Analytical Performance of Delay due to Bandwidth Variability in Concurrent Transactions Mixture

Ahmad aL-Qerem
Computer science department
Zarqa University, Zarqa, Jordan
Email: ahmad_qerm {at} zu.edu.jo

***Abstract*—unpredictable propagation delay of mobile transaction delivery imposes a serious overhead on the execution performance of fixed host transactions. The mobility of clients in a mobile computing system also greatly affects the distribution of workload in the communication network. A shared data item that is locked by a mobile transaction could hinder operations of other fixed host transaction from being executed. The delay due to the bandwidth variability is highly affected the blocking of fixed transactions. Past studies have used locking protocols to maintain consistency of transaction regardless the difference of underling communication media that used by those transactions, however, no quantitative guideline has been given. In this paper, we provide analytical study of the delay due to bandwidth variability and proposed a heuristic lock request procedure which can be adopted by any locking protocol used for concurrent transactions mixture by dynamically reordering fixed transaction's operations therapy decreasing the affect of delay due to bandwidth variability. Simulation results show that our solution effectively improve the concurrency protocol in mixture transactions environments.**

Keywords—transactions mixture; concurrency; bandwidth variability

I. INTRODUCTION

In recent years, the research in mobile database system is receiving growing attention due to a large number of potential mobile computing applications [1].

Owing to the intrinsic limitations of mobile computing systems, such as bandwidth variability and frequent disconnection, the design of an efficient and cost-effective mobile database system which accessed by mixed mobile fixed host transactions requires techniques, which are quite different from those developed for traditional database systems over wired networks [2], [6] and [10]. How to decrease the effect of wireless environment on mixed mobile fixed host transactions over a mobile network and, at the same time, to maintain the consistency of the databases involves various challenging research issues [1] [11]. Up to now, to our knowledge, this is the first paper that studies the effect of bandwidth variability on the concurrency control protocol.

The delay of mobile transaction operations caused by the bandwidth variability can greatly affect the performance of any adopted concurrency control protocol. In locking based concurrency control protocol the contention and blocking increase as the number of conflicting active transaction increase; for example; in two phase locking protocol if the mobile transaction operation hold lock on a data item remain locked until all of mobile transaction operations delivered. The time between arrivals of the mobile transactions is highly affected by the transmission delay caused by the bandwidth variability. As a result the contention and conflict increase, so the performance of fixed host transactions will degrade dramatically.

Although many excellent concurrency control protocols have been proposed for mobile database system,[4],[5],[6] and [8], most of these protocols ignore the effect of mobile transaction scheduling on the performance of fixed host transaction execution. The unpredictable propagation delay of mobile transaction delivery imposes a serious overhead on the execution performance of fixed host transactions. The mobility of clients in a mobile computing system also greatly affects the distribution of workload in the communication network. Disconnection between clients and stations is common [10] [11]. The poor quality of service provided by a mobile network usually seriously increases the overheads in resolving data conflicts and affects the performance of existing concurrency control protocols which do not consider the characteristics of mobile networks.

This paper proposes augmented lock request procedure which could be used with any concurrency protocol that based on locking, we use 2PL as its popularity and apply the proposed lock request procedure and called it Augmented-2PL such augmentation is based on the concept of conflict reduction due to bandwidth variability which used

The remaining parts of the paper are organized as follows. Section II defines the mixture environment Section III study analytically the effect of bandwidth variability. Section IV introduces the augmented procedure Section V reports the performance evaluation The Conclusions of the paper are in Section VI.

II. MIXTURE ENVIRONMENT

We consider a mobile computing environment with a network consisting of fixed and mobile hosts (FH, MH), see Fig.1. MHs could be of different nature ranging from PDAs to personal computers. Shared data are distributed over several database servers running, generally, on FHs. MH may run database management system (DBMS) modules and may provide some services to other hosts. While in motion, an MH may retain its network connection through a wireless interface supported by some FHs which act as Base Stations (BS). The geographical area covered by BSs a cell. Each MH communicates with the BS covering its current cell. The process during which an MH enters into a new cell is called hand-off. Applications in mobile environments are confronted to particular characteristics and limitations imposed by hardware such as: low and variable bandwidth, frequent disconnections, high communication prices, variable hardware configuration (due to plug-in components), limited display, battery autonomy, processing power and data storage. These limitations/variations lead to a lot of potential failure modes that affect data management process (e.g. queries, replication, caching, transactions, etc.) [9].

