions," International Journal of Computer Science and Network Security (IJCSNS), vol. 7, pp. 205-215, 2007.
18. A. Chianese, A. d'Acierno, V. Moscato, and A. Picariello, "Pre-serialization of long running transactions to improve concurrency in mobile environments," in Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on, 2008, pp. 129-136.
19. C. S. Lee, K.-W. Lam, and S. H. Son, "Concurrency Control Using Timestamp Ordering in Broadcast Environments," The Computer Journal, vol. 45, pp. 410-42 2, January 2002.
20. C. S. Lee, K. Wa Lam, and T.-W. Kuo, "Efficient validation of mobile transactions in wireless environments," Journal of Systems and Software, vol. 69, pp. 183-193, 2004.
21. Salman Abdul Moiz ,Dr. Lakshmi Rajamani, "Concurrency Control Strategy to Reduce Frequent Rollbacks in Mobile Environments," CSE, vol.2, pp.709-714, 2009 International Conference on Computational Science and Engineering, 2009.