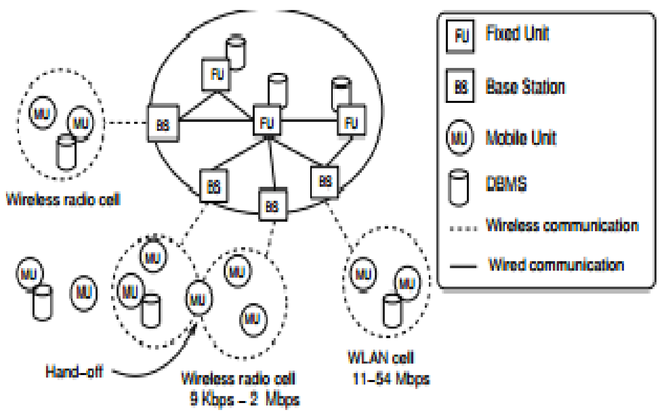

Fig.1: Reference Model of a Mobile Computing Environment

Informally, a transaction is a set of operations that translate a database from a consistent state into another consistent state. Like in [7], an MT is a transaction which issued by a mobile host. The participation of an MH introduces dimensions inherent to mobility such as: movement, disconnections and variations on the quality of communication. As we will see in the following, TM supporting has to adapt their functionalities to deal with these dimensions. In the scope of this paper we focus on systems with a client-server architecture where clients are MH interacts with mobile databases by invoking transactions. A transaction represents a sequence of read write operations on database items. There are two types of transactions in the mobile environment: fixed or wired transactions and mobile transactions. A fixed transaction is submitted directly to a database on the same host while the mobile transaction submitted by mobile device.

III. Bandwidth Variability

Bandwidth Variability occurs as the MH changes location, the ability to change location while retaining network connection is the key motivation for mobile computing. As mobile computers move, signal strength to the device varies, which can cause a loss of data or variations in bandwidth. Due to such variation in bandwidth, the enter arrival time of mobile transactions will be vary. As a consequence, the number of blocked fixed host transaction will be increased. Bandwidth variability occurs due two main reasons:

A. Different traffic loads:

Since the Bandwidth is divided among the mobile users sharing the same cell, as in Fig.2(a) the mobile host MH1 is send the operations $o_1$and o_2 to the database server and delivery of o_3 and o_4 are delayed.

B. Handoff:

Due to a change in the physical location, an MH can switch its supporting BS when moving to a different cell. This leads to the need for a hand-off procedure to enable the new BS involved to support and maintain the connection with the MH. Fig.2 (b) the mobile host is send o1 and o2 then switch to another BS which may delay the delivery of operations o3 and o4.

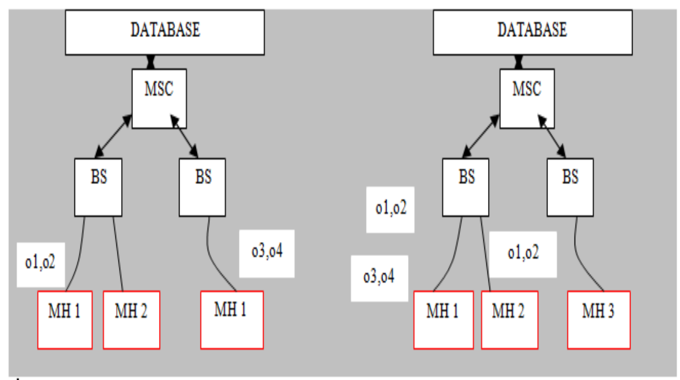

Fig.2: (a): Different traffic loads (b): Handoff

Consider the queuing system in Fig.4 with *n* data items. For each data item D_i there is a queue of mix transaction operations o^t_{bg} where *t* for transaction type (i.e. *m*, *f*: for mobile and fixed host transaction respectively and *b* for transaction number, if *b* not specify, we assume that these operation belong to any arbitrary transaction and *g* for operation number within transaction.
Assumption:

(a) Uniform data access probability for both mobile and fixed host transaction.
(b) Regular time between arrivals of fixed host transaction operations.
(c) Ignoring the deadlock overhead in the delay estimation of bandwidth variability.

Transaction T_a have to wait for the mobile transaction T_m on data item D_1 ,and the mobile transaction operation O^m_{b2} on data item D_n see *p* fixed host operations ahead of it. So, T_a remains blocked until all operation ahead of it release their locks as we see in Fig. 3. Transaction T_a also can't release the lock on a data item D_n until it got the lock on a data item D_3

(i.e. assume for simplicity T_a has two operations only). While the case of low bandwidth variability there is *q* operation instead of *p* that O^m_{b2} see a head of it , where $q < p$ so the blocking for mobile transaction is less in the low bandwidth variability environment . In general the blocking amount of transaction T_a is a function of queue length ahead of the last transactions operation and this can be estimated in term of delay cause by a transaction as follow:

O^f_k	O^f	O^f	O^f	O^m_{b1}	D_1

O^f	O^f	O^f	O^f	O^f_k	D_2

O^f	O^f_{a2}	O^f	O^f	O^f	D_3

•

•

O^m_{b2}	……p-operation ………..	O^f_{a1}	D_n

Fig.3: queuing system of mixture transaction's operations

For the first operation of the mobile transaction the cost will be $j*(T + D)+T$ where *j* is the number of locks to be acquired after the current one , *T* is the average processing time for any transaction operation at the database server . The second term represents the processing time required for the data item currently being queued for. The first term accounts for the fact that the data item to be locked on the current queue will have to remain locked until the transaction acquires his remaining *j* locks (requiring *j* delays) and serving those *j* operations (requiring *j* multiples of *T*).

In case of granting lock to the mobile transactions operation, it has to remain locked until the mobile transaction acquires its remaining locks. if *p(i)* is the probability that a mobile transaction operation have *i* remaining operations after the first which granted its lock and *q(i)* is the probability that the latest mobile transaction operation sees *i* fixed host operations ahead of it , then *D (n)* will be the delay caused by the first mobile transaction operation if latest delivered operation see at most *n* fixed host transaction operations ahead of it.

$$D(n) = q(0)*0+q(1)*\left[\sum_{j=1}^{m-1} b(j)*j*(T+D)+T\right] + q(2)*$$
$$\left[\sum_{h=0}^{m-1}\sum_{j=0}^{m-1} b(h)*b(j)*j*(j+h)*(T+D)+2*T\right]$$
$$+\ldots\ldots+q(n)*$$
$$\left[\sum_{l=0}^{m-1}\ldots\ldots\sum_{h=0}^{m-1}\sum_{j=0}^{m-1} b(l)*\ldots\ldots*b(h)*b(j)*j*(l+\ldots\ldots+h+j)*(T+D)+n*T\right]$$

Since value of n increase as the bandwidth variability increase the number of active fixed host transactions will increase which may increase the average queue length that a fixed host operation sees ahead of it. To simplify the above equation, let J be the average number of locks transactions require after the current one, then delay caused by the mobile transaction where the last delivered operation sees n fixed host operations ahead of it become average queue length multiply by the time needed for each operation in the queue to got the lock on the data item.

$$D(n) = q(0)*0 + q(1)*(J*(T+D)+T) + q(2)*2*(J*(T+D)+T) + q(3)*3*(J*(T+D)+T) + \ldots + q(n)*n*(J*(T+D)+T)$$
$$= (q(0)*0+q(1)*1+q(3)*3+\ldots.+q(n)*n)*(J*(T+D)+T)$$

D (n) =Q(J*(T+D) +T)) Where Q is the average queue length.*

In presence of bandwidth variability there is a high probability for the time between arrival of mobile transaction operations to be of variable length .which result in increasing the number of active transactions in the system which increase the contention and decrease the response time and throughput of fixed host transactions. Based on our assumption (i.e. uniform data access and constant time between arrivals) the average queue length ahead of any fixed host operations Q_f .so the delay caused by bandwidth variability D_{bv} will be

$$D_{bv} = (n-Q_f)*(J*(T+D)+T))$$
$$= n*(J*(T+D)+T)) - Q_f*(J*(T+D)+T))$$
$$= n*(J*(T+D)+T)) - c$$

where c is the delay part of fixed host transaction which is constant time based on our assumption and n is the maximum queue length seen among all mobile transaction operations.

Based on the first part of D_{bv} which represent a bandwidth variability delay, the average number of conflicting active transaction in the system will be increase by the value *Inc*

$$\text{Inc} = \frac{T_L * D_{bv}}{\lambda}$$

Where T_L the transaction length, D_{bv} is bandwidth variability delay and λ is the time between arrivals of fixed host transaction operations. In the following section we show how our augmented lock method attempt to decrease the number of conflicting active transaction based on heuristic which combine *RR* and *RBBAR* rules.

IV. Augmented Lock Request

To deal with different level of bandwidth variability in concurrent transactions mixture, a lock request procedure proposed see Fig. 5 based on the following two rules:

A. Restart- Reinitiate (RR)

Restart-reinitiate locking method try to maintain the same number of active transactions based on the notion that when the number of active transactions increases, the contention increase and the average wait time for each transaction increase and the response time and throughput decrease. So we restart the mobile transaction which have operation not delivered yet, followed by system initiated without intervention of the user after delivery all of its operations. Restart reinitiate is economic solution when there is a high bandwidth variability or environment of high disconnection probability.

B. Reorder Based on Blocking Activity Ratio (RBBAR)

Another issue that bandwidth variability can affect the performance of fixed host transaction schedulability even that all of mobile transaction operations are delivered is the blocking time Occur as consequence of variable time between operations arrival.

To come over this affect the reorder rule based on the blocking –activity ratio that try to minimize the blocking time by mobile transaction and increase the response time for both mobile and fixed host transaction as follow

Activity to blocking ratio $(T, n) = \frac{k}{D_{(n-k)}}$ where k is the number of operation already held the lock on their data items and D (n-k) is the time need for this transaction to get the lock on the remaining data item. For example the blocking –activity ratio for transaction T_i in the Fig 3 is 1/(3 *T) since the transaction T_i hold the lock on a data item D_n and waiting for the lock on the data item D_3 which have *3* operation ahead of it in the queue on this data item.

In our protocol, the fixed host transactions are given preferences in using the data item as their operations dose not suffer from the unreliable communication on other hand the update made by the fixed transaction increase the currency of the data accessed by the mobile transactions. Restart initiate and reorder method are used to decrease the effect the of mobile transaction delivery in presence of bandwidth variability. When a first operation of mobile transaction requests a lock, the transaction manager check the delivery of the last operation , if its delivered then adjust the lock request based on the blocking activity ratio of conflicting transactions in order to minimize the number of active fixed host transaction that may blocked by mobile transaction in case of granted the lock on a data item for the mobile transaction operation, if the last operation is not delivered yet the transaction manager did not know if the delay cause by the bandwidth variability or the mobile host may disconnected, so the transaction manager chose the economic solution in this case by restarting the transaction and reinitiate it by the system at delivery time of the last operation.

Lock-Request procedure (Ti, Mode, Transaction type, D_k)
Begin
If requested Transaction type = mobile **Then**
If (delivery of the last operation = true) **Then**
For all conflicting transaction with T_i find the blocking activity ratio
Adjust the position of each operation to be after the most active fixed transaction with least blocking value (RBBAR)
(Max ($\frac{k}{D_{(n-k)}}$) for all T_j where T_j in the conflict set of T_i
And mark T_i as fixed host transaction).
End for.
Else restart the transaction followed by system initiated as a fixed host transaction. *(RR)*
Else follow the locking protocol rule
End

Fig.4: Lock-request procedure

V. PERFORMANCE EVALUATION

A simulation model was developed to study the performance of 2PL with augmented lock request procedure against conventional implementation of the two-phase locking. Actually this procedure is independent of the locking protocol. It can be used with any locking protocol. We use here as example the popular 2-phase locking protocol. When the lock on the requested data item is released, then one or more of the waiting transactions may be granted the requested lock. The choice of such transactions has always been based on the policies were implemented by assigning cost values (i.e. activity ratio) for blocked transactions based on RR and RBAR rules *algorithm 1* we name it Augmented 2PL to distinguish it from conventional 2PL. Various simulation experiments were conducted to study the relative performance of the proposed lock requests using the general transaction processing model. Figs. 5 and 6, respectively, plot the throughput and the restart ratio, for using lock request procedure and the conventional one by changing the bandwidth variability at the x-axis. We use the zipf distribution to simulate the bandwidth variability change, as the theta increase the variability increase. Here we notice that with and without using lock request procedure approximately have the same performance when the bandwidth variability low. At high level of bandwidth variability, differences start to appear, where using the lock request procedure outperformed the conventional protocol.

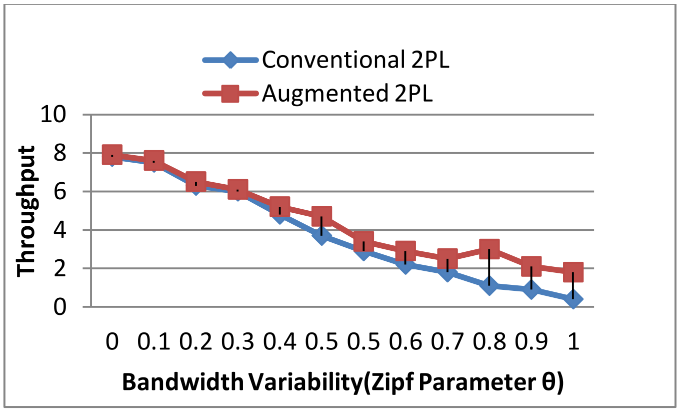

Fig. 5: Effect of lock-request procedure (Throughput).

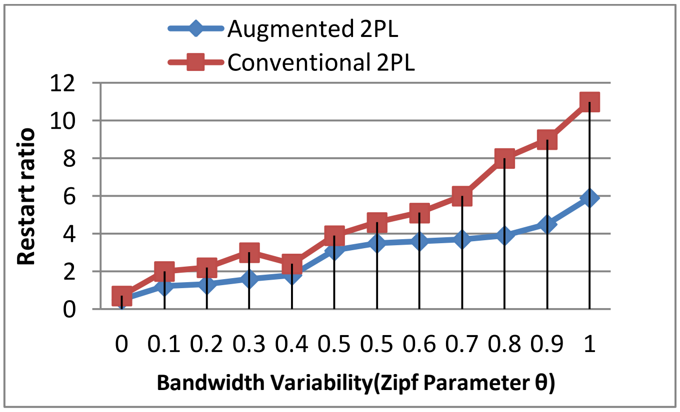

Fig. 6: Effect of lock-request procedure (Prestart ratio).

VI. CONCLUSION

Aspects related to bandwidth variability for concurrent transactions mixture and the delays due to such variability have been studied. The results show that, in this case, the queuing delay is highly affect the fixed transaction throughput and should take into considerations in designing the concurrency protocol in the mixture environment. This delay occurs due to high bandwidth differences between wired and wireless media. The results also bring out the importance role concurrency control mechanisms may play in performance of fixed transactions in the mixture environment.

ACKNOWLEDGMEN

This research is funded by the Deanship of Research and Graduate Studies in Zarqa Private University /Jordan

REFERENCES

[1] M. mohana; C. jayakuma" A Review on Concurrency Issue in Mobile Enviroenmnet "International Journal of Computer Networking,Wireless and Mobile Communications (IJCNWMC)ISSN(P): 2250-1568;Vol. 3, Issue 5, Dec 2013

[2] Darin Chan and John F. Roddick, Summarisation for Mobile Databases, Journal of Research and Practice in Information Technology, Vol. 37, No. 3, pp281 August 2005

[3] Jan Lindstrom, "Optimistic Concurrency Control Methods for Real-Time Database Syatem", Helsinki University Printing House,2003.

[4] EPFL, U. Grenoble, INRIA-Nancy, INT-Evry, U. Montpellier 2, U. Paris 6 and U. Versailles, "Mobile Databases: a Report on Open Issues and Research Directions", CNRS, the National Center for Scientific Research in France.December 2002.

[5] Ashish Srivastava, Udai Shankar and Sanjay Kumar Tiwari, "A Protocol for Concurrency Control in Real-Time Replicated Databases System", IRACST – International Journal of Computer Networks and WirelessCommunications (IJCNWC), ISSN: 2250-3501, Vol.2, No.3, June 2012.

[6] KAM-YIU LAM, TEI-WEI KUO, WAI-HUNG TSANG, and GARY C.K LAW, "Concurrency Control in MobileDistributed Real-time Database Systems", Information Systems Vol. 25 No. 4, pp. 261–286, 2000 ã 2000 ElsevierSciences Ltd. All rights reserved Printed in Great Britain 0306-4379/00

[7] Benimon N and Ashvanth B, "Server Based MANETs with Updatable Cache for Consistency and Data Replication for Minimizing Network Contention", International Journal of Computer Networking, Wireless and MobileCommunications (IJCNWMC), ISSN 2250-1568 Vol. 3, Issue 1, Mar 2013, 321-332

[8] 6. K. Chitra and Al-Dahoud Ali, "Prioritized Transaction Management for Mobile Computing Systems", International Journal of Intelligent Computing Research (IJICR), Volume 1, Issue 4, December 2010.

[9] Khuzaima Daudjee and Kenneth Salem, "Lazy Database Replication with Snapshot Isolation", VLDB '06, September 1215, 2006, Seoul, Korea. Copyright 2006 VLDB Endowment, ACM 1595933859/06/09.

[10] K.Segun, A.R. Hurson, and A. Spink, "A Transaction Processing Model for the Mobile data Access System" LNCS2127, pp. 112–127, Springer-Verlag Berlin Heidelberg 2001

[11] Deepak G, Pradeep B." Challenging Issues and Limitations of Mobile Computing", Computer Techology & Applications,Vol 3 (1),177-181, 2012

Ahmad Alqerem obtained a BSc in 1997 from JUST University, a Masters in computer science from Jordan University in 2002 and PhD in mobile computing at Loughborough University, UK in 2008. He is interested in concurrency control for mobile computing environments, particularly transaction processing. He has published several papers in various areas of computer science. After that he was appointed a head of internet technology Depts. Zarka